THE DESIGN, DEVELOPMENT, CHARACTERIZATION, AND VALIDATION OF A PATHWAY MEASUREMENT TOOL (PATHMET)

by

Eric Joseph Sinagra

B.S. in Physics and Mathematics, Duquesne University, 2011

Submitted to the Graduate Faculty of

The School of Health and Rehabilitation Sciences in partial fulfillment

of the requirements for the degree of

Masters of Science

University of Pittsburgh

2014

UNIVERSITY OF PITTSBURGH

SCHOOL OF HEALTH AND REHABILITATION SCIENCES

This thesis was presented

by

Eric Joseph Sinagra

It was defended on

April 11, 2014

and approved by

Rory Cooper PhD, FISA/PVA, Distinguished Professor and Chair, Department of

Rehabilitation Science and Technology

Mark Schmeler PhD, OTR/L, ATP, Assistant Professor, Department of

Rehabilitation Science and Technology

David R. Smith MURPI, Interlocking Concrete Pavement Institute

Thesis Director: Dr. Jonathan Pearlman PhD, Assistant Professor, Department of

Rehabilitation Science and Technology

THE DESIGN, DEVELOPMENT, CHARACTERIZATION, AND VALIDATION OF

A PATHWAY MEASUREMENT TOOL (PATHMET)

Eric Joseph Sinagra, MS

University of Pittsburgh, 2014

Over 3.6 million people in the United States use a wheelchair for their primary means of mobility, and they rely on functional and accessible pathways to participate in their communities. These wheelchair users are often exposed to dangerous conditions, including vibrations, as they traverse pedestrian pathways. Ambulatory pedestrians also face health risks; approximately one-third of adults 65 or older fall each year, over half of which occur outdoors. Consequently, improving pedestrian pathways is an important task, and this paper describes the development of a tool, referred to as Pathway Measurement Tool (PathMeT), that characterizes pedestrian pathways according to the Americans with Disabilities Act Accessibility Guidelines. The design goals for the product were to develop a user-friendly device that accurately measures slope, level change, and roughness along pedestrian pathways, while gathering pictures and GPS location. Data collection occurred at multiple locations around the United States. Reliability testing was performed to assess the repeatability of PathMeT. Results show that PathMeT is capable of measuring pathway roughness accurately while identifying hazardous level changes. This information can then be uploaded into Geographic Information Systems. Although inclination data was collected, additional development and filtering must occur to record more accurate data to indicate slope. PathMeT has shown to be a reliable device in identifying rough pathways and potential tripping hazards. Many stakeholders believe that PathMeT has great market potential to assist in the planning and reconstruction of pedestrian pathways.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

I would like to thank the University of Pittsburgh, the Human Engineering Research Laboratories (HERL) and all of the tremendous faculty, staff, and students who have guided me along the way. Thank you especially to my advisor, Dr. Jon Pearlman, for his support and wisdom, and for working tirelessly with me on our endeavor to commercialize PathMeT. Thank you to all the shop staff at HERL, Josh Brown, Garett Grindle, Mark McCartney, Ben Gebrosky, and Dalton "Roadhouse" Relich for teaching me how to think like an engineer and for cooperating with me during last minute tasks. Much thanks to Dr. Rory Cooper who continually inspires his students to be more than they think they can be. My deepest gratitude goes to all of the interns and co-ops with whom I have had the honor of working: Tianyang (Tim) Chen, Noelle Greenwald, Alex Kortum, Ian McIntyre, and Dianna Stuckey. I truly could not have done this without you. Thank you very much Jon Duvall for being by my side these past few years, even with our number of arguments, which you most often won. We have become close friends, and I look forward to working with you more in the future. Additional thanks to David Smith and Leroy Danforth for their support and commitment to the PathMeT team. Thank you to Dr. Mark Schmeler for you guidance and wisdom. Thank you to my friend Ben Moreno, who encouraged me to join this program. Most importantly, thank you to my family for always being there to love, support, and encourage me: my parents, Ken and Rosalie; my brothers, Nick and Ken; my gram; and my amazing girlfriend, Ali, for putting up with my procrastination. I love you all!

Abbreviations:

ADA = Americans with Disabilities Act of 1990;

ADAAG = Americans with Disabilities Act Accessibility Guidelines;

ASTM = ASTM International, formerly the American Society for Testing and Materials;

BIA = Brick Industry Association;

FHWA = Federal Highway Administration;

GPS = Global Positioning System;

GIS = Geographic Information Systems;

HPMS = Highway Performance Monitoring System;

Hz = Hertz;

ICPI = Interlocking Concrete Pavement Institute;

IRI = International Roughness Index;

ISO = International Organization for Standardization;

m = meter;

mm = millimeter;

MPa = Megapascal;

PathMeT = Pathway Measurement Tool;

PRI = Pathway Roughness Index;

PROWAG = Public Rights-of-Way Accessibility Guidelines;

PSD = Power Spectral Density;

PSI = Present Serviceability Index;

RMS = Root Mean Square;

SPSS = Statistical Package for the Social Sciences;

SurPRO = Surface Profiler;

ULIPs = Ultra Light Inertial Profiler for sidewalks;

WBVs = Whole-Body Vibrations;

WT = Wavelet Theory

1.0   INTRODUCTION

The Americans with Disabilities Act (ADA) indicates that "physical or mental disabilities in no way diminish a person's right to fully participate in all aspects of society…" (ADA, 1990). Title V of the ADA directs the Architectural and Transportation Barriers Compliance Board (Access Board) to create minimum guidelines "to ensure that buildings, facilities, rail passenger cars, and vehicles are accessible, in terms of architecture and design, transportation, and communication, to individuals with disabilities" (ADA, 1990). However, the Access Board has only established one guideline concerning ground surfaces, stating that they "shall be stable, firm, and slip resistant" (ADAAG, 2002). No guidelines currently exist that relate pathway roughness to pedestrian safety and comfort.

The need for improved pathway regulations continues to grow. Approximately 3.6 million Americans currently use wheelchairs (Brault, 2012) and 26% of the population is over 55 years old, many of which have an increased risk of tripping or falling (Census, 2012). Consequently, litigation continues to increase and cost cities millions of dollars as a result of sidewalks not being compliant with the ADA. In Los Angeles, the city settled two cases about sidewalk accessibility worth $85 million combined. Los Angeles stated that 42% of the 10,750 miles of sidewalk are in disrepair, sparking the desire for improvement.   California has committed $1.1 billion over the next 30 years to improving its state-controlled pedestrian

pathways. Similarly, Sacramento has committed 20% of its annual transportation fund over the next 30 years to repairs of its 2,300 miles of sidewalk (Bloomekatz, 2012; Works, 2014).

1.1 PATHWAY HEALTH RISKS

1.1.1 Wheelchair Users

One measurement that helps determine the safety and comfort of wheelchair users is their level of exposure to whole-body vibrations (WBVs), which research has shown can lead to a variety of medical issues (Wolf, Cooper, Pearlman, Fitzgerald, & Kelleher, 2007), especially with the back and neck (Boninger, Cooper, Fitzgerald, Dicianno, & Liu, 2003; Seidel & Heide, 1986). Secondary conditions that wheelchair users often face, such as pressure ulcers and back pain, are associated with use of rough or uneven pathways and can be detrimental to recovery (DiGiovine et al., 2003). Furthermore, research has shown that many wheelchair cushions actually amplify vibration exposure, worsening conditions for the wheelchair user (Garcia-Mendez, Pearlman, Cooper, & Boninger, 2012).

Although WBVs can be dangerous, falling from the wheelchair is the most common type of injury for a wheelchair user (Faul, Xu, Wald, & Coronado, 2010); many of these falls occur while traversing poorly maintained outdoor pathways (Kirby & Ackroyd-Stolarz, 1995). Similarly, sloped riding surfaces create stability issues for wheelchair users (Cooper & MacLeish, 1995; Stout, 1979). A study shows that 90% of the tips and falls that occurred while riding uphill were in the backward direction (Gaal, Rebholtz, Hotchkiss, & Pfaelze, 1997).

1.1.2   Ambulatory Pedestrians

Non-wheelchair users also face potential hazards, such as falling while traversing unmaintained or poorly designed pathways. Falling is the most common cause of traumatic brain injury (Sterling, O'Connor, & Bonadies, 2001) and is considered a public health risk because it impinges on the public rights-of-way (Materials, 1996). Trips and falls are the number one cause of fatal and nonfatal injuries in older adults with 2.3 million fall-related injuries yearly, 662,000 of which result in hospitalization (Centers). In addition, more than 33% of people older than 65 fall each year (Tromp et al., 2001), and are two to three times more likely to fall again (Control). In 2010, the direct medical costs due to trips and falls was $30 billion (Stevens, Corso, Finkelstein, & Miller, 2006).

Over half the number of falls among older adults occurs outdoors (Weinberg & Strain, 1995; Bergland, Pettersen, & Laake, 1998; Bath & Morgan, 1999; Bergland, Jarnlo, & Laake, 2003). Table 1 shows the results of a study (Li et al., 2006): "outdoor falls accounted for 72% of the most recent falls among middle-aged men, 57% of the falls among older men, 58% of the falls among middle-aged women, and 51% of the falls among older women." Furthermore, these falls were most common on sidewalks, curbs, and streets. Outdoor falls were most commonly caused by one or more environmental causes, including an uneven surface, a wet surface, and tripping over an object. Falls on sidewalks were predominantly caused by an uneven surface. These results present an important reason for the improvement of pedestrian pathway conditions.

Table 1: Percentages of self-reported outdoor falls with selected characteristics by gender and age group (Li et al., 2006)

		Men			Women		
Characteristic	Total (N = 297)	Middle-Aged (45-64 y) (n = 52)	Older (≥65) (n = 31)	Total (n = 83)	Middle-Aged (45-64 y) (n = 133)	Older (≥65) (n = 81)	Total (n = 214)
Place							
Sidewalk, curb, street	34.0	32.7	25.8	30.1	35.3	35.8	35.5
Garden, patio, porch, deck	23.6	5.8	48.4	21.7	20.3	30.8	24.3
Outdoor park, recreation area	14.5	23.1	6.5	16.9	17.3	7.4	13.6
Parking garage, parking lots	10.5	25.0	3.2	16.8	4.5	13.6	8.0
Outdoor stairs	8.8	1.9	3.2	2.4	14.3	6.2	11.2
Other outdoor place	8.8	11.5	12.9	12.0	8.3	6.2	7.5
Activity during fall							
Walking	47.3	30.8	48.4	37.4	54.9	45.0	51.2
Vigorous activity[a]	15.2	38.5	3.2	25.3	12.8	8.8	11.3
Walking up or down stairs	16.9	13.5	9.7	12.1	18.8	18.8	18.8
Other outside activity[b]	20.6	17.3	38.7	25.3	13.5	27.5	17.8
Environmental cause							
Uneven surface	47.6	36.0	45.2	39.5	56.2	42.0	50.7
Wet surface	21.0	13.7	30.0	19.8	27.5	12.3	21.7
Tripped on something	33.5	27.5	35.7	30.4	32.1	38.9	34.6
Slipped on something	20.2	19.6	9.7	15.9	28.7	13.6	22.9
Any of the above	73.4	61.5	74.2	66.3	78.9	71.6	76.2
Other							
On hard surface	71.3	70.6	58.1	65.9	72.2	75.3	73.4
From standing height or higher	95.2	96.0	100.0	97.5	95.4	92.5	94.3
Forward	46.8	36.3	48.4	41.5	46.6	52.6	48.8
Sideways	23.6	31.4	25.8	29.3	18.8	25.6	21.3
Backward	15.0	19.6	12.9	17.1	18.1	7.7	14.2
Straight down	12.0	5.9	9.7	7.3	13.5	14.1	13.7

[a]Vigorous activities include jogging, running, bicycling, dancing, and other activities.
[b]Other activities include turning around (n = 9), reaching up or down (n = 11), getting into or out of a chair (n = 1), sitting or lying down (n = 2), bending over (n = 4), getting into or out of a motor vehicle (n = 3), gardening or house repair (n = 4), climbing up or down a ladder or a stool or getting over a large object (n = 3), and other (n = 10).

1.2 ROUGHNESS STANDARD

For the purposes of limiting WBVs in wheelchair users, the current guidelines are insufficient, being both subjective and making no mention of surface roughness, an important metric when

the average manual and power wheelchair user travels 2.0 km (Cooper et al., 2002) and 1.6 km (Tolerico et al., 2007) per day respectively. ISO 2631-1, *Mechanical vibration and shock -- Evaluation of human exposure to whole-body vibration* (ISO, 1997), is a widely accepted international standard that establishes recommended limits on the exposure of humans to WBVs (Figure 1). Specifically, the ISO 2631-1 states that a maximum exposure of a root mean squared (RMS) value of 1.15 m/s^2 over 4-8 hours is the recommended limit. In order to limit these harmful vibrations, regulations regarding the roughness of pedestrian pathways are needed.
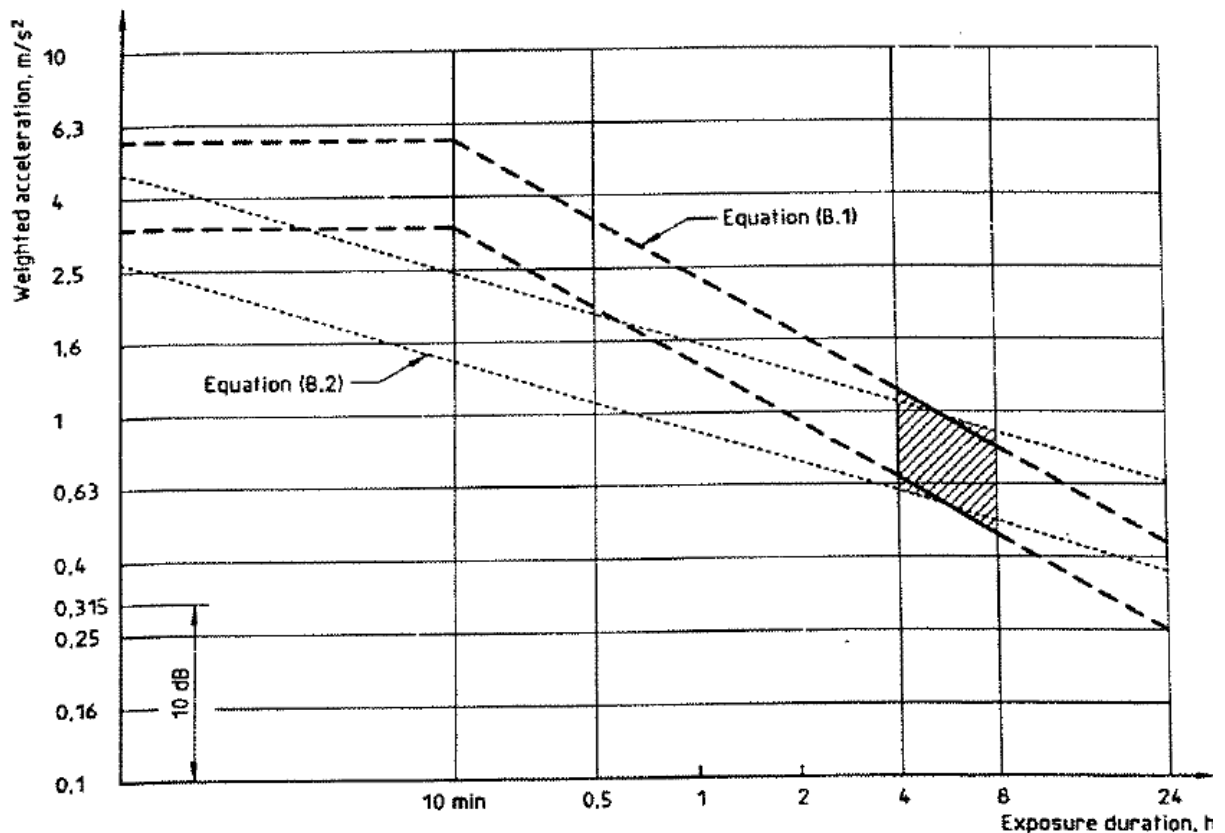
Figure 1: ISO Standard 2631. Caution zone between the dashed lines (Duvall, 2013)

To address this need, the Access Board funded a study to investigate the correlation between surface roughness of pathways and vibrations experienced by wheelchair users as they travel over these surfaces (Duvall et al, 2013; Duvall, Sinagra, Stuckey, Pearlman, & Cooper,

2014). Both engineered and community-based surfaces were used in the study. Nine engineered wooden surfaces were used with periodic gaps for different roughness. The gaps were 0, 20, 32, 39, and 51-milimeter widths, and were spaced every 0, 102, or 203 millimeters. Wheelchair users propelled their wheelchair over the surfaces three times each while acceleration data were collected on the seat, backrest, and footrest. After each surface, the wheelchair user answered questions concerning rider comfort and rating of the surface. The identical protocol was used over a selection of community-based surfaces.

The results from the study show that wide cracks in surfaces cause wheelchair users to be exposed to dangerous WBVs. In addition, regardless of the level of WBVs, wide cracks can cause discomfort as reported by wheelchair users. Figure 2 shows that as the roughness of a surface increases as RMS acceleration increases. Consequently, average acceptability rating (which is related to comfort) decreases (Duvall et al., 2013). The ISO 2631-1 standard (Figure 1) specifies that RMS accelerations of 1.6 m/s^2 or greater are dangerous for a period of one hour or longer (ISO, 1997). Figure 2 demonstrates that travelling over some surfaces may be harmful to the health of wheelchair users.
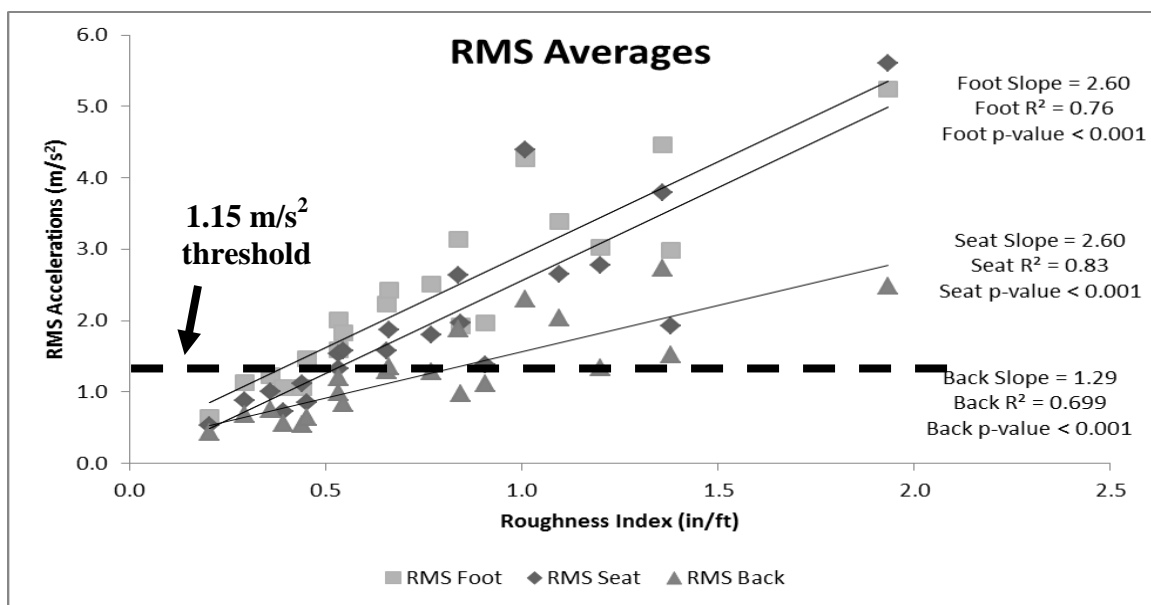
Figure 2: Total RMS averages across all surfaces (Duvall, 2013)

6

Table 2 shows the results from the study by Duvall (2013). The RMS values, ISO 2631-1 health guidance zone limits, and roughness indices are shown. Based on these results, which were compared with questionnaire results from test subjects, a roughness limit of 100 mm/m was established for distances less than 3 m. This corresponds with the lower limit for ISO2631-1 for vibrations lasting less than 10 minutes. This roughness relates to surfaces that were acceptable by 50% of wheelchair users in the study. Similarly, a threshold of 50 mm/m was selected for distances greater than 30.5 m, and corresponds to the ISO 2631-1 2-hour health guidance zone boundary. A surface with this roughness was found to be acceptable by more than 75% of wheelchair users in the study.

Table 2: Roughness index values for duration of exposure

Exposure Time	2 hr		1 hr		30 min		< 10 min	
Health Guidance Zone Boundary	Low	High	Low	High	Low	High	Low	High
RMS Limit (m/s^2)	0.7	1.6	0.85	2.5	1.1	3.5	3.5	6
All surfaces, All Chairs	0.30	0.67	0.36	1.04	0.47	1.44	1.44	2.47
All Surfaces, Manual Chairs	0.25	0.52	0.29	0.79	0.37	1.09	1.09	1.84
Engineered Surfaces, Manual Chairs	0.20	0.53	0.26	0.86	0.35	1.22	1.22	2.13

1.3 PUBLIC RIGHTS-OF-WAY ACCESSIBILITY GUIDELINES

The Public Rights-of-Way Accessibility Guidelines (PROWAG) are currently being finalized for publication by the Access Board, and include guidelines on pathway cross slope, running slope, and level change. The preamble of the PROWAG will discuss surface roughness, but the timing of the PROWAG public review occurred well before results of the pathway roughness research, and thus it will not be included explicitly. Thus, promulgation and enforcement of the roughness standard will occur after publication of PROWAG and will occur according to two potential

timelines. One possibility is that if the Department of Transportation or Department of Justice adopts the PROWAG, which they are widely expected to do, then the roughness standard could be adopted simultaneously as an amendment to the PROWAG. If this were to occur, then the roughness standard could be enforceable before 2015. The second possibility is that the roughness standard is added as an amendment to the PROWAG when it is revised, which would likely be in 2015 or 2016. Figure 3 and Table 3 provide the Americans with Disabilities Act Accessibility Guidelines (ADAAG) for accessible route as determined by the Access Board.

Regardless of when the roughness standard is adopted, once the PROWAG preamble is available for review, stakeholders will recognize the importance of considering roughness in pathway compliance, and make an effort to measure it. In order to evaluate pedestrian pathways in an objective manner, a device that measures surface roughness and the other accessibility characteristics is required. Although devices designed to measure the roughness of roads and highways exist, these are insufficient for measuring walkways. These devices, being designed for cars, typically measure surface profiles at a minimum resolution of 25 millimeters along the route of travel. In contrast, the smaller wheels of wheelchairs are more sensitive to minor surface imperfections necessitating a more accurate profile measurement.
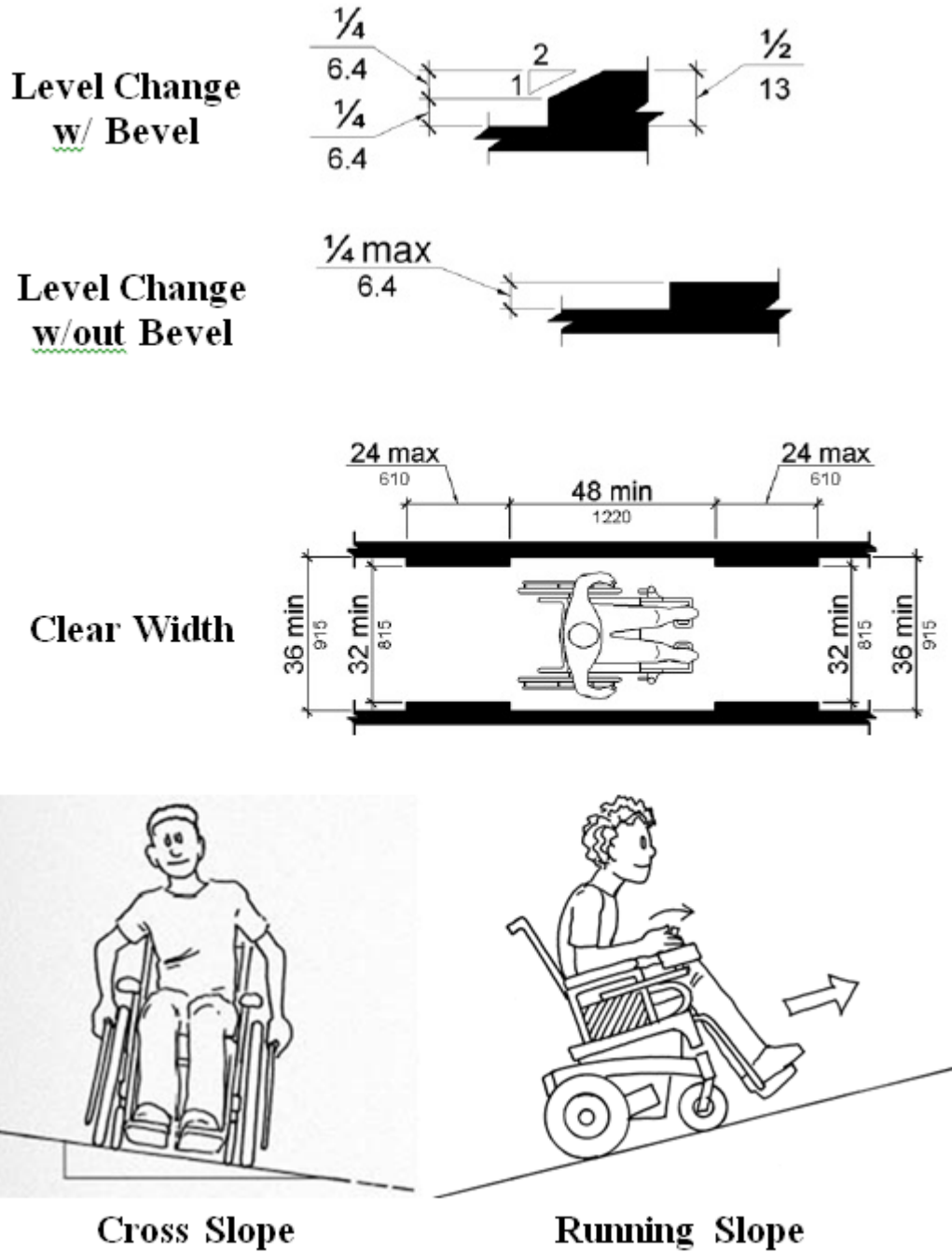
Figure 3: ADAAG guidelines for accessible routes (Justice, 2010; FHWA, 2014)

Table 3: ADAAG accessible route guidelines

Parameter	Requirement
Clear Width	Minimum 914 mm
Openings	Maximum 13 mm
Obstacle Height:	
< 6.35 mm	No slope required
6.35 – 12.7 mm	Beveled with maximum 1:2 slope
> 12.7 mm	Treat as ramp
Ramps Max Slope:	
1:12 – 1:16	Maximum 762 mm high, 9.1 m long
1:16 – 1:20	Maximum 762 mm high, 12.2 m long
Cross Slope	Maximum 1:50

1.4 MEASUREMENT TECHNIQUES

Roughness is calculated from a longitudinal profile along a wheel path of the surface. There are several methods of capturing these profiles including the rod and level, dipstick, profilometer (Figure 4), rolling profilers, and inertial profilers (Figure 5). Table 4 shows the pros and cons of each measurement system to be adapted to sidewalks and wheelchair pathways.
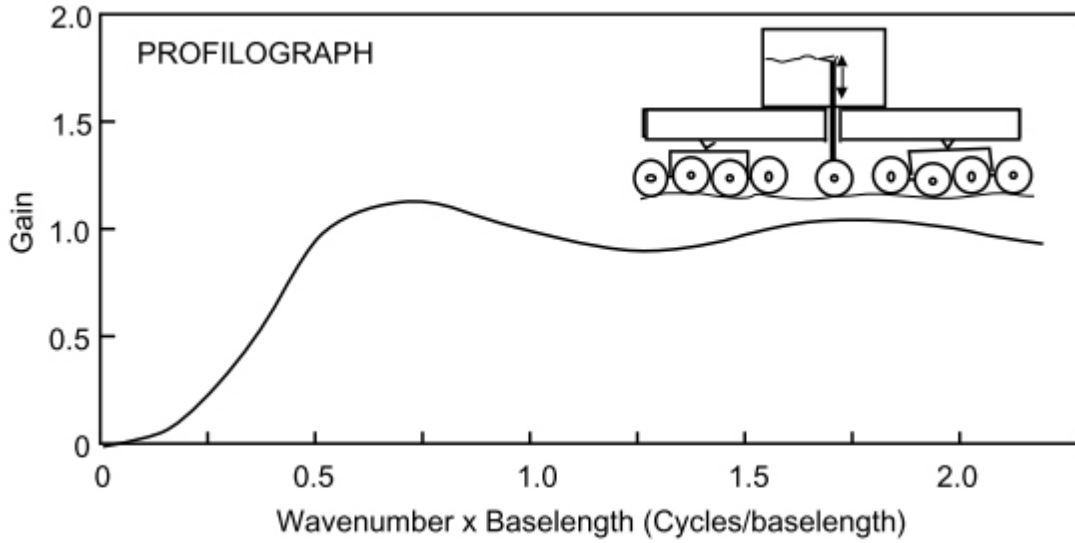
Figure 4: Image of profilograph measurement technique (Sayers & Karamihas, 1998)
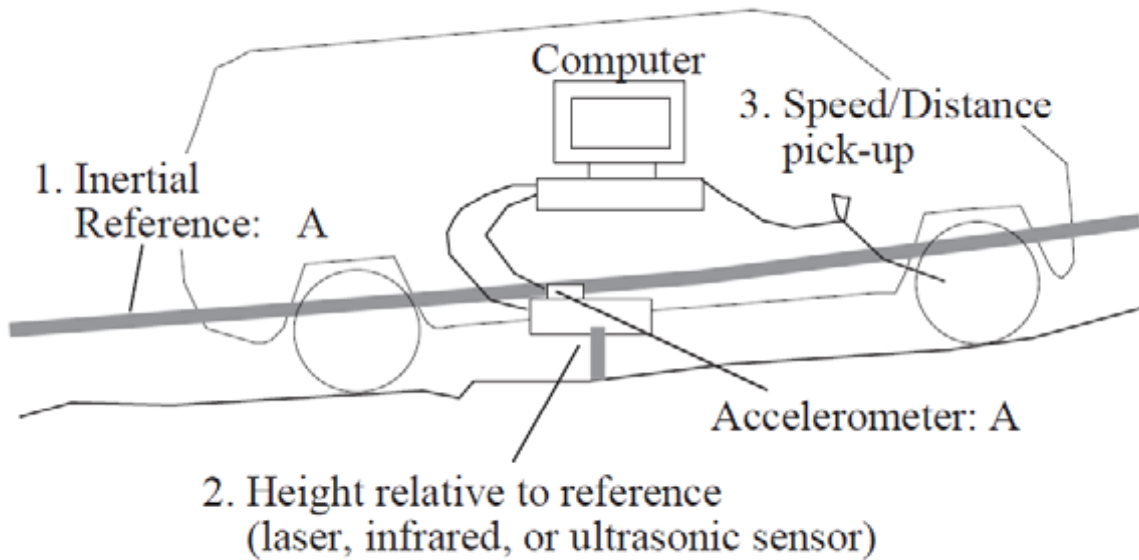
Figure 5: Image of inertial profiler measurement technique (Sayers & Karamihas, 1998)

Table 4: Comparison of surface measurement techniques (Sayers & Karamihas, 1998)

Measure Device	Measurement Process	Advantage	Disadvantage
Rod and Level	Inclinometer and Laser	Simple, Extremely accurate	Slow process
Dipstick	Inclinometer	Simple, Very accurate, Low cost	Short profiles, slow process
Profilometer	Profilograph	Cost effective, Faster measures	Wide variation in response properties, Only measures certain wavelengths
Inertial Profiler	Accelerations and Displacements	Can be mounted to any vehicle	Expensive, Requires a certain amount of speed to work

Another method to evaluate roadway surfaces is the Present Serviceability Index (PSI), which is the most commonly used method for relative objective measures of surface condition with the public's perception of serviceability. The primary use of PSI is to evaluate the ability of the pavement to serve its users by providing safe and smooth driving surfaces. This method involves a group of panelists riding in a car over the roadways and filling out a PSR (Present Serviceability Rating) form. Manual observation is still considered possibly the strongest and most accurate evaluation of a road surface because of the attention to detail; however, it requires a substantial amount of labor-hours and other associated cost (Sayers & Karamihas, 1998; Washington, 2005; Latif, 2009).

1.4.1 Analysis Techniques

There are also many analysis techniques to make sense of these profiles. These include the International Roughness Index (IRI), Power Spectral Density (PSD), and Wavelet Theory (WT). The technique used to calculate the roughness of a pedestrian pathway surface is similar to the

IRI. ASTM E1926, *Standard Practice for Computing International Roughness Index of Roads from Longitudinal Profile Measurements*, (International, 2008) reports that IRI roughness data for roadways is used by local, state and federal agencies in pavement management systems. In addition, IRI is used by the U.S. Federal Highway Administration (FHWA) as the input to their Highway Performance Monitoring System (HPMS) (Sayers & Karamihas, 1998). For this reason, the key aspects of the IRI are utilized for analysis since it is a widely accepted measurement of roughness. IRI is calculated as the sum of vertical deviations normalized by the horizontal distance travelled (i.e. meters/kilometer or inches/mile). For pathways, the Pathway Roughness Index (PRI) has been defined as the sum of vertical displacements of a wheel normalized by the distance traveled (Duvall et al., 2013). The calculation of PRI is based on the wheel-path of a 63.5 mm wheel traveling over the surface, which acts as a low-pass filter to the raw longitudinal profile data. This wheel size was chosen because it is the smallest and highly common wheel found on manual wheelchairs. Figure 6 shows this path as the 63.5 mm wheel travels across gaps within a surface profile. The PRI of a surface is reported in units of millimeters per meter or inches per foot.

As the number of wheelchair users in the United States increases each year, it is important to reduce the number of uneven pathways that cause harmful WBVs to these Americans. The upcoming development of the surface roughness standard and the accompanying analysis technique for measuring surface profiles creates a need for a commercially available product capable of determining pathway roughness in a community setting. This paper describes the design, fabrication, testing, and characterization of a Pathway Measurement Tool (PathMeT), developed for the purpose of measuring pathway accessibility characteristics.
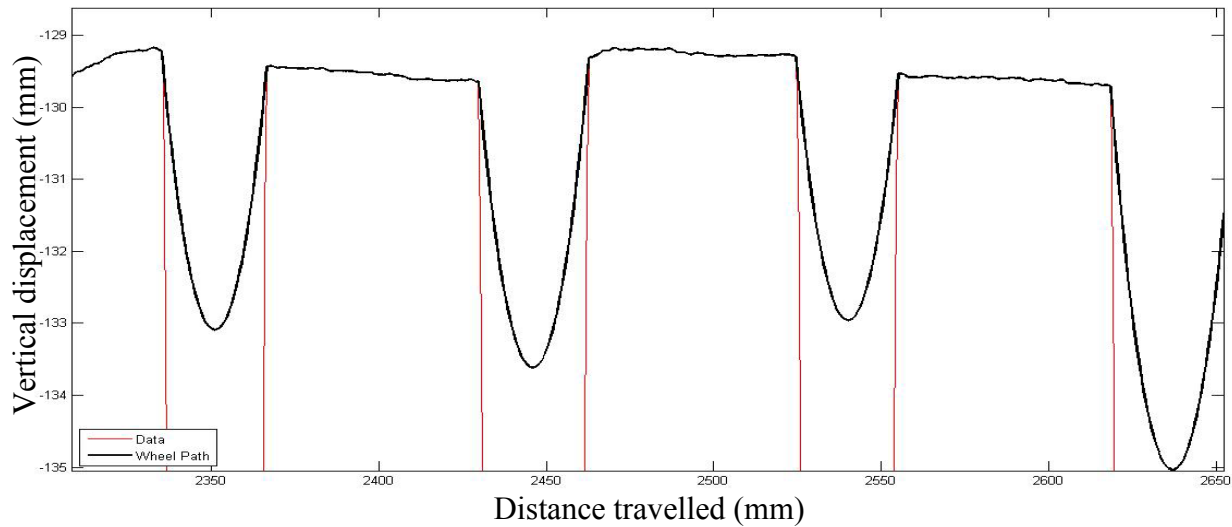
Figure 6: Graph of wheel-path analysis to determine PRI

1.5 SIDEWALK MEASUREMENT DEVICES

1.5.1 SurPRO 3500

SurPRO 3500 (Figure 7) is a rolling surface profiler whose primary use is for roads, structures, runways, and floors.   SurPRO uses an inertial stabilizer that allows it to collect unfiltered elevation profiles. It claims to produce profile data that compares with that of the Dipstick, as well as the rod and level (SurPRO, 2011).   The product is commercially available through International Cybernetics, Inc. Table 5 shows a list of product specifications for SurPRO 3500.

Figure 7: SurPRO 3500 device

Table 5: SurPRO 3500 specifications

Specification	Metric
Sample Interval	1, 2, 5, 10, 20, 25, 30, 50, 100 cm 1.0, 2.0, 5.0, 9.5, 9.8, 10, 11.8, 12.0 in
Data Collection Rate	Nominal: 2 km/hr (1.25 mile/hr) Max: 4 km/hr (2.5 mile/hr)
Weight	20.4 kg (42 lb)
Size	254 mm (10") W x 482 mm (19") L
Battery Life	15 hours w/out motor drive
Wheel Size	152.4 mm (6") diameter x 70 mm (2.76") wide

1.5.2 Magic Cart

The Magic Cart (Figure 8) is a three-wheeled cart developed by Beneficial Designs. The purpose of the device is to assess sidewalk compliance according to the ADA. Consequently, Beneficial Designs has developed the Public Rights-of-Way Assessment Process to accompany data collection. Magic Cart uses sensors that measure grade, cross slope, and distance travelled,

which is collected with a laptop. A GPS receiver and camera collect location and imagery, respectively. The front wheel is detachable, allowing the user to measure areas that the device cannot reach. Finally, there is a detachable tool that is used to manually measure trip hazards (Greenwald, 2013).

Figure 8: Magic Cart device

1.5.3   ULIPs

The ULIPs (Figure 9), which stands for Ultra Light Inertial Profiler for sidewalks, is a Segway-based system developed by Starodub, Inc. in Kensington, MD. Data collection occurs by a user who rides the Segway over a surface. Attached to the Segway are a laser measurement system, three accelerometers, and a gyroscope that are capable of measuring the sidewalk profile at 10,000 data records per second. The sensors gather accurate data about location, slope, and

surface variations, which can then be integrated into a city's asset management database and Geographic Information Systems (GIS). The ULIPs has been used in the following places: Bellevue, WA; County of St. Louis, MO; City of San Carlos; City of Clovis, CA; and City of San Marcos, CA (Khambatta & Loewenherz, 2011; Starodub, 2009).

Figure 9: ULIPs device

1.5.4 Level and Tape Measure

The current gold standards for measuring sidewalks are the level and tape measure (Figure 10). According to Khambatta & Loewenherz (2011), one user walks along a sidewalk placing the level every meter measuring the running slope and cross slope, while slab-to-slab faulting is measured with the tape measure. Measuring with this technique can progress at less than one mile per hour. Using digital imagery is also common practice in the analysis of sidewalks (Applied Research Associates, personal communication, February 13, 2014). Furthermore,

analysis can be difficult due to the numerous images and measurements (Khambatta &

Loewenherz, 2011).

Figure 10: Image of level and tape measure

2.0 METHODS

2.1 GOALS

The initial goal of this project was to develop a device that is capable of measuring roughness according to the new pathway roughness standards. The goal quickly pivoted towards developing a tool that could characterize the accessibility of pedestrian pathways based on the following parameters: flatness, running slope, cross slope, level change, and roughness. The design is targeted for all stakeholders involved in designing, constructing, and evaluating pedestrian pathways. Table 6 shows the design objectives. Specifications were set (Table 9) and testing was performed to determine if these objectives were fulfilled.

Table 6: PathMeT project design objectives

Objectives
1. Measure pathway roughness accurately and quickly
2. Measure pathway profiles with 1mm resolution or better.
3. Fit inside the trunk of a typical automobile for easy transport
4. Compatible with ProVAL and other surface analysis software
5. Capable of recording the specific measurements for surfaces to comply with ADA Accessibility Guidelines, including cross slope, running slope, and level change
6. Capable of operating for the duration of a typical work day on a single charge of its battery
7. Measure surfaces consistently

2.2 INITIAL PROTOTYPES

Two initial proof-of-concept prototypes were developed by using two sensors to determine the profile of surfaces: an Acuity AR700 laser displacement measurement tool and an optical incremental encoder. The laser collected distance measurements by using a single point laser beam and a triangulation technique to geometrically determine the distance from the laser device to the ground surface. In addition, two gears, one attached to the encoder and one to the wheel hub, enabled the encoder to collect data determining the distance travelled by PathMeT.

2.2.1   Prototype 1 (P1)

The first prototype, P1 (Figure 11), was developed from the base of a power wheelchair and was supported by funding from the Access Board. This device was used in the research to develop the roughness standard (Duvall et al., 2013). P1 was a robotic system that was controlled by a joystick. When collecting profile data, P1 was driven over two parallel pieces of plywood 'tracks' (Figure 12) allowing travel over a flat surface. This ensured that measurements were unbiased by the wheels traversing rough terrain. P1 was used as a starting point for the development of the next prototype, P2, which was supported by the Interlocking Concrete Pavement Institute (ICPI) and Brick Industry Association (BIA).
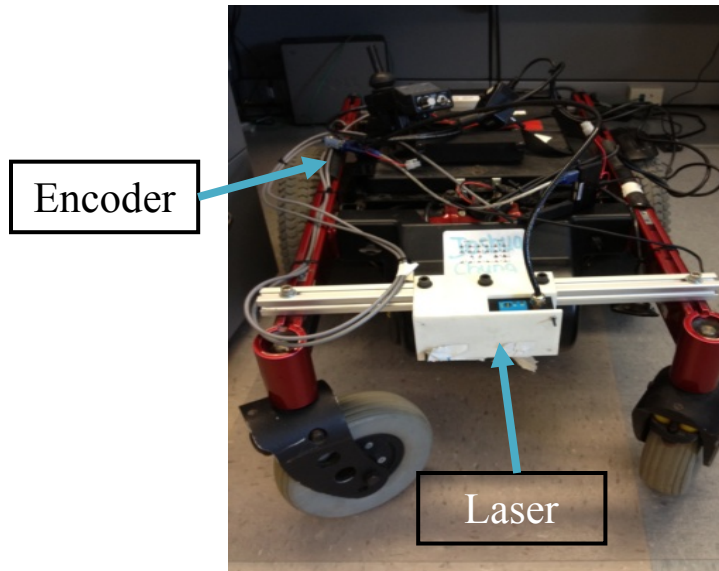
Figure 11: P1 prototype

Figure 12: P1 driven over plywood tracks

2.2.2 Prototype 2 (P2)

The chassis of P2 (Figure 13), the second prototype, was taken from a jogging baby stroller since jogging strollers are designed to be pushed over sidewalks at a fast pace while eliminating

vibrations. The frame was adapted to include enclosures for the laser and encoder. In order to reduce the deformation of the tires while travelling over cracks and imperfections, the wheels of this manually propelled device were inflated to a firmness of 0.4 MPa (60 psi). Two different data collection methods were utilized: driving directly over surfaces and driving over plywood tracks, similar to the method for P1. Users made sure not to bias data by pushing down on the stroller handle as to lift the front wheel off the ground.
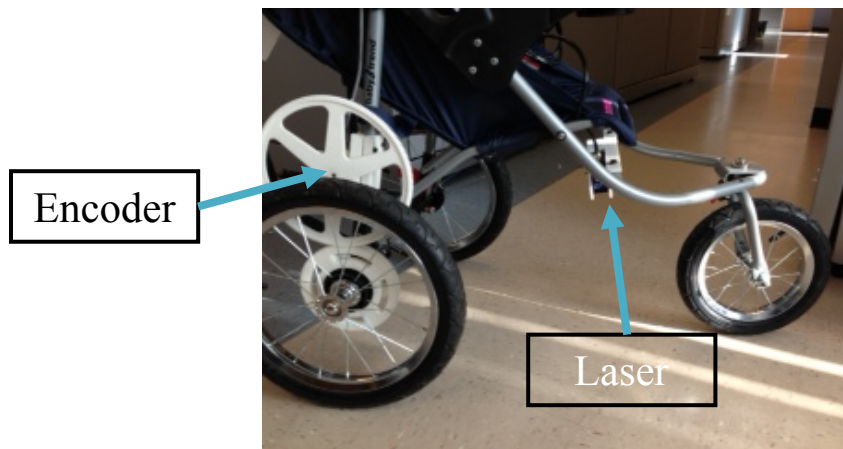
Figure 13: P2 prototype

2.3 SURFACES

2.3.1   Engineered Surfaces

Nine different engineered surfaces were analyzed in order to compare measured roughness with the corresponding known roughness. These surfaces are the same ones used in the study to determine a roughness standard for pedestrian pathways (Duvall, 2013). They were developed on a 4.9x1.2 m wooden runway (Figure 14a) where interchangeable wooden planks (Figure 14b)

of varying sizes generated different surfaces with different roughness. Table 7 shows the nine

surfaces with gaps ranging from 0 to 51 mm, evenly spaced at 102-, 203-, or 305- mm intervals.
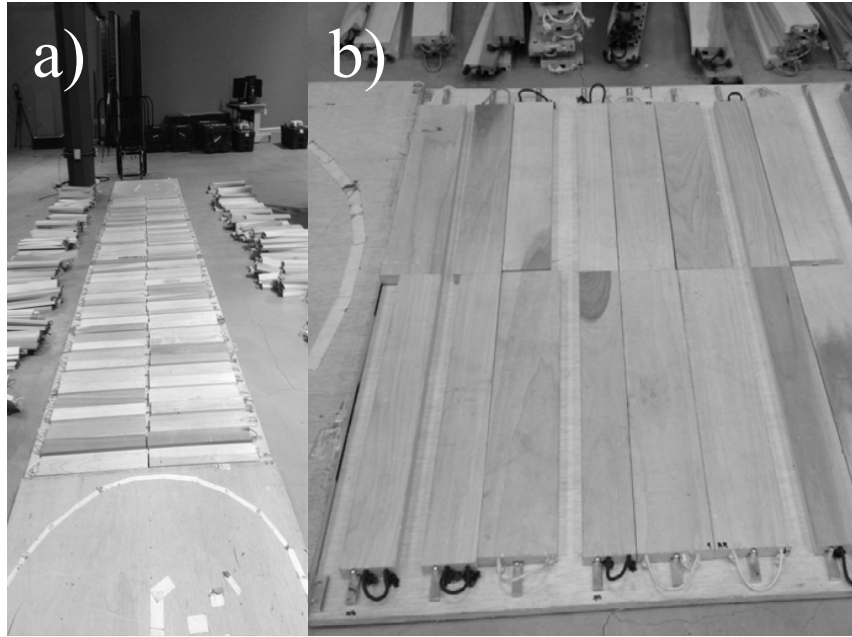
Figure 14: Engineered wooden surface design

Table 7: Engineered surfaces specifications

Surface ID	Crack Frequency (mm)	Crack Width (mm)
E1	N/A	0
E2	305	20
E3	203	20
E4	305	32
E5	102	20
E6	203	32
E7	203	39
E8	305	32
E9	203	51

2.3.2 Community-Based Surfaces

Six different community-based surfaces (Figure 15) were studied. The surfaces were made of

concrete, stamped brick, brick, and asphalt. The quality and method of laying the surfaces

varied. In order to normalize the method of collecting data, only 4.9 meters were examined for each surface.

Figure 15: Six community-based surfaces used for
prototype data collection

2.3.3 Data Collection and Analysis

Both PathMeT prototypes were used to measure surface roughness while rolling at a speed of 1 m/s (+/- 10%). This speed was selected to match the speed at which subjects were asked to propel their wheelchairs in the study to develop a roughness standard (Duvall et al., 2013). Three successful trials were collected for each surface, and the roughness indices were averaged. Data from the laser and encoder were collected using a laptop with MATLAB and Realterm. Laser data was collected at a sampling rate over 1000 Hz to guarantee high accuracy. The encoder collected data at a sampling rate of approximately 1100 Hz. Time stamps were recorded for every measurement taken.

Additional techniques were used internally to guarantee that PathMeT measures with better than 1mm resolution. It was found that the difference between consecutive encoder points is always one. (Two programs were written, one in MATLAB and one in Java programming language ensure 1mm resolution. The source code can be seen in A.1.) This means that PathMeT obtains a reading every encoder point. It was determined that there are approximately 35 encoder points for every 25.4 mm of distance travelled. Therefore, there is an encoder reading approximately every 0.7 mm. This means that millimeter resolution is indeed obtained.

Analysis began by filtering the laser using a 3-point moving average to minimize noise. Next, a "wheel path" algorithm was applied to determine the path a wheel would take when travelling over a surface. This is the same analysis technique described above. The wheel path is not necessarily the same as the profile since a large wheel would never travel to the bottom of a deep but narrow crack. The wheel size used for analysis was a 63.5 mm diameter tire with no deflection, similar to rollerblade wheels. This size was used to simulate a worst-case scenario for wheelchair users; some manual wheelchairs have front caster wheels this size. PRI resulted from analysis of the surface.

2.4 PATHMET DESIGN CONCEPTS

The design of PathMeT, which stands for Pathway Measurement Tool, began by considering multiple concepts (Table 8). Original sketches can be found in Appendix B. One concept was the three-wheeled rolling design, based on the P2 design. Ultimately this design was selected because it is easy to propel over uneven surfaces and collect data quickly. A second design

concept considered was a tracked system. This system would be beneficial because it would eliminate errors experienced by traversing rough terrain with wheels. However, this treaded tracked system would need to be a robotic system, which would increase costs, since it would be difficult to propel with tracks. A third design concept was a railed system that allows sensors to slide along a track without touching the ground. The advantage of the long railway track would be its ability to completely eliminate errors caused by wheels since the track would be stationary on the ground. In addition, data would be collected in one long pass over the surface. However, the time and effort needed to set up this design does not make it user-friendly. Table 8 shows a comparison of the pros and cons of the three methods. Therefore, a user-friendly rolling device was designed to improve speed, while maintaining accuracy and keeping costs low. This method of data collection is referred to as Rolling Mode (described below).

Table 8: Comparison of three PathMeT design concepts

	Method	Pros	Cons
Wheeled		- Quick data collection - Inexpensive	- Some noise due to contact with surface
Tracked		- Autonomous robot - Tracks reduce noise due to contact with surface	- Expensive
Railed		- Eliminates noise due to contact with surface	- Time consuming compared with Wheeled and Tracked methods

2.5   PATHMET TARGET SPECIFICATIONS

After collecting pilot roughness data with P1 and P2, it was deemed that the use of the laser and encoder were appropriate sensors to use with future prototypes. Although P1 was user-friendly because of the joystick capabilities, it was decided that the development of a robotic system would be expensive. The jogging stroller was the primary influence for the final design. The three-wheeled rolling system makes it easy to manually propel. The size of the wheels assists in

reducing vibrations, although solid tires are preferred. These specifications are the foundation of the next design. Additional target specifications (Table 9) are provided here:

Table 9: PathMeT target specifications

Target Specifications	Related Objective
1. Target Weight (Disassembled): 22.7 kilograms	N/A
2. Target Weight (Assembled): 29.5 kilograms	N/A
3. Target Physical Dimensions (Disassembled with push-handle collapsed) (mm): 1016L x 635W x 508H	Objective 3
4. Target Physical Dimensions (Assembled) (mm): 1524L x 635W x 1219H	Objective 3
5. Battery Life: 8 hours	Objective 6

2.6 PATHMET RELIABILITY TESTING

Figure 16 shows the three different surfaces measured during reliability testing and characterization of PathMeT. Surface A is a 4.9 x 1.2 meter engineered surface comprised of two rows of pieces of 19 mm poplar hardwood. The 24 pieces in each row are arranged so that there is a 32 mm gap every 203 mm. Surface B and Surface C are a typical concrete and stamped concrete surface, respectively. A 4.9-meter segment of data was collected along Surfaces B and C so that it would compare with Surface A for reliability testing.
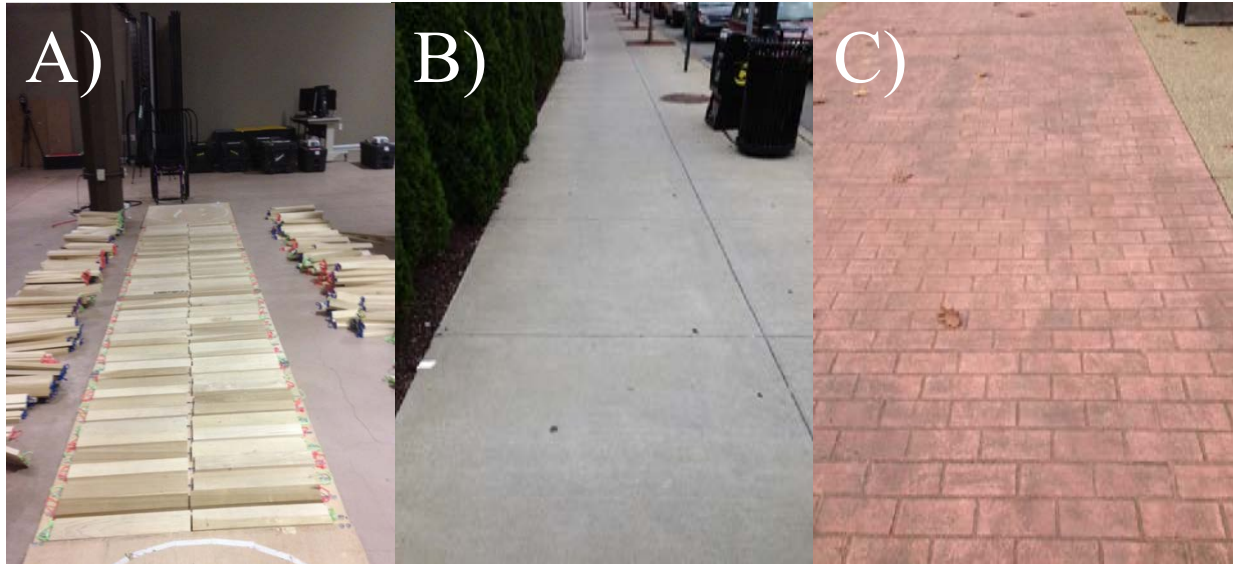
Figure 16: Surfaces used for testing and characterization of PathMeT:
(A) engineered surface (B) concrete surface (C) stamped concrete surface

There were two different phases of the testing protocol: 1) intra- and inter-rater reliability 2) level change characterization. In order to test for inter- and intra-rater reliability, three individuals propelled PathMeT over Surfaces 1, 2, and 3. For each surface, the user propelled PathMeT along three different paths. Level change characterization consisted of one user propelling PathMeT three times up and down steps of 6.35-, 12.7-, 19.05-, and 25.4- mm. The steps were comprised of 610 x 1220 mm sheets of 6.35 mm thick MDF wood placed above collapsed tabletops to provide a solid level surface. Figure 17 shows this testing assembly. Additional testing was conducted to observe the effect of laser placement on surface profiles. Figure 18 shows the three laser placements considered: behind, under, and in front of the back axle.
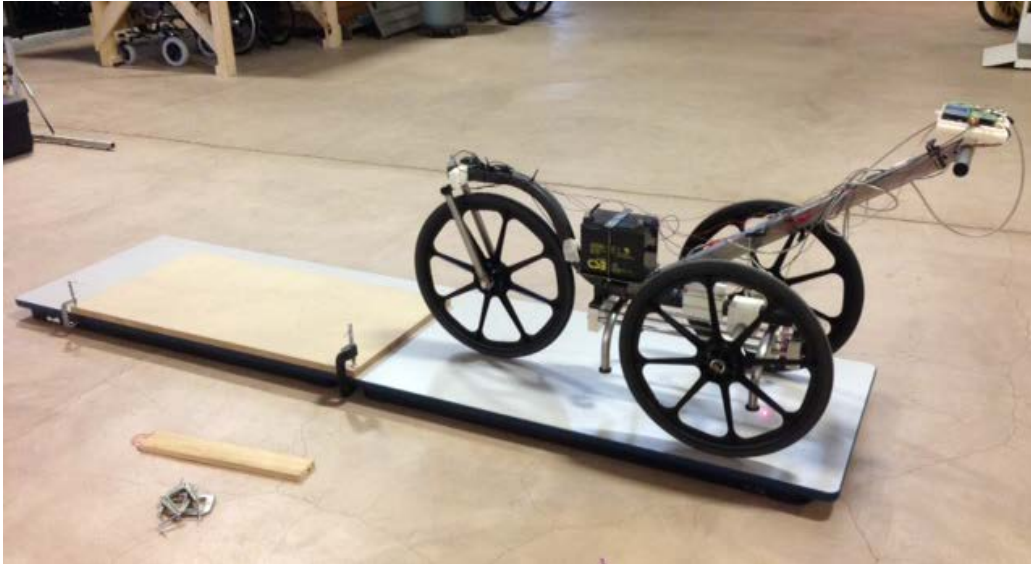
Figure 17: Level change characterization setup

Laser behind
back axle

Laser under
back axle

Laser in front
of back axle
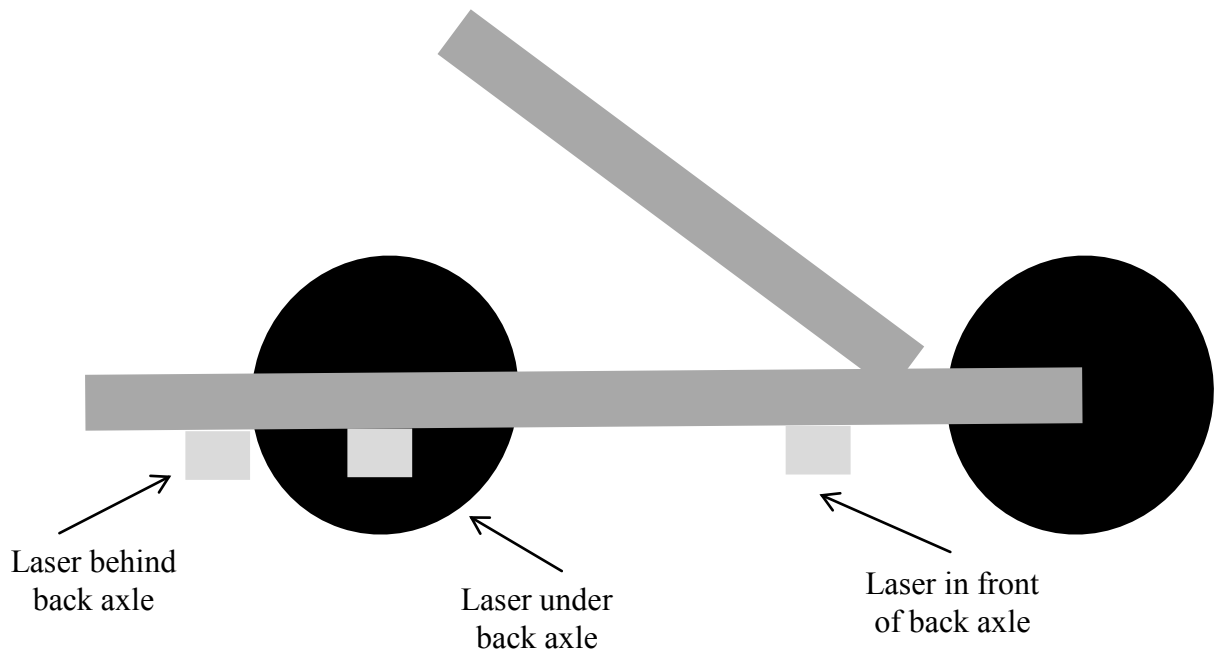
Figure 18: Three different locations of laser for level change testing

2.7 COMMUNITY-BASED DATA COLLECTION

Data collection in the community occurred at numerous locations in the United States. PathMeT was manually propelled (Figure 19) at approximately 1.0 m/s by a user who walked behind the system. Only Rolling Mode (discussed below) was used during data collection. The user surveyed each surface by measuring three paths, all of which exceeded 30.5 meters in length. The surfaces varied in design, including concrete, asphalt, and pavers. After data collection was complete, the PRI was calculated for each path along a specific surface, and the three PRIs were averaged to get the final PRI for that surface.

Figure 19: Data collection during Rolling Mode

2.8 STAKEHOLDER INTERVIEWS

Nine interviews with potential PathMeT stakeholders were conducted to better understand its commercial need. Table 10 presents sample questions that were discussed during conversation. According to Steve Blank, a "startup is a series of untested hypotheses" (Blank & Dorf, 2012). These hypotheses only become facts by "getting out of the building" and talking with customers and stakeholders. If one wants to commercialize technology, he/she must fully understand the needs of the customer. The current "pains" of the customer must be recognized, and a resulting "gain" should be obtained through use of the new technology. To facilitate this process, it is good practice to develop a *business model canvas* (Osterwalder & Pigneur, 2010), where one tests the "Product/Market Fit" relationship between the value proposition and customer segments. (Blank & Dorf, 2012).

Table 10: Sample questions used during stakeholder interviews

What are the strengths of PathMeT?
Which criteria or design features are the most important?
What is the current process to analyze/install sidewalks?
Who currently inspects sidewalks?
How are sidewalks maintained?
What tools/devices are used to analyze sidewalks?

3.0   RESULTS

3.1 PATHMET DESIGN

3.1.1   Electronics Design

PathMeT is composed of sensors (Table 11) integrated into one embedded design. Two of the sensors include a Riftek RF603 laser displacement measurement tool capable of measuring up to 9.4kHz and an S5 optical shaft encoder. The laser device is oriented perpendicular to the ground and, using a triangulation technique and trigonometry, measures the distance to the ground. The encoder measures the distance travelled by PathMeT. Together, the laser and encoder data provide a profile of the measured surface.

Table 11: List of PathMeT sensors

Sensor	Manufacturer/Model
Laser	Riftek RF603
Encoder	US Digital S5 Optical Shaft Encoder, S5-360-250-IE-S-B
Inclinometer	US DigitalX3M Multi-Axis Absolute MEMS Inclinometer
Accelerometer	Freescale MMA7260Q
Camera	RobotShop Color JPEG Camera w/ Infrared, RB-Lin-48
GPS	Sparkfun Venus GPS Logger

All sensors are integrated into a customized electronics board (Figure 20) in order to collect data. The Bill of Materials to produce this board can found in C.1. Altium Designer was used to design and develop this custom printed circuit board. Schematics of the sensor layout

and the associated connections are shown in C.2. There is a thin-film-transistor (TFT) touchscreen display that acts as the interface between the user and the sensors. The TFT displays a graph of the profile during data collection for real-time feedback. Data processing is done through the use of one dsPIC33EP512MU810, three dsPIC33EP128MC microcontrollers, and two microSD cards are used for data collection. MikroC programming language was used for coding these four microcontrollers. The source code can be found in A.2.
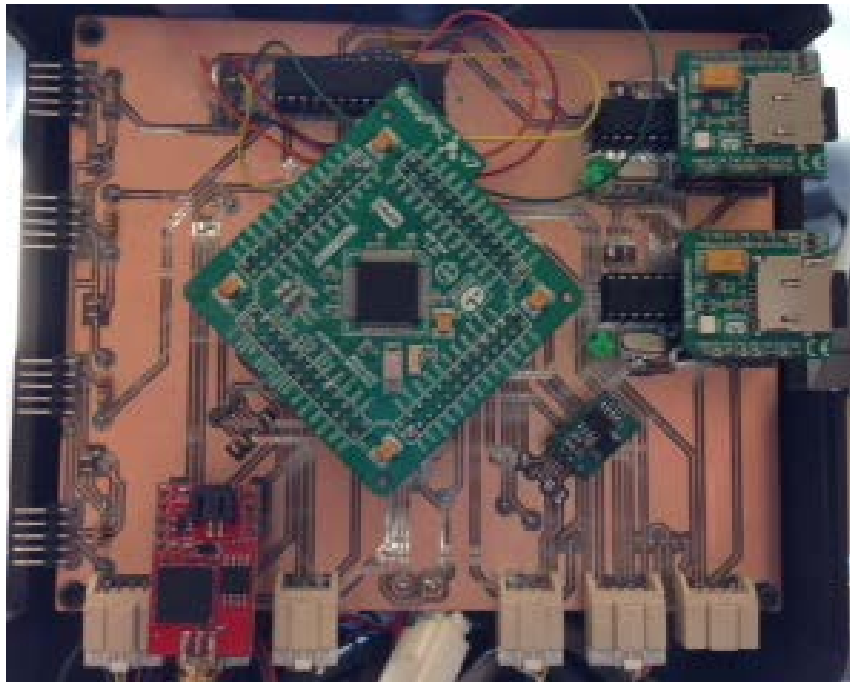
Figure 20: PathMeT printed circuit board

Data is collected via serial communication between the microcontrollers and the sensors. The microcontrollers collect one byte of data at a time, alternating between the laser and encoder. This ensures that all data is collected by a one-to-one ratio between the two sensors. A time stamp is recorded with every byte to ensure accurate timing. The microcontroller collects on average, but no less than, one laser and encoder reading every millisecond. This is based on a speed of 1.0 m/s ±10%, which is recommended propulsion speed. This speed was selected as a common walking speed and also the average speed of wheelchair users (Control, 2013; Cooper

et al., 2002). If more than one reading per millisecond is recorded, the data is down sampled by averaging the numbers for that specific millisecond. Therefore, the data is collected at 1000Hz sampling rate, resulting in 1mm resolution.

3.1.2   Mechanical Design
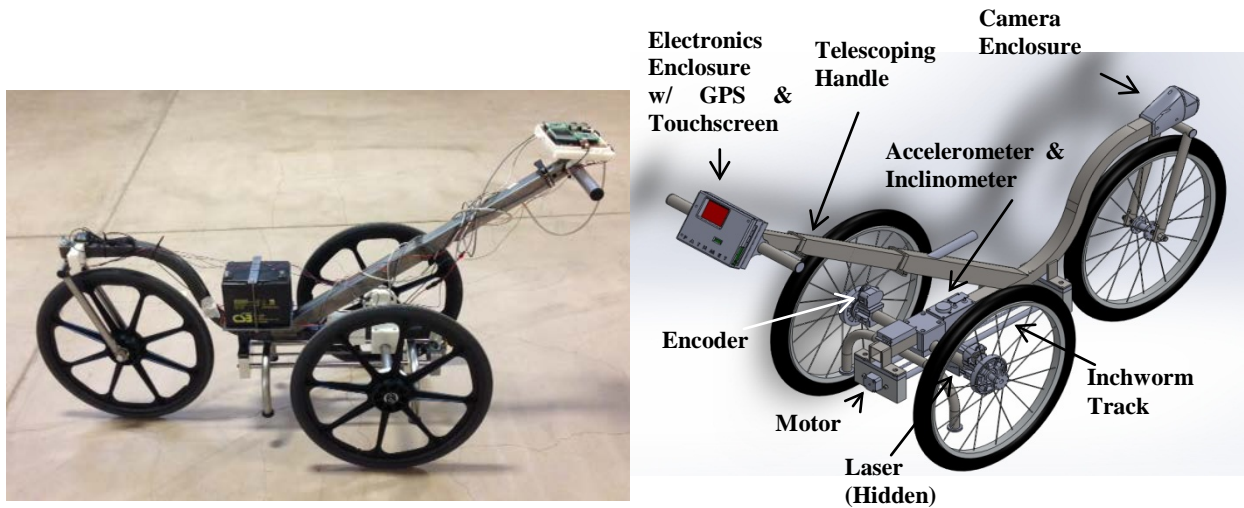
Figure 21: Pictures of the inside of PathMeT

The mechanical design of PathMeT, shown in Figure 21, includes a square tube steel frame with three 560 mm solid wheelchair tires. Solid tires were selected to eliminate sources of error that might be experienced with pneumatic tires through a variation in tire pressure. The front wheel is a caster, allowing for PathMeT to make turns easily. All three wheels can be quickly removed for increased portability. Furthermore, a twice-telescoping adjustable handle bar extends out to the user for increased comfort. Figure 22 shows a completed PathMeT with enclosures included. Appendix D shows an additional 3D model of PathMeT, a Bill of Materials for all mechanical hardware and sensors, and mechanical drawings. Enclosures and circuit board

components are not included in this bill of materials. An early design brief of PathMeT can be seen in Appendix E.
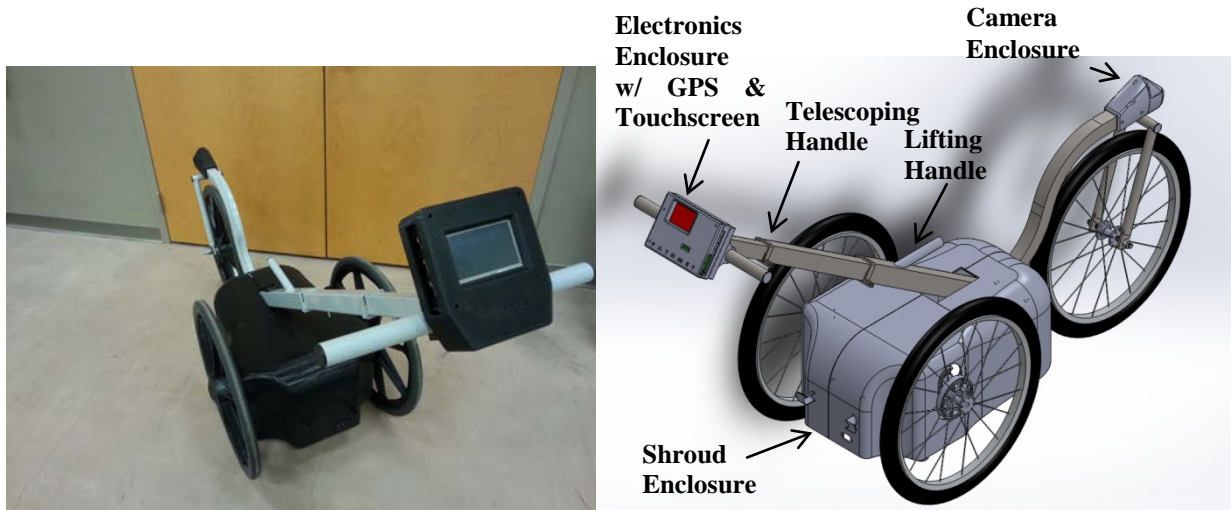
Figure 22: Pictures of PathMeT

3.1.3 Interface Design

Figure 23 shows the user interface for PathMeT and the step-by-step process for data collection. The interface was programmed using VisualTFT programming language. A.2 shows the source code. Step 1 shows the display upon powering the device. When ready, the user presses the "Initialize" button to initialize the system. Step 2 shows the loading bar after Step 1. This loading screen takes twenty seconds, allowing the laser to warm up and initialize. Step 3 shows that the system is waiting for a command. When ready, the user presses "Start sampling" to ensure that all the sensors are sampling properly. Step 4 shows PathMeT in "sampling mode". In order to progress, the user first selects "Stop sampling," then "Logging Mode." Before data can be collected, the user creates a folder in which to save data for that specific data collection run. The user does this by selecting "Create a Folder" (Step 5), entering a file name up to six characters and pressing "Confirm." Data collection begins in Step 7 when the user selects either Rolling

36

Mode (RM) (shown in the picture) or Inchworm Mode (IW), presses "Start Profiling" and begins propelling the device, if in Rolling Mode. Step 8 shows the four instantaneous outputs that are displayed: A) Running Slope, B) Profile of the surface, C) Speed, D) Cross Slope. When the user is propelling PathMeT at an appropriate speed, the speedometer is green; otherwise, the speedometer is red. When data collection is complete, the user presses the "Stop Profiling" button. Step 9 shows that the camera must finish transferring data before proceeding to the next run. Finally, Step 10 shows a summary of the run, including time elapsed, distance travelled, file name, last instantaneous speed, and average speed. The user can then select "new profile" to begin a new run, "About" to learn about the previous run, or "Help" for help in proceeding. These are the steps required for data collection.
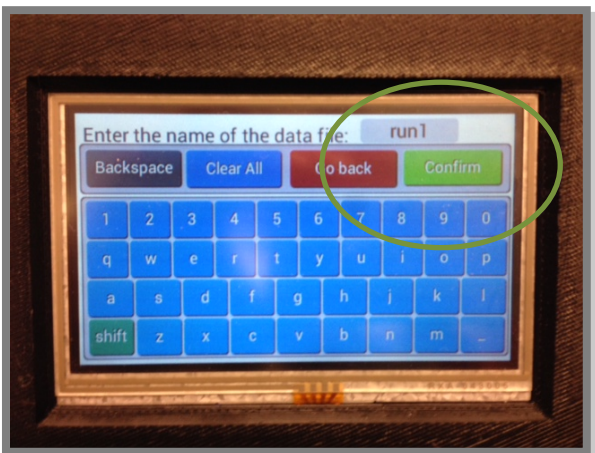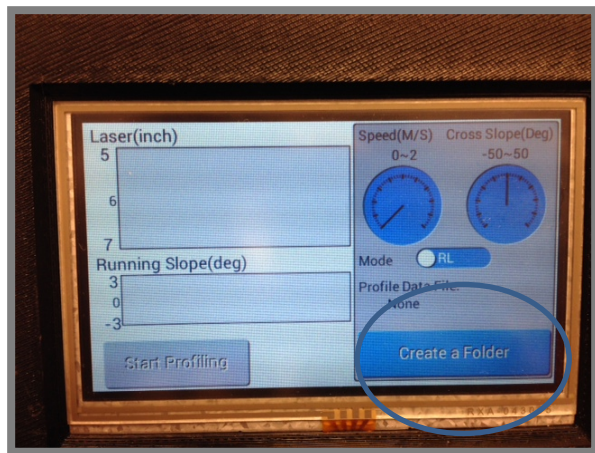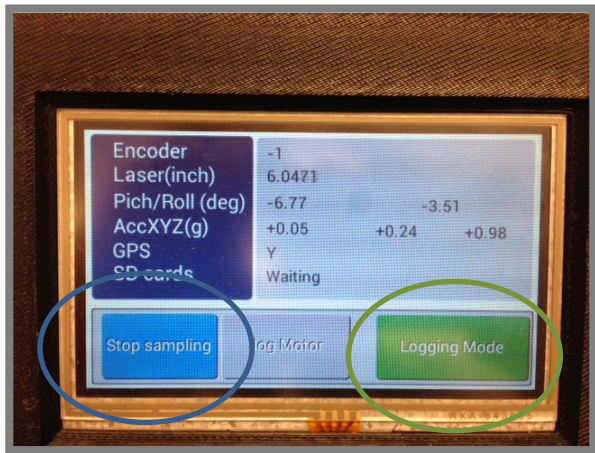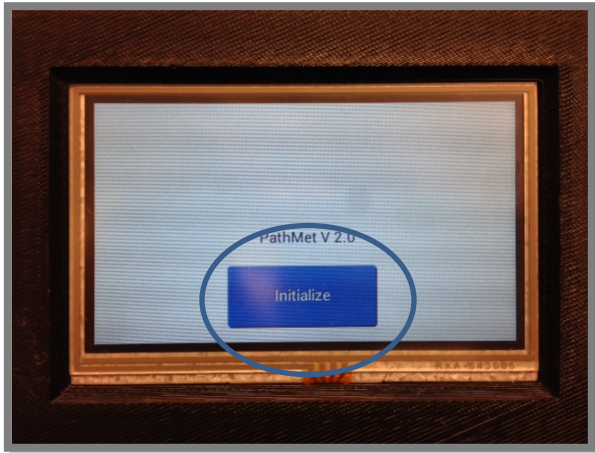
Figure 23: PathMeT interface
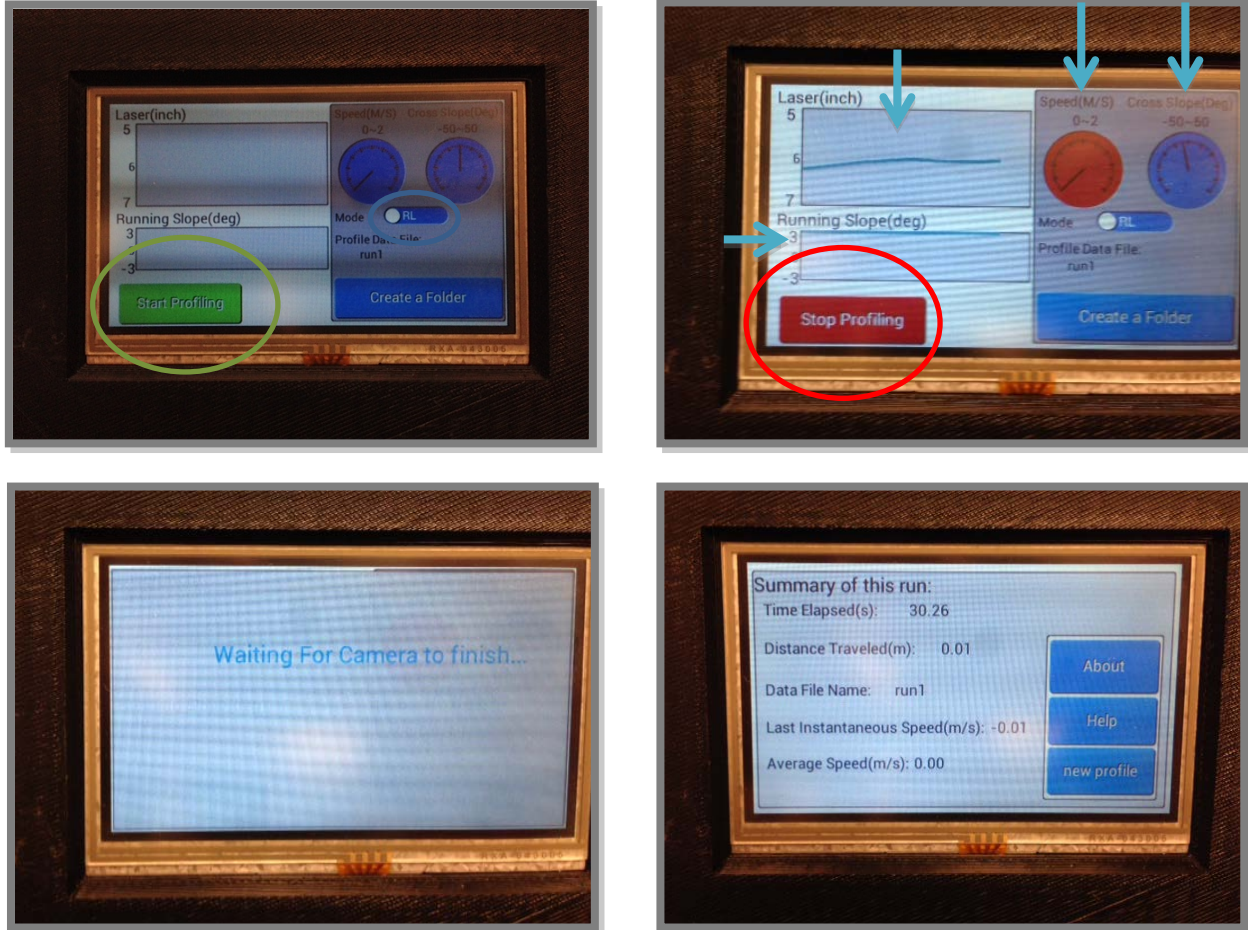
Figure 23 (continued)

3.1.3.1 PathMeT Modes

3.1.3.2 Rolling Mode

Rolling Mode allows the user to push PathMeT continuously at a speed of 1.0 m/s ± 10%. This speed has been chosen in order to ensure 1mm resolution and to be consistent with typical walking and wheelchair speeds. This mode allows for a large amount of data to be collected in a relatively short period of time. Although data collection during Rolling Mode has been implemented, additional work is needed in order to execute the following: While data is continuously collected, the user is alerted of pathway segments that fall out of compliance with

standards or are rough enough to introduce errors to the continuous data collection; these data from these segments will need to be recollected using the Inchworm Mode outlined below. When PathMeT is stopped, each rough segment is displayed in a queue. The queue displays the distance the user must backtrack in order to recollect the data for each segment. The user can either decide to recollect the data or ignore the error.

Figure 24 shows a flow chart for operation within the Rolling Mode. If the user decides to recollect that data, he/she will need to backtrack the original path travelled. As the user moves backwards, the odometer counts down until it read "0 mm" for the first error listed. The errors are ordered from newest to oldest. The user then applies the break and runs the Inchworm Mode (discussed below). Then, the user has the option to ignore the next error, measure the next flagged segment, or continue data collection as normal. This process is repeated for each segment that displayed an error in the queue.
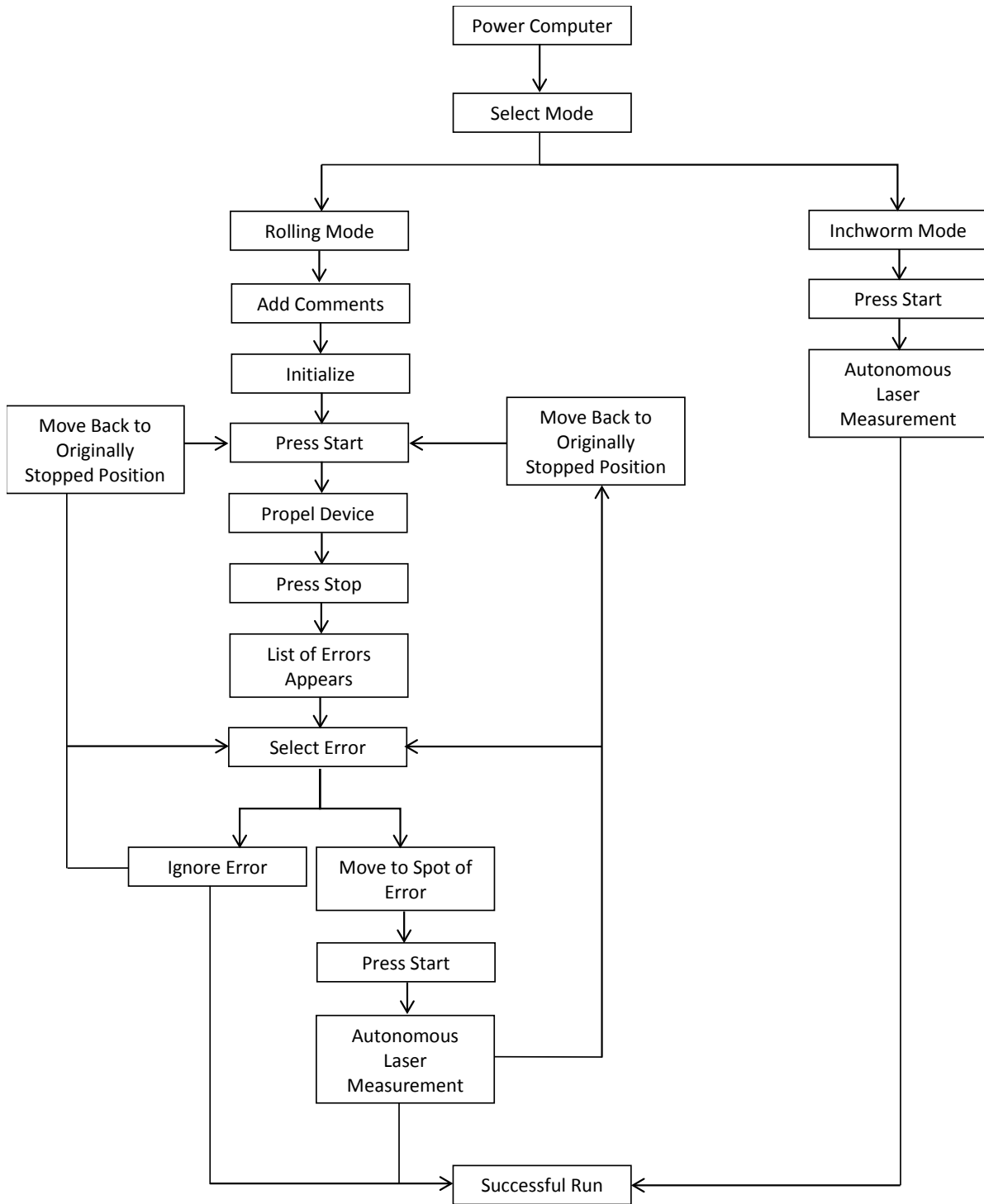
Figure 24: Flow chart of user operation within each mode

3.1.3.3 Inchworm Mode

PathMeT also has the ability to collect data via a method referred to as Inchworm Mode. Figure 24 shows the flow chart for Inchworm Mode operation. In Inchworm Mode, PathMeT remains stationary after the user places it over the area of the surface where data is to be collected. While inchworm mode is engaged, the user holds PathMeT stable. The user presses a button to begin data collection, and PathMeT's motor-driven laser moves along a 500 mm track collecting data for the area of the surface immediately below it (Figure 25).
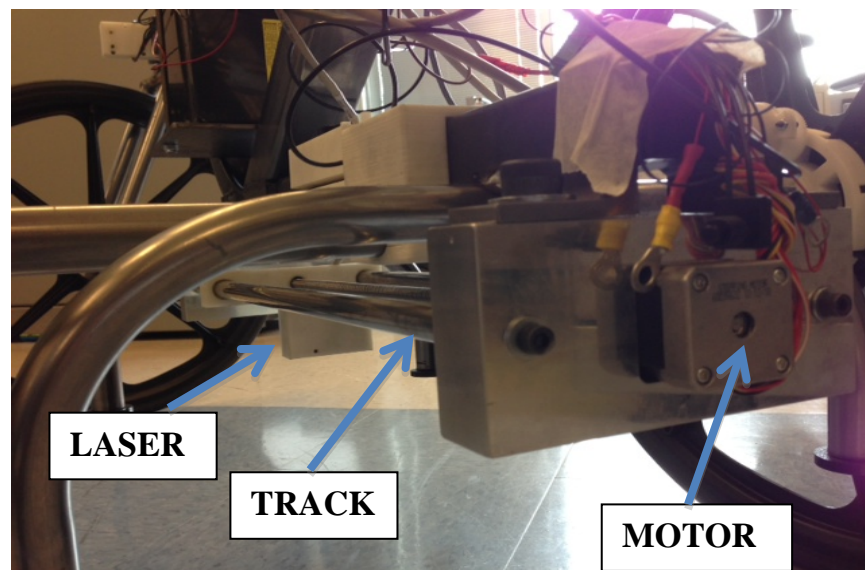
Figure 25: Inchworm assembly

For example, consider a 3-meter length of surface that needs to be measured. In order to collect data using Rolling Mode, the user would manually propel PathMeT for a total of three meters and data collection would be complete. To collect data the same data in Inchworm Mode, the user would need to position PathMeT for six successive measurements since the laser only moves on a 500 mm track. Thus, the user would place PathMeT over the first 500 mm of the 3-meter section, hold PathMeT stationary while that 500 mm section is measured, then place PathMeT over the next 500 mm and repeating until data for all three meters is collected.

3.1.4 PathMeT Specifications

Table 12 shows a list of the final specifications for the latest PathMeT design.

Table 12: Current PathMeT specifications

PathMeT Specifications
1. Material Type: Steel
2. Weight (Disassembled without wheels): 31.3 kilograms
3. Target Weight (Assembled): 38.1 kilograms
4. Physical Dimensions (Disassembled with push-handle collapsed) (mm):   1181L x 457W x 584H
5. Physical Dimensions (Assembled) (mm): 75.5"L x 23.5"W x 42.5"H
6. Battery Life: Undetermined

3.2 TESTING

3.2.1 Reliability Testing

Table 13 shows the results of the testing protocol for intra- and inter-rater reliability. Each cell represents the average PRI for each user on a particular surface. The standard deviation within the trials is displayed, with the engineered surface showing the largest variance in standard deviation between users. After performing IBM SPSS Statistics analysis for intraclass correlation, results show a 0.993 intraclass correlation of average measures with a 95% confidence interval of [0.976, 0.998]. Similar analysis shows a 0.979 intraclass correlation of single measures with a 95% confidence interval of [0.932, 0.995]. In addition, an inter-item correlation results in a 0.983 mean with 0.976 and 0.997 minimum and maximum values, respectively. Finally, SPSS analysis presents Cronbach's alpha equal to 0.993.

Table 13: Average PRI (with standard deviation) of three 4.9-meter surfaces by three users

	User 1 Avg (Std Dev)	User 2 Avg (Std Dev)	User 3 Avg (Std Dev)
Engineered	0.92 (0.01)	0.99 (0.13)	0.94 (0.06)
Concrete	0.39 (0.02)	0.39 (0.02)	0.40 (0.04)
Stamped Concrete	0.47 (0.05)	0.46 (0.03)	0.44 (0.04)

3.2.2   Step Function Testing

Level-change testing results can be seen in Figure 26. The Figure shows the effects of different laser placements when propelling over a 25.4 mm step. The line labeled *Theoretical* is the actual profile of the step. The lines labeled *After*, *Under*, and *Before* show profiles of the step when the laser is placed behind, under, and in front of the back axle, respectively.

The *After* plot illustrates that when the front wheel hits the step, the laser moves closer to the ground giving the illusion that there is a small bump. Then, the back wheel reaches the step raising the laser 25.4 mm higher than the original starting position. This makes the profile appear as if PathMeT has experienced a gap. However, the laser has not experienced the step until the final vertical line that brings the profile back to its original height.

The plot labeled *Under* shows that this placement similarly affects laser data, when the wheels reach the step. On the other hand, when the laser is placed in front of the axle, the resulting profile makes the position of the 25.4 mm step clear. In addition, there are two 12.7 mm steps, one before and after the step, which correspond to the front and back wheel ascending the step.
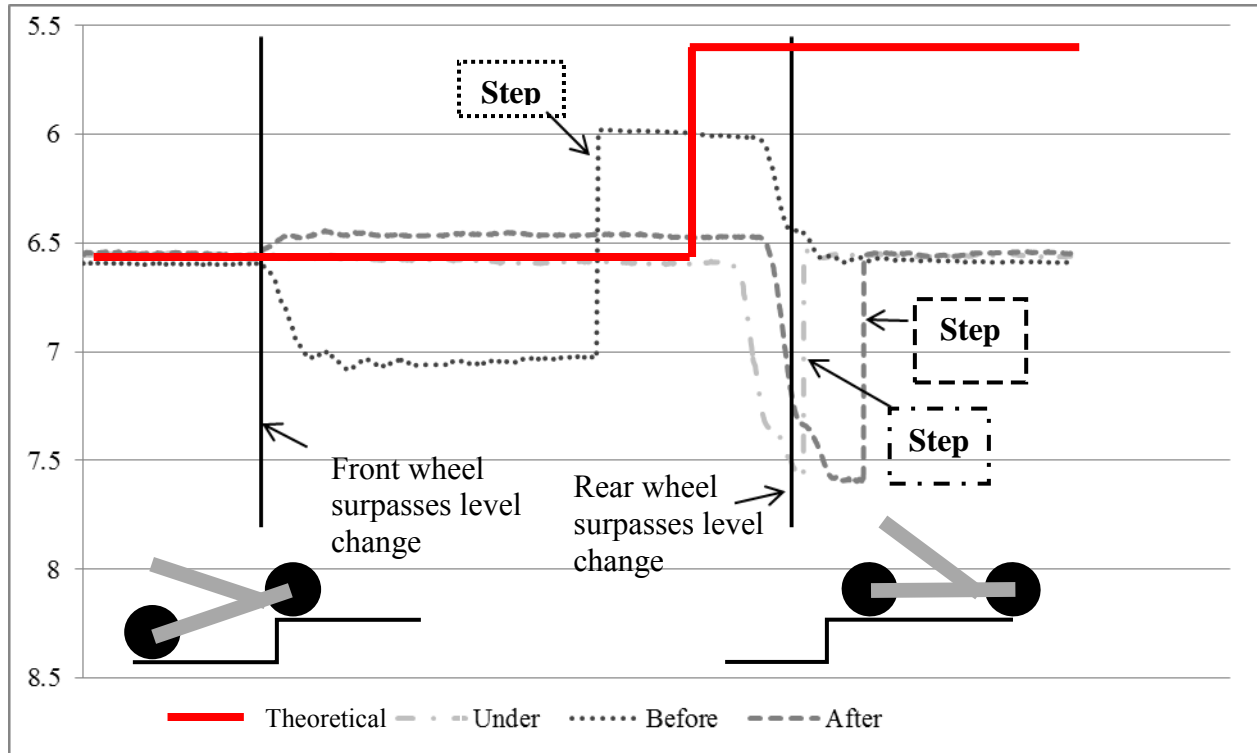
Figure 26: Profile results of a 25 mm level change at three laser locations

3.3 DATA COLLECTION

3.3.1 Basic Data Collection & Processing

Figure 27 shows the results of data collection on Surface C1. The red line represents the profile determined by P1 and the black line represents the wheel path as determined by the wheel path algorithm. One can see how the wheel path algorithm is not contingent upon the depth of the cracks in the profile. This is due to the fact that the type of wheel used in the wheel path algorithm analysis has a large enough diameter that it will roll over these deep cracks without falling all the way in the crack.
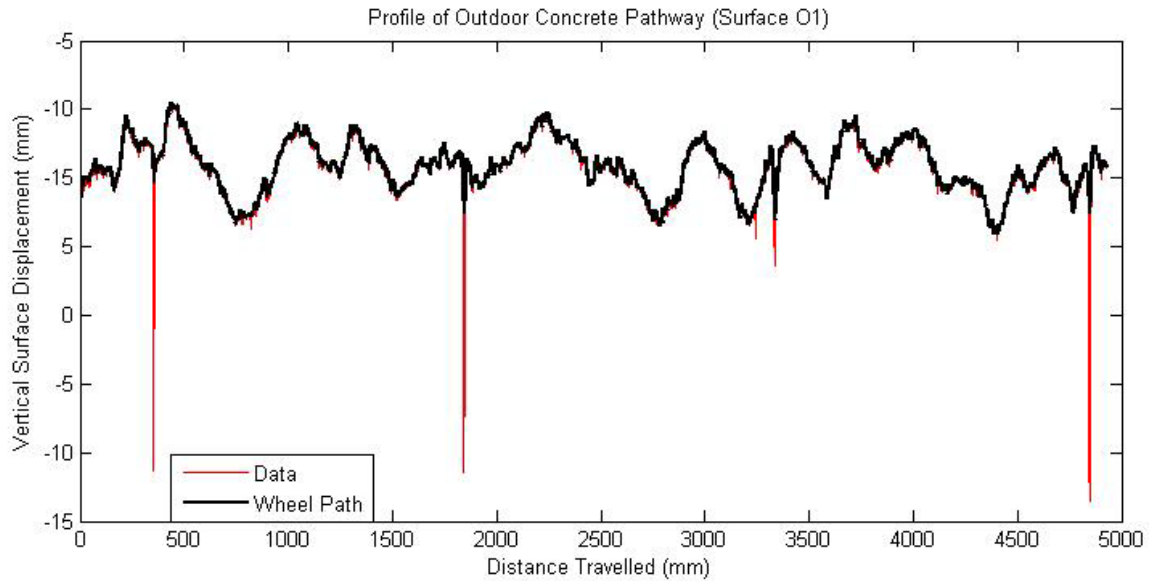
Figure 27: Profile of surface C1

Table 14 lists the roughness indices for the six different community surfaces when measured by P2 without plywood tracks. Table 15 shows roughness indices for the nine different engineered wooden surfaces, comparing P2 with and without plywood tracks. A linear regression of the P2 results in Table 15 shows a significant correlation between the two different techniques with an R^2 value of 0.99.

Table 14: PRI for community-based
surfaces measured by P2 without plywood tracks

Surface ID	P2 w/out Plywood Tracks (mm/m)
C1	35.00
C2	41.67
C3	70.00
C4	95.00
C5	104.16
C6	165.83

Table 15: Comparison of PRI for engineered wooden surfaces collected by P2 with and without plywood tracks

Surface ID	P2 w/out Plywood Tracks (mm/m)	P2 w/ Plywood Tracks (mm/m)	Percent Difference (%)
E1	22.50	13.33	49
E2	35.00	24.17	36
E3	39.17	25.83	40
E4	60.00	35.83	50
E5	60.83	38.33	45
E6	70.00	53.33	27
E7	88.33	69.16	25
E8	118.33	93.33	24
E9	148.33	119.16	22

3.3.2   Roughness

3.3.2.1 Engineered

Roughness results from data collection along the engineered surfaces are shown in Table 16. The surfaces are ordered by increasing roughness. The P1 results shown are due to driving P1 over the tracks, described above, in order to eliminate vibration errors. A linear regression (Figure 28) was conducted to examine the linearity between the two sets of results; an R^2 value of 0.99 resulted. This shows that there is a correlation between the P1 and PathMeT PRI results. Consequently, by knowing PathMeT PRI, one can determine the PRI calculated by P1. This allows for a PRI that is comparable with the roughness standard, since the standard was developed using P1.

Table 16: Comparison PRI between P1 and PathMeT

Surface ID	P1 Roughness (mm/m)	PathMeT Roughness (mm/m)	Crack Frequency	Crack Width (mm)
E1	16.67	19.17	No cracks	0
E2	24.17	36.67	12	20
E3	30.00	47.50	8	20
E4	44.16	57.50	12	32
E5	44.16	66.66	4	20
E6	55.00	78.33	8	32
E7	70.00	98.33	8	39
E8	91.66	127.49	4	32
E9	113.33	167.49	8	51
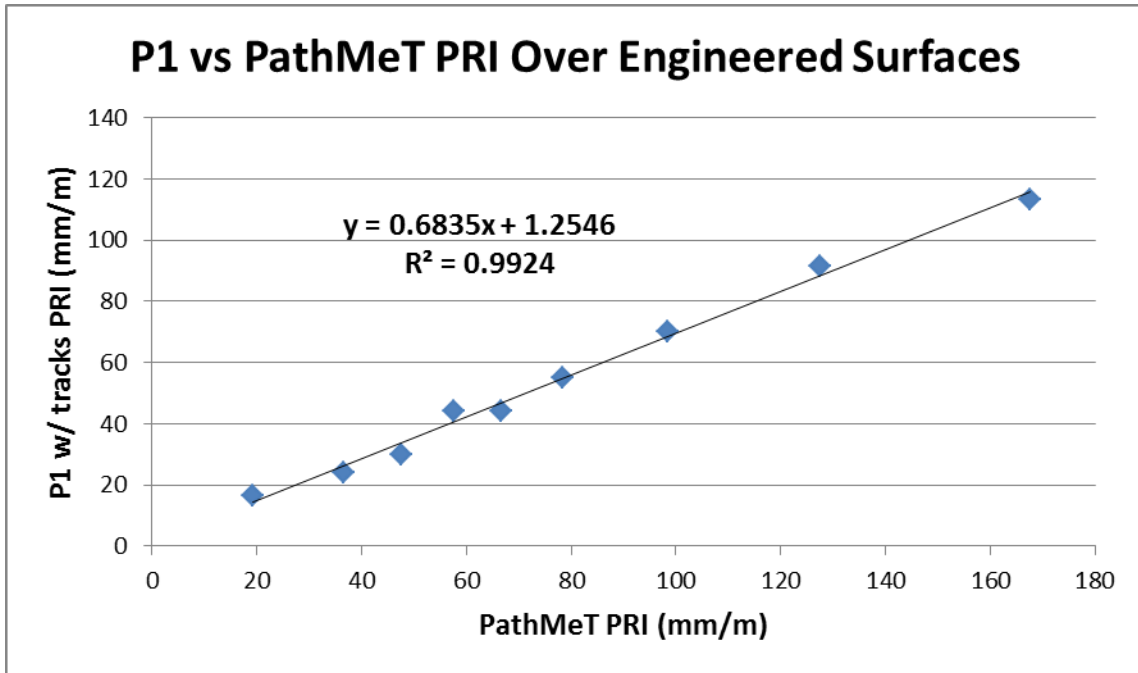
Figure 28: Linear regression of P1 and PathMeT PRI

3.3.2.2 Community

PathMeT was used to collect data from various different pedestrian pathway surfaces in the community. It was manually propelled in Rolling Mode, collecting various characteristics of the surfaces, including roughness, level change, GPS location, and picture. Table 17 shows the locations where data were collected; the image and PRI of the surfaces are listed. A profile of portion of Surface 5 is shown in Figure 29.

According to the current threshold for roughness proposed to the Access Board, PRI should not exceed 50 mm/m for distances of 30.5 meters or along what would be considered the accessible route. For distances less than 3 meters, the proposed threshold is 100 mm/m. Although the calculation of PRI is the same for all distances, the threshold varies because of the increased risk due to vibration exposure from travelling over rough surfaces for long distances. A method to distinguish PRI by varying distances is to have a moving "window" that calculates PRI every 3 meters, then every 30.5 meters. This would facilitate the distinction between long- and short-range measurements; however, no such method has been developed. Therefore, for this paper PRI below the proposed 50 mm/m threshold are considered compliant (highlighted in green), surfaces with PRI between 50 and 100 mm/m are considered moderately compliant (highlighted in yellow), and surfaces with PRI exceeding the 100 mm/m threshold are considered non-compliant (highlighted in red). Notice that Surfaces 11 and 12 were measured using the P1 prototype.

Table 17: Community-based surfaces with image and PRI
**Data was collected with P1

Surface ID	Picture	PRI (mm/m)	Surface ID	Picture	PRI (mm/m)
1		30.67	2		42.41
3		45.50	4		55.08
5		62.16	6		64.25
7		74.08	8		81.75
9		82.91	10		85.08
11**		115.00	12**		160.83
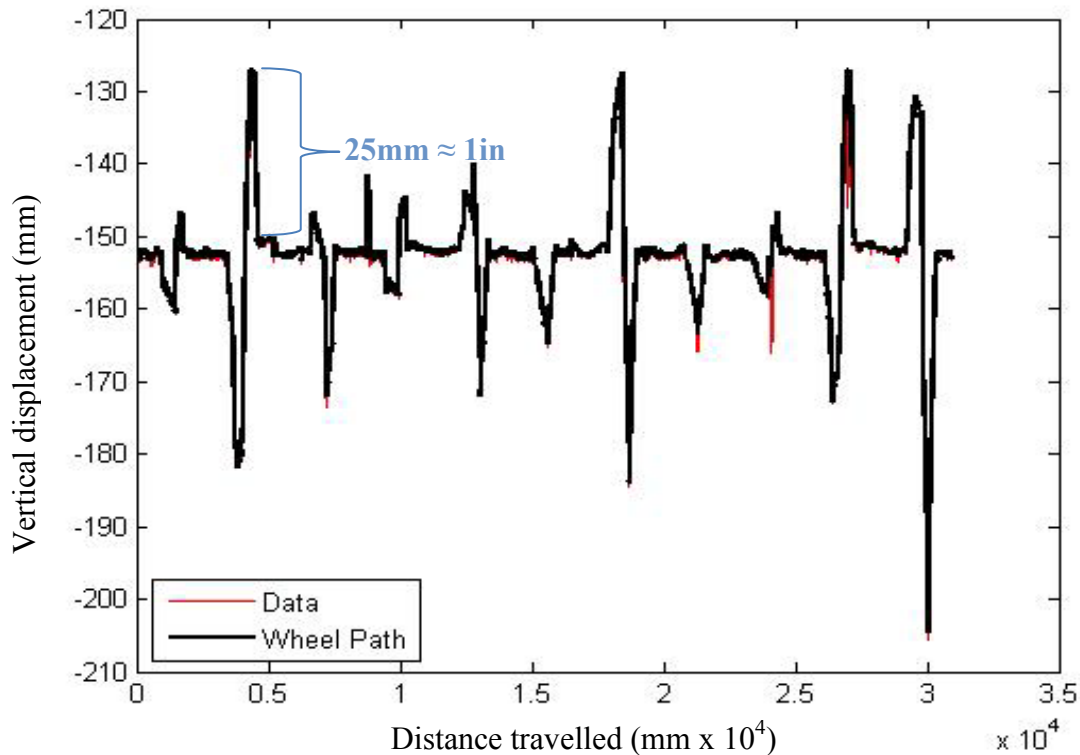
Figure 29: Profile of community Surface 5

3.3.3 Running Slope

During data collection, PathMeT analyzed slope by use of the inclinometer and accelerometer. As described above, maximum running slope (rise:run) is 1:12 (4.76°) and 1:16 (3.58°) for pathway ramps 9.1 and 12.2-meters long, respectively. Figure 30 and Figure 31 show plots of the running slope for PathMeT with inclinometer data in blue and the calculated pitch from the accelerometer in pink. The inclinometer plot was a result of performing a Butterworth filter on the raw inclinometer data shown in red. Figure 30 shows graphs from traversing engineered Surface E1, a flat surface, while Figure 31 results from analyzing engineered Surface E2. Both graphs present an initial flat line, which shows when PathMeT did not move. Once, PathMeT

51

begins its linear acceleration, there is a jump in the graphs to approximately three degrees, even though the PathMeT device remains relatively level. For a short period of time, PathMeT remains at this angle until it begins to reach a constant velocity, where the slope begins to decrease. The slope steadily decreases until the end of each run.
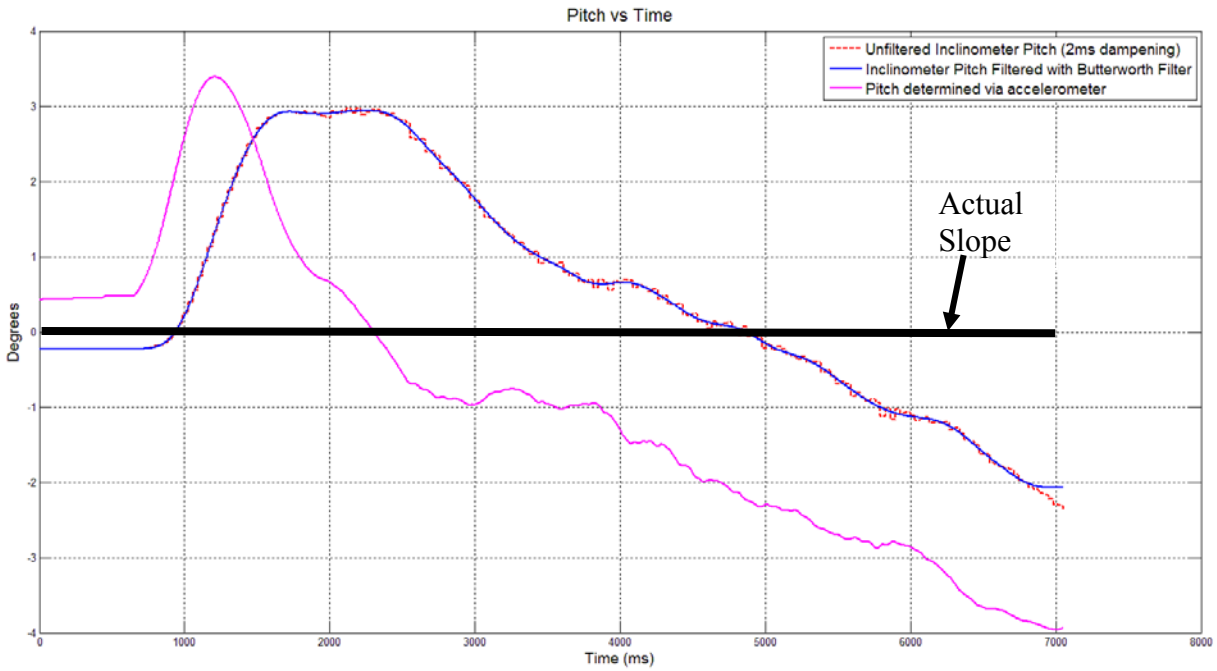
Figure 30: Running slope plot of engineered Surface 1

52
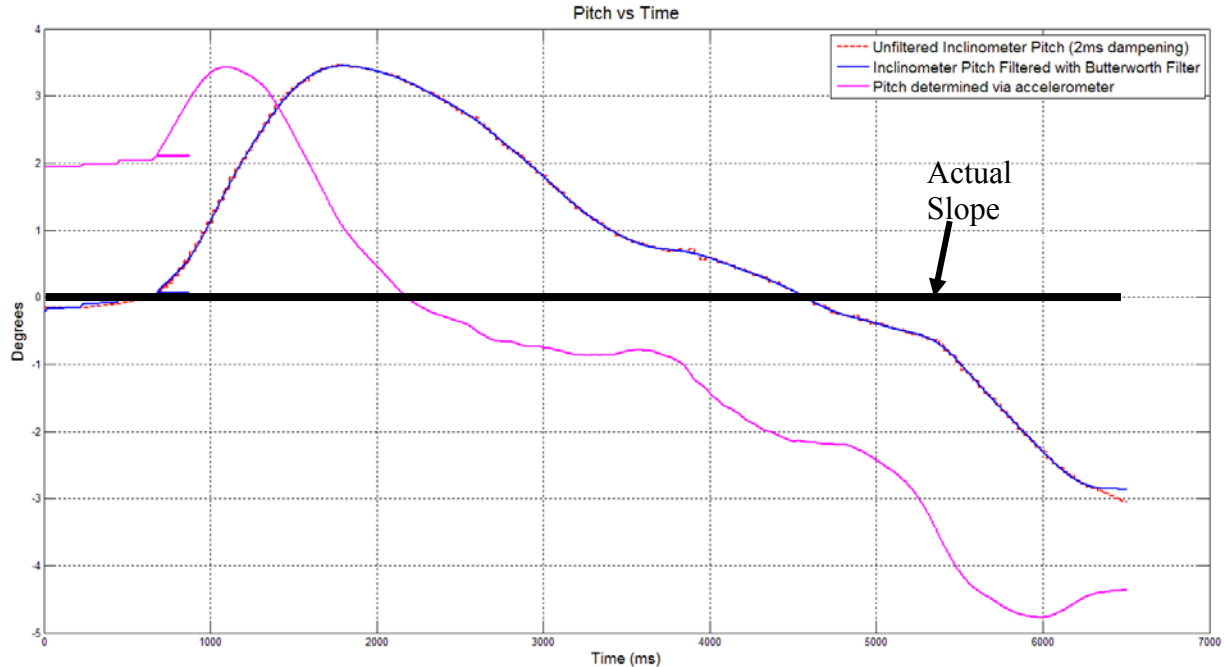
Figure 31: Running slope plot of engineered Surface 5

Now observe Figure 32 that depicts data collection from a level surface, to a ramp, and back to a level surface. The actual slope of the ramp varied between 2.2° and 2.8° at different segments, with an average slope of 2.5°. Again, a Butterworth filter was applied to the inclinometer data (resultant in blue) and slope was calculated using accelerometer data and trigonometry (pink). Figure 32 shows the various stages of data collection during this run. Upon initial propulsion of PathMeT on level ground, an increase in slope is shown due to linear acceleration. When approaching a constant velocity, the slope begins to approach zero slope and returns a relatively flat line. Once PathMeT reaches the start of the ramp, there is another gradual increase in slope, until it remains constant at approximately 3°. At this point, PathMeT is travelling at a constant velocity while completely on the ramp. Then, PathMeT reaches level ground at the top of the ramp, and the slope begins to approach zero. The slope remains at zero

until PathMeT is brought to a stopped position, where a negative slope results due to the deceleration.
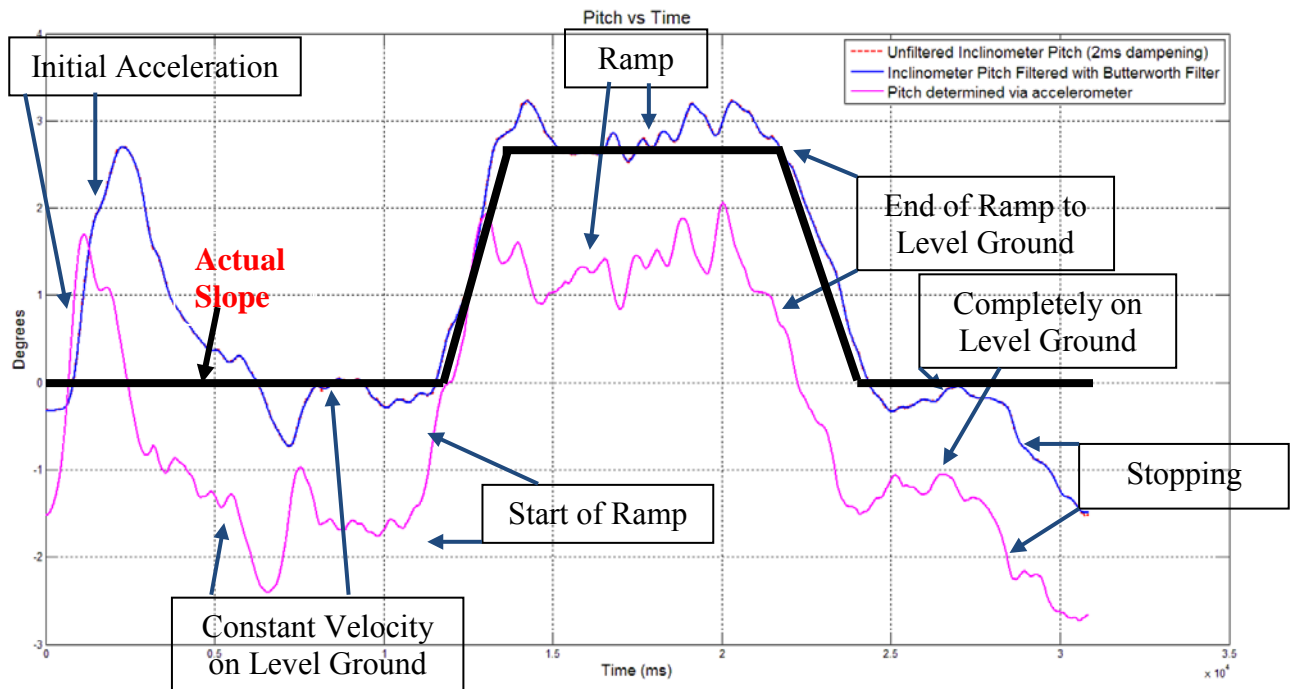
Figure 32: Running slope plot up a ramp

3.3.3.1 Google Earth Mapping

Furthermore, Figure 33 and Figure 34 show a Google Earth mapping of the surfaces with the same color coordination as Table 17. Each place mark has a picture linked with it that shows the specific area of the pathway (Figure 35). The source code for creating the .KML file to be used with Google Earth is shown in A.3.

Figure 33: Google Earth mapping of selected surfaces (Part 1)
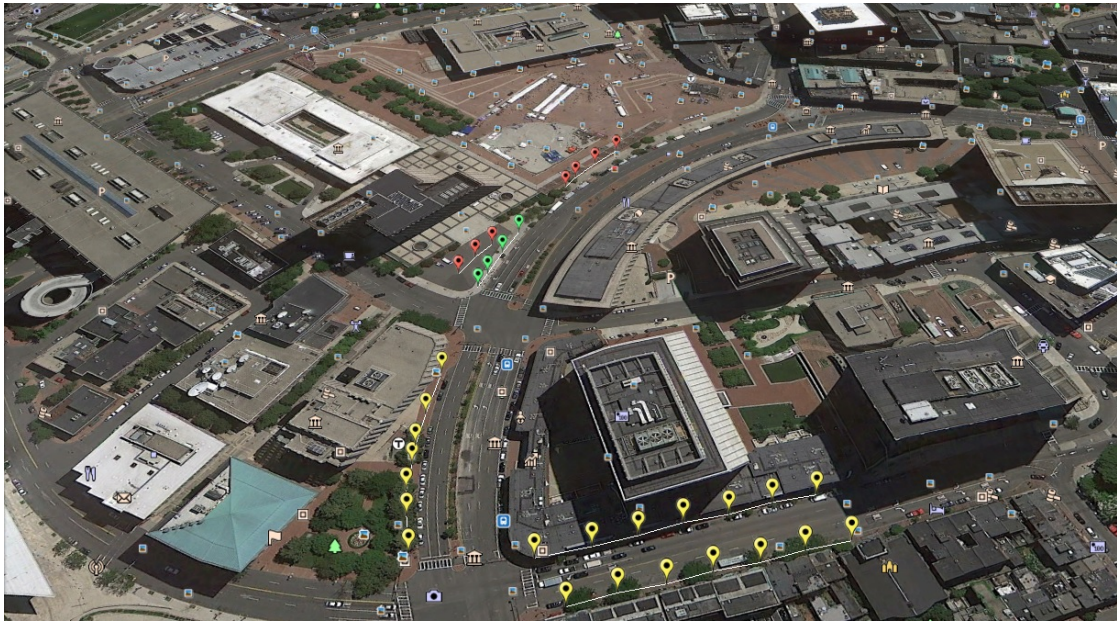
Figure 34: Google Earth mapping of selected surfaces (Part 2)
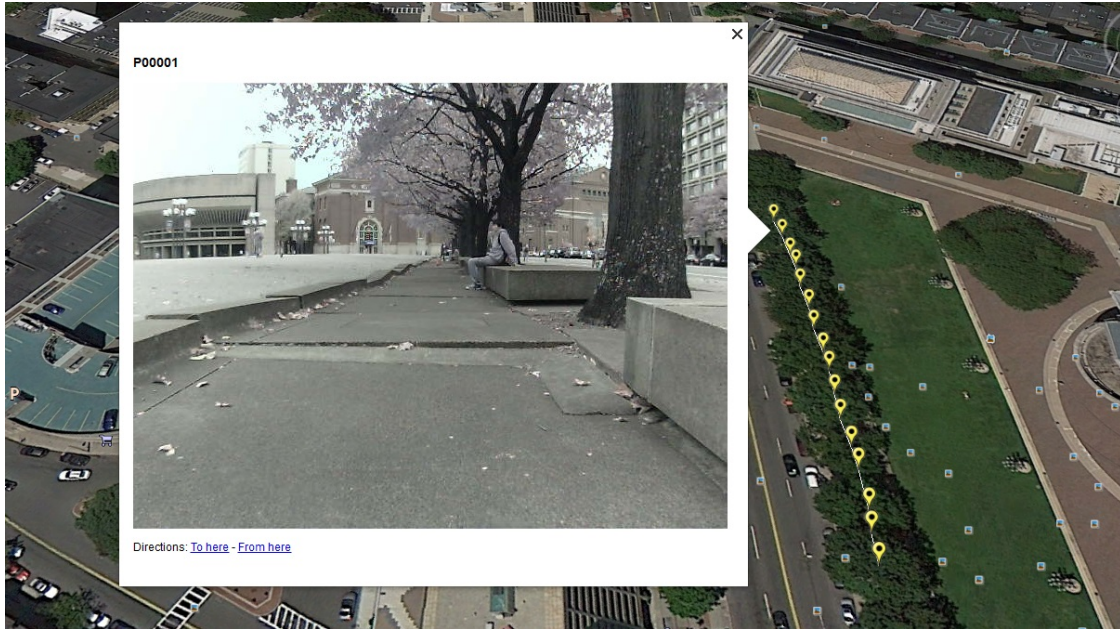
Figure 35: Placemarks and related picture of community-based Surface 5

Data collection at Surface 5 resulted in a PRI of 62.16 mm/m, averaged from five different runs over the surface. Although the PRI is in the cautioned range, the profile for Surface 5 indicates level changes of approximately 25 mm at four locations. These changes in level are not acceptable according to ADAAG. According to ADAAG, a change in level cannot exceed 6.35 mm, or 12.7 mm with bevel. The 25 mm change in level means that the surface does not comply with ADAAG at those four specific locations. Figure 36 shows the location where the first noncompliance occurs and correlates with the first spike shown in the Figure 29 profile. The level changes do not necessarily provide any other information regarding the level of compliance for the entire surface. However, if there were no other accessible routes, then due to the non-compliant level changes, it would be deemed that there is no accessible route.
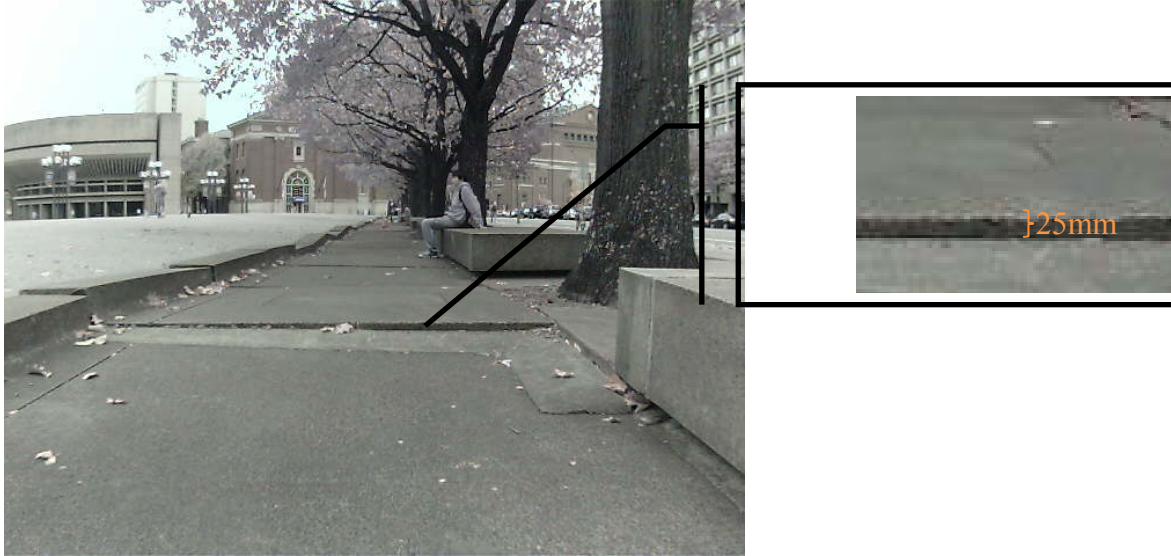
Figure 36: Image of community-based Surface 5 with level change enlarged

Among the data collected, comparisons between Surfaces 1 and 3 (Table 17) are particularly interesting. Surface 1, made of pavers, has a lower PRI than that of Surface 3, made of poured concrete. This data challenges frequent general assumptions that "continuous" surfaces are smoother than those with joints and shows that when designed, installed and maintained properly, a paver surface can be smoother than a poured concrete surface. This data alludes to the qualification that roughness is largely due to the quality of joints. Figure 37 shows a 25mm expansion gap from Surface 3, which results in an increase in roughness. The increased roughness for these surfaces appears to be a problem of maintenance and deterioration rather than design.
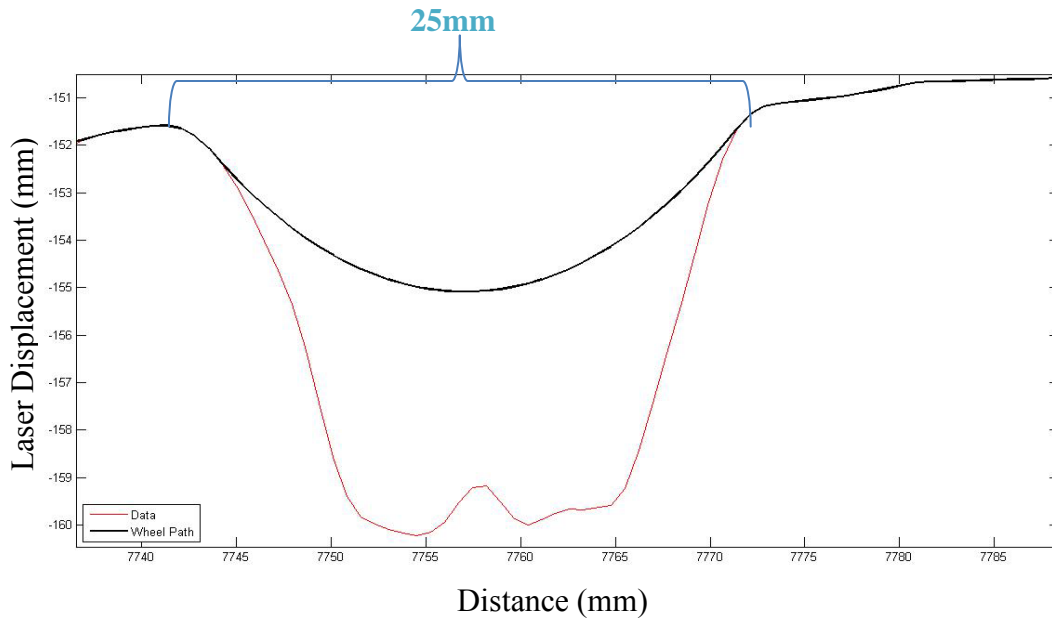
Figure 37: Enlarged profile of community-based Surface 3 expansion gap

Comparisons of the PRI of Surfaces 4 and 6 (Table 17) are also noteworthy. These two surfaces are on opposite sides of the street, generally composed of the same paver material. However, Surface 6 has concrete slabs mixed in with the pavers. The addition of a second type of material, although concrete, results in rougher surface than one consisting of pavers alone. This shows the importance of ensuring smooth transitions between two dissimilar surfaces. Although both are in the cautioned zone, Surface 6 has a PRI that is 8.33 mm/m greater than Surface 4.

Detailed analysis of Surface 8 (Figure 38) shows large gaps in between pavers. Figure 39a shows a profile of the surface, while Figure 39b shows an enlarged section of the graph. The periodic gaps between pavers can be seen in this image. Upon further enlargement, Figure 39c shows the size of one of the gaps. The result is a gap of 25mm. It appears as if a major contributor to the roughness of this surface is the large gap size.

Figure 38: Close-up of community-based Surface 8
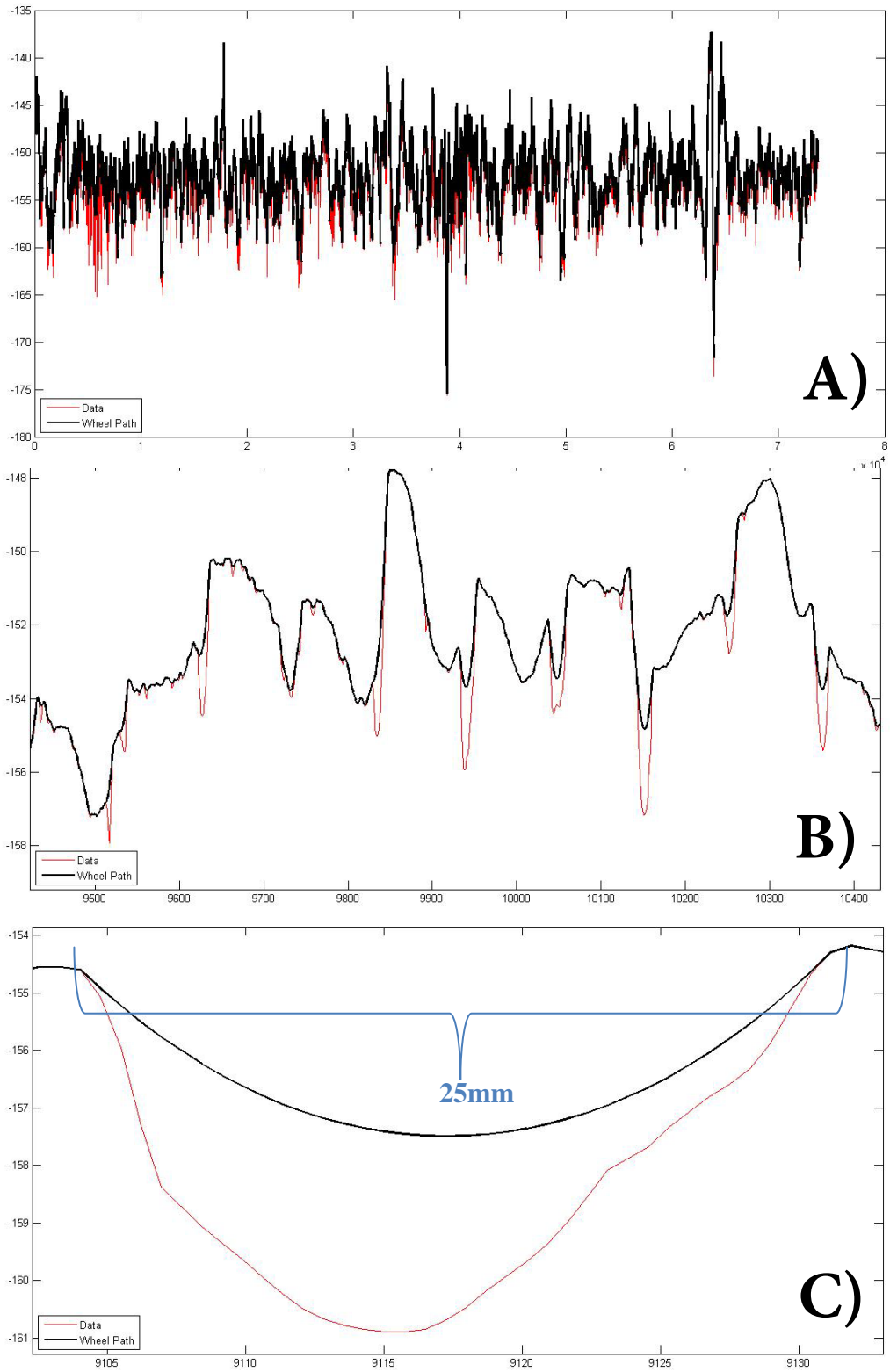
Figure 39: Profiles of community-based Surface 8

Surface 10, one of the roughest of the surfaces measured, is composed of a broken asphalt surface. Portions of the surface remained intact, but many potholes were observed, causing the increased roughness.

3.4 TARGET VS. PATHMET SPECIFICATIONS

Table 18 shows the comparison between the target and current specifications for PathMeT. Although the weight and dimension specifications are not within the target specification range, improvement is possible. The weight is more than anticipated due to the use of steel. For ease in prototyping, steel was used so that it can be welded. The use of aluminum would drastically decrease the weight of PathMeT, enhancing its portability. However, testing would be needed to ensure accuracy and reliability of an aluminum frame.

Table 18: Comparison of target and current PathMeT specifications

	Target Specifications	PathMeT Specifications
Weight (Disassembled without wheels)	22.7 kilograms	31.3 kilograms
Weight (Assembled)	29.5 kilograms	38.1 kilograms
Physical Dimensions (Disassembled with push handle collapsed) (mm)	1016L x 635W x 508H	1181L x 457W x 584H
Dimensions (Assembled) (mm)	1524L x 635W x 1219H	1917L x 597W x 1080H
Battery Life	8 hours	Undetermined

3.5 STAKEHOLDER INTERVIEWS

Table 19 shows the results from nine interviews conducted with potential customers and stakeholders. The first column shows the type of person interviewed, while column two presents the hypothesis tested. The main takeaways from each interview are listed in the third column. Any identifying information, such as name or company, has been removed from to assure interviewee confidentiality. Figure 40 presents a business model canvas with the current PathMeT value propositions and customer segments completed.

Table 19: Interview summaries with tested hypotheses

Interview Type	Hypothesis Tested	Interview Summary
Customer (Walnut Capital)	Business owners would be interested in purchasing a certificate of compliance	Work with Bureau of Building Inspection (BBI). BBI does everything. BBI sometimes outsources inspection. Had reservations about buying a certificate because everything has to be built up to code…But interested in a crowd-sourced method to provide data.
Customer (Zeke's Coffee)	Business owners would be interested in purchasing a certificate of compliance	They can just look outside to see if sidewalk is bad. Would not care about certificate unless it was somehow attached to BBI, but changed mind slightly after they understood roughness was part of it. Maybe have "sidewalk insurance." May pay for subscription and if there is a problem, someone comes out to fix it.
Customer (Applied Research Associates)	PathMeT would be a good product for asset management companies	PathMeT is an "all-in-one" product. It has all important features including picture and GPS. They would buy PathMeT right now. They could have used PathMeT in their recent sidewalk project.

Table 19 (continued)

Urban Redevelopment Expert	PathMeT could be utilized by urban developers	Understanding the width of sidewalks is important. Business improvement districts (BIDs) would be very interested. Nearby BIDs may want to go in together to purchase one device. A service model may be preferable for BIDs.
Urban Redevelopment Expert	PathMeT could be utilized by urban developers	PathMeT would be useful for Bureau of Building Inspection to find trip hazards. Sidewalks currently assessed using spreadsheet and 1-5 scale by looking at cracks and heaving.
Urban Redevelopment Expert	PathMeT is a useful product for urban developers	Current process is by visual inspection, rating "major" or "minor" hazards. People complain current process is not scientific. PathMeT may be the scientific method people want. Leasing PathMeT may be a good office.
Sidewalk Industry Expert	Business owners would be interested in purchasing a certificate of compliance	Mostly complaint drive: People contact BBI and they come out to assess. Businesses may pay for assessment if it were paired with cost to upgrade
Bicycle Industry Expert	PathMeT could be useful for cities to examine bike lanes	Cities would not pay for to examine bike lanes because it presents them with a liability. However, the information may be important for bicyclists.
Insurance Expert	Insurance companies could offer discount for home owners who use PathMeT	Would not discount because liability costs are not too high. Liability is more expensive for commercial property. They must inspect every home if they are going to write a policy. Best target is bad landlords.

The Business Model Canvas

Key Partners	Key Activities	Value Propositions	Customer Relationships	Customer Segments
		Help to manage sidewalk infrastructure/Management Plan		**Compliance**
		3x Faster		Cities/Municipalities (Public Works)
		Decrease in trips/falls		Asset Management Companies
		Limit liability/lawsuits		Consumers that do compliance for Insurances & Attorneys
	Key Resources	Performance: All-in-one	**Channels**	
		GIS Integration		**Maintenance**
		Roughness Measurement		Cities/Municipalities (Public Works)
		Accessibility		Retailers
		Maintenance Plan/Updates		Property Owners
				Brick Layers

Cost Structure	Revenue Streams

Figure 40: Example business model canvas with PathMeT Product/Market Fit

64

4.0   DISCUSSION

PathMeT has ability to measure pathway roughness and other characteristics with the high detail and accuracy necessary to determine whether a new or existing surface is considered accessible, based on compliance with impending Access Board thresholds.

The fast sampling rates of the laser and the encoder sensors allow PathMeT to measure with 1mm resolution. The mechanical design also facilitates accurate data collection. The 560 mm non-pneumatic solid foam-filled tires allow PathMeT to roll over large cracks without being affected by crack characteristics. These mechanical design features that reduce the amount of errors in the system will result in fewer filters needed in code.

PathMeT is user-friendly in both electronic and mechanical design. The touchscreen display allows simple, intuitive, interaction with the system. In addition, the graph of the profile during data collection shows the user any extremely rough patches. At these positions, data collection may be repeated to ensure accuracy. In addition, the rolling design improves usability by the ease with which data is collected. The user can collect data in a timely fashion; with an average propulsion speed of 1m/s, the user can measure a mile of surfaces in less than 30 minutes. After data collection, the user can easily transport PathMeT by removing the wheels and collapsing the handle. PathMeT was designed so that any person with a high school education or better can operate. Typical training time would consist of 30 minutes to one hour, which includes demonstration and sample runs and analysis by the user.

PathMeT measures running slope and cross slope, but low-pass filtering techniques are being implemented in order to get more accurate results by reducing high frequency noise. Furthermore, the SignalQuest GravityGyro inclinometer, which is not sensitive to linear accelerations, will be implemented in the next version of PathMeT to ensure accurate slope readings. However, much progress has been made on measuring roughness and level change and the consistency with which PathMeT measures these aspects. The results from the intra- and inter-class correlation analysis shows that with a correlation of 0.993 and 0.983, respectively, and a Cronbach's alpha of 0.993, the use of PathMeT is highly reliable. Furthermore, an intra-class correlation of single measures value of 0.979 is highly valuable, especially since each surface is most likely to be measured by a single user. Both single measures and average measures intra-class correlation results showed a significance of $p<.001$. Since PathMeT has shown to be highly reliable in terms of measuring these surfaces, further testing can continue without concern about biased results.

Examination of the results from testing for level change indicates that placement of the laser greatly affects the profile. The excess vibrations and errors in the system cause a greater roughness than expected. When the front wheel ascends a step, a certain amount of error is experienced. The same occurs when the back wheel reaches the step. Placement of the laser behind the axle appears to be least desirable since the resulting profile is most difficult to determine when the laser actually ascends the step. Laser placement in front of the axle provides the clearest indication of where the step occurs. However, more testing is recommended to investigate the effects of putting the laser directly under the axle. It is clear that when the laser is directly under the axle, it resulting profile is similar to that of the laser placement behind the axle. This may be due to imperfect alignment under the axle or the realization that the wheel reaches the step before the laser. Therefore, it may be beneficial to test how the laser reacts when

placed at the point where the back wheels first hit the step. Regardless of the laser placement, algorithms must be developed to translate laser & encoder measurement results into accurate profiles of the surface.

With the use of post-processing programs, it was shown that PathMeT can not only measure roughness accurately, but this data can be uploaded into GIS, such as Google Earth. The KML Generator program allows data from PathMeT to be quickly uploaded to a database; then, a .kml file is created for compatibility with Google Earth. This ability to map information is important, especially for cities and business improvement districts that need to develop strategic plans for the improvement of sidewalk infrastructure. The heat map of compliant (green), moderately compliant (yellow), and non-compliant (red) surfaces can assist in this planning. Another useful tool for planning is the inspector's ability to visually examine the picture from the mapping.

These capabilities, features, and data that PathMeT provide make it a viable product for commercialization. Based on the nine stakeholder interviews, it was discovered that municipalities and asset management companies are among the customers who could utilize PathMeT. The *business model canvas* helps with customer discovery, understanding the market, and the development of a business model. The current status of the business model allows for revenue from both a service and PathMeT sales. Some large entities may prefer to purchase PathMeT and use it themselves; while others might prefer to have it contracted. Regardless, there is much potential in a startup company founded around PathMeT.

4.1 PATHMET VS. EXISTING TECHNOLOGY

Table 20 shows a comparison of PathMeT with other existing technology, which were described above. It can be seen that no other device is capable of measuring sidewalk roughness according to the developing standard. While SurPRO measures IRI, it is typically used for road roughness at a resolution of 25 mm. This is unsatisfactory for measuring sidewalk roughness because it can miss important characteristics, such as small joints. Furthermore, the ULIPs is capable of obtaining texture data, but cannot extract PRI. The ULIPs costs an estimated $120,000 (Loewenherz). PathMeT is significantly less expensive than this, selling for approximately one third of this price. Devices by Beneficial Designs are capable of measuring pathway characteristics, but three devices are needed to make all the measurements made by PathMeT. Finally, the use of a level and tape measure are extremely tedious and time-consuming tools to use. PathMeT can measure the same surfaces in 1/10 the amount of time.

Another way PathMeT differs from other surface measurement instruments is its capability of gathering data while in motion or stationary. These two data collection modes are referred to as Rolling Mode and Inchworm Mode respectively. If sections of the pathway being measured become too uneven or produce errors in the collection process, Inchworm Mode can be utilized to continue collection. During Inchworm Mode, the laser moves automatically along a 500 mm track while the PathMeT device remains stationary. Concerning precision, other devices such as the SurPRO 3500 can only measure longitudinal profiles with 6mm resolution. PathMeT is an all-in-one package, measuring cross slope, running slope, level change, roughness (1mm resolution), and obtaining GPS location and a picture.

68

Table 20: Comparison of PathMeT characteristics with existing technology: ●=automated, ◉=manual, ○=none

	PathMeT	ULIPs	Magic Cart	SurPRO	Level & Tape Measure
PRI	●	○	○	○	○
GPS Location	●	◉	●	○	○
Picture	●	●	●	○	○
Level Change	●	●	◉	○	◉
Running Slope	●	●	●	○	◉
Cross Slope	●	●	●	○	◉
Width	●	○	◉	○	◉

4.2 LIMITATIONS

Although much progress was made in the development of PathMeT, limitations exist. For ease in manufacturing, PathMeT was made of steel; and, consequently, the target weight of PathMeT was exceeded. Similarly, the use of 560 mm wheels required the footprint to be larger than intended. These two features cause the design to lack in user-friendliness. With its current design, PathMeT does not fit in a standard automobile trunk; it can only be accommodated by larger trunks, such as those in vans, SUVs, and trucks. In addition, it is difficult for the average

person to easily lift PathMeT in and out of the vehicle. Although these characteristics do not make PathMeT any less functional, they are, nonetheless, limitations in the design.

Another limitation is PathMeT's inability to consistently measure slope. As discussed, linear accelerations disrupt the accurate readings from the inclinometer and accelerometer. Although the results can be accurate at a constant speed, a new off-the-shelf device or new filtering technique should be implemented to get more consistent and accurate results. The SignalQuest GravityGyro is an off-the-shelf inclinometer that could be used to correct this problem. It costs around $1000. One possible filtering technique is the use of a Kalman filter. Similarly, GPS data is often skewed due to interference with satellite signals. This creates inaccurate GPS readings, necessitating corrections to be made manually. Improved GPS filtering and post-processing techniques are required to improve location mapping. Inaccurate location readings will continue to be a problem for any measurement device until sidewalks are mapped with an expensive, highly accurate GPS. Such GPS devices can cost upwards of $7000-$10,000.

Furthermore, surface profiles collected by PathMeT can often include noise as a result of traversing rough terrain. Although Inchworm Mode can correct for this limitation, it has not yet been fully implemented in software. As shown in Table 16, PRI values tend to be larger than those measured by P1 over the plywood tracks. A potential method to resolve this problem is through filtering techniques with an accurate inclinometer or accelerometer.

Although some limitations require the implementation of new, expensive devices, many improvements can be made by an increase in time and effort. Filtering techniques and methods to improve PathMeT have been identified; the critical task is to put more time into the project to resolve any issues.

5.0   FUTURE WORK

Future work will consist of further testing and characterization of PathMeT. Additional tests include testing how different speeds, light exposure, and weather conditions affect the results. In addition, more surfaces should be tested to ensure consistency of the device. These tests should also be done by moving PathMeT slowly over the surface to ensure the device is not biased by displacements experienced when PathMeT traverses a bump or crack. The use of aluminum would allow the device to be lighter and more user-friendly.

A method, such as use of a Kalman filter, to gather more accurate slope data is required. As shown, the inclinometer data is not reliable. A better inclinometer or filtering techniques should be implemented to eliminate errors due to linear acceleration. Although there are sensors that exist that accommodate for linear acceleration, such as the SignalQuest GravityGyro, these devices are expensive, around $1000. A filtering technique or the use of a gyroscope, are the preferred methods. Similarly, GPS filtering techniques should be implemented to remove the effect of triangulation interference due to buildings or other objects. There are methods that exist and have shown to be successful; however, they have not yet been implemented into PathMeT due to time constraints.

Additional data collection and analysis should occur in the community to find a correlation between PathMeT and P1 when driven over the plywood tracks. The resultant correlation would be analogous to the one established in Figure 28. Furthermore, additional

reliability testing should be conducted to ensure single operator and multi-operator reliability according to ASTM Precision and Bias testing guidelines. This will be conducted to ensure repeatability and reproducibility according to the national standard. Then, the coefficient of variance will be provided to describe PathMeT's precision, and the extent to which human interaction affects results.

In addition, a method to integrate data with sidewalk pavement management systems, commonly used by municipalities, should be developed. Many municipalities have a database that locates their streets and curbs, but it is uncommon for sidewalks to be part of this database. Furthermore, it is important for pedestrian pathway accessibility characteristics to be logged in this database. It will provide municipalities with a comprehensive resource that will allow them to manage their sidewalks. As a result, a municipality will be able to easily develop a strategic plan for improving its sidewalks. They will have quick access in identifying the most inaccessible and noncompliant areas, making these the top priorities for repair. This database will facilitate municipalities in their transition plan as the new standards and regulations are adopted. Then, cities like Los Angeles and Sacramento, as mentioned earlier in this report, will be able to address inaccessibility issues before litigation occurs, potentially saving them millions of dollars.

APPENDIX A

SOURCE CODE

A.1   POST-PROCESSING

A.1.1   MATLAB

Filter_ian.m

```
% Adapted from Tim's PostProcessing program (01082014)
% Written by Ian McIntyre on Jan 8, 2014

% (4laser + 4enc + 3time)*8 + 8inc(pitch,roll) + 6acc(x,y,z)

% Basic stuff. Clear the workspace, clean-up the command prompt, close all
% open graphs.  Creat prompt for importing the file, and then import the
% raw data bytes into raw_data.  Remember to close the file! The file name
% string variable (file) is used later for exporting the data (appended
% with ("_out.txt") for use with the WheelPath Algorithm.
clear all
clc
close all
file = input('Enter the file name ','s') ;
fid = fopen(file);
raw_data = fread(fid);
fclose(fid);

% Prototype the pitch and roll references.  You can comment out the input
% section, or set the references to zero again if you do not have a
% reference angle.
```

```
% Set the acceleration references. Determined on 03052014
pitch_ref = 0;
roll_ref = 0;
accx_ref = -0.0567;
accy_ref = 0.3171;
accz_ref = 0.70000; %2.9120

% % Used for pitch and roll references. Comment out to use raw data without
% % references
% pitch_ref = input('Pitch reference (deg): ');
% roll_ref = input('Roll reference (deg): ');

% Prototype all arrays. It may make the program faster.
laser = [];
encoder = [];
time = [];
inc = [];
acc = [];

byte = 1; % Counter for which byte your reading, and where that
% byte belongs in the data stream
array_loc_let = 1; % array counter for las enc and time
array_loc_ia = 1; % array counter for inc and acc
cycle_count = 0; % catches inc and acc data when = 8
sector_end = 0; % used only when = 5

%====BEGIN IMPORTING DATA====

while(1)

    % Check if we are at the end of the data file by finding eight
    % congruent zeros
    if(raw_data(byte) == 0 & raw_data(byte + 1) == 0 & raw_data(byte + 2) == 0 &
raw_data(byte + 3) == 0 & raw_data(byte + 4) == 0 & raw_data(byte + 5) == 0 & raw_data(byte
+ 6) == 0 & raw_data(byte + 7) == 0)
        break
    end

    % Reset the bitshift start after each iteration
    l = 12;
    e = 24;
    t = 16;

    % Collect raw data and place in temporary variables
    raw_laser = raw_data(byte:byte+3);
    raw_encoder = raw_data(byte+4:byte+7);
    raw_time = raw_data(byte+8:byte+10);
```

```matlab
for i = 1:length(raw_laser)
    % Perform bitshifts to correctly orient the numbers
    raw_laser(i) = bitshift(raw_laser(i),l);
    raw_encoder(i) = bitshift(raw_encoder(i),e);
    % Decrement the bitshifting value accordingly
    l = l - 4;
    e = e - 8;
end

for i = 1:length(raw_time)
    raw_time(i) = bitshift(raw_time(i),t);
    t = t - 8;
end

% Perform calculations to store in proper arrays
laser(array_loc_let) = (sum(raw_laser)/16463*19.5+5)*25.4; %mm
encoder(array_loc_let) = sum(raw_encoder)/34.632*25.4; %mm
time(array_loc_let) = sum(raw_time); %ms

% Check for 2's complement in encoder data to account for negative
% values
if (sum(raw_encoder) >= hex2dec('80000000'))
    encoder(array_loc_let) = sum(raw_encoder) - 4294967295;
else
    encoder(array_loc_let) = sum(raw_encoder);
end

array_loc_let = array_loc_let + 1;

% Catch the inclinometer and accelerometer every 8 cycles
cycle_count = cycle_count + 1;
if(cycle_count == 8)

    % Reset the bitshift start each iteration
    pr = 24;
    a = 8;

    raw_pitch = raw_data(byte+11:byte+14);
    raw_roll = raw_data(byte+15:byte+18);
    raw_accx = raw_data(byte+19:byte+20);
    raw_accy = raw_data(byte+21:byte+22);
    raw_accz = raw_data(byte+23:byte+24);

    byte = byte + 14;

    for i = 1:length(raw_pitch)
```

```matlab
            raw_pitch(i) = bitshift(raw_pitch(i),pr);
            raw_roll(i) = bitshift(raw_roll(i),pr);
            pr = pr - 8;
        end

        for i = 1:length(raw_accx)
            raw_accx(i) = bitshift(raw_accx(i),a);
            raw_accy(i) = bitshift(raw_accy(i),a);
            raw_accz(i) = bitshift(raw_accz(i),a);
            a = a - 8;
        end

        %pitch(array_loc_ia) = sum(raw_pitch)/1000;
        %roll(array_loc_ia) = sum(raw_roll)/1000;
        accx(array_loc_ia) = (sum(raw_accx)*3.3/1023 - 1.65)/0.3;
        accy(array_loc_ia) = (sum(raw_accy)*3.3/1023 - 1.65)/0.3;
        accz(array_loc_ia) = (sum(raw_accz)*3.3/1023 - 1.65)/0.3;

        % Check for 2's complement in pitch and roll data to account for
        % negative values
        if (sum(raw_pitch) >= hex2dec('80000000'))
            pitch(array_loc_ia) = (sum(raw_pitch) - 4294967295)/1000 - pitch_ref;
        else
            pitch(array_loc_ia) = sum(raw_pitch)/1000 - pitch_ref;
        end

        if (sum(raw_roll) >= hex2dec('80000000'))
            roll(array_loc_ia) = (sum(raw_roll) - 4294967295)/1000 - roll_ref;
        else
            roll(array_loc_ia) = sum(raw_roll)/1000 - roll_ref;
        end

        array_loc_ia = array_loc_ia + 1;
        cycle_count = 0;
        sector_end = sector_end + 1;
    end
    byte = byte + 11;

    if sector_end == 5
        byte = byte + 2;
        sector_end = 0;
    end

    if byte == length(raw_data) - 512
        break
    end
end
```

```matlab
%====COMPLETE IMPORTING DATA====

% Assess the differences in encoder position and time
encoder_diff = diff(encoder);
time_diff = diff(time);
encoder_error = length(find(encoder_diff ~= 0 & encoder_diff ~= 1))/length(encoder)*100;
time_error = length(find(time_diff ~= 0 & time_diff ~= 1))/length(time_diff)*100;

% Flip the laser data so that all distances are referenced from the
% starting laser value.
laser = -laser;

%====BEGIN FILTERING DATA====

% Filter the inclinometer data using a 7th order butterworth filter with
% at an extremely low cuttoff frequency (2% percent of sampling
% frequency).  This gives a smooth response of the inclinometer and filters
% the noise
[num,denom] = butter(7,0.02);
pitch_butter = filtfilt(num,denom,pitch);
roll_butter7 = filtfilt(num,denom,roll);

% Also try a 2nd order butterworth with w_c = 0.02w_s
[num,denom] = butter(2,0.02);
pitch_butter2 = filtfilt(num,denom,pitch);
roll_butter2 = filtfilt(num,denom,roll);

% % Run the laser data through a 1st order butterworth filter
% [num,denom] = butter(1,0.25);
% laser_butter1 = filtfilt(num,denom,laser);
% laser_error = laser_butter1 - laser;
% laser_mse = mean(laser_error.^2);

% Filter the accelerometer data, provide a start-up low pass filter
N = 800;
[num,denom] = butter(1,0.005);
accx_butter = filtfilt(num,denom,[accx(N:-1:1),accx]);
accy_butter = filtfilt(num,denom,[accy(N:-1:1),accy]);
accz_butter = filtfilt(num,denom,[accz(N:-1:1),accz]);
accx_butter = accx_butter(N+1:end);
accy_butter = accy_butter(N+1:end);
accz_butter = accz_butter(N+1:end);

%====COMPLETE FILTERING DATA====

% Take 1/8 of the encoder and time data for plotting with inclinometer and
```

```matlab
% accelerometer data.  Also need 1/8 of the filtered laser data for
% creating the road profile
j = 1;
encoder_eight = zeros(1,length(pitch));
time_eight = encoder_eight;
laser_eight = time_eight;
for i = 8:8:length(time)
   encoder_eight(j) = encoder(i);
   time_eight(j) = time(i);
   laser_eight(j) = laser(i);
   j = j + 1;
end

% Create the road profiles
[x_profile_pitch,y_profile_pitch] = linFunctTransform(encoder_eight,...
   laser_eight,pitch_butter);
[x_profile_roll,y_profile_roll] = linFunctTransform(encoder_eight,...
   laser_eight,roll_butter7);

% Create stellar 3D plots
laser_3d = [laser;laser];

% Calculate pitch angle based on acceleration
% Use the areospace rotation sequence to calculate pitch and roll based on
% the accelerometer readings
% tan(pitch) = -g_x / sqrt(g_y^2 + g_z^2) [-90deg:90deg]
% tan(roll) = g_y / g_z [-180deg:180deg]
for i = 1:length(accx_butter)
%    gravMag(i) = sqrt(accz_butter(i)^2 + accx_butter(i)^2 + accy_butter(i)^2);

%    test = mean(accz_butter)/accz_butter(i);
%    if(test > 1)
%       pitch_calc(i) = acosd(mean(accz_butter) - test);
%    else
%       pitch_calc(i) = acosd(test);
%    end


%    pitch_calc(i) = asind((accz_ref - accz_butter(i))/accz_butter(i));

   bottom = sqrt(accy_butter(i)^2 + accz_butter(i)^2);
   top = -accx_butter(i);
   pitch_areo(i) = -atand(top/bottom);
end

% for i = 1:length(accx_butter)
%    acc_mag(i) = sqrt(accx_butter(i)^2 + accy_butter(i)^2 + accz_butter(i)^2);
```

```
%     if (i == 1)
%         pitch_calc2(i) = acosd(accz_butter(i)/acc_mag(i)) - acosd(accx_butter(i)/acc_mag(i));
%     else
%         pitch_calc2(i) = acosd(accz_butter(i)/acc_mag(i))-acosd(accz_butter(i-1)/acc_mag(i-1)) -
acosd(accx_butter(i)/acc_mag(i));
%     end
% end

% Calculate linear velocity and acceleration based on encoder data.
% Low-pass filter the velocity to remove noise
raw_vel = diff(encoder_eight)./diff(time_eight);
[num,denom] = butter(3,0.008);
lin_vel = filtfilt(num,denom,raw_vel);
raw_acc = diff(lin_vel)./diff(time_eight(1:length(lin_vel)));
lin_acc = filtfilt(num,denom,raw_acc);

%====BEGIN PLOTTING ALL DATA====

% Laser vs Encoder
figure(1), plot(encoder,laser,':r','LineWidth',1.6),grid,...
    title('Laser vs Encoder'),...
    xlabel('Encoder Position (mm)'), ylabel('Laser (mm from PathMeT)')

% % Encoder vs Time
% figure(2), plot(time,encoder,'-m','LineWidth',2.1),...
%     grid, title('Encoder vs Time'),...
%     xlabel('Time (ms)'), ylabel('Encoder Position (mm)')

% % All three (xyz) Accelerations vs Encoder (eight)
% figure(3), plot(encoder_eight,accx,':r',...
%     encoder_eight,accy,':g',...
%     encoder_eight,accz,':b',...
%     encoder_eight,accx_butter,'-c',...
%     encoder_eight,accy_butter,'-k',...
%     encoder_eight,accz_butter,'-m',...
%     'LineWidth',1.6),...
%     grid, title('Accelerations vs Position'),...
%     xlabel('Encoder Position (mm)'), ylabel('Accelerations (Gs)'),...
%     legend('Raw x-axis',' Raw y-axis','Raw z-axis',...
%     'Filtered x-axis','Filtered y-axis','Filtered z-axis')

figure(4), plot(...
    time_eight,accx,':r',...
    time_eight,accy,':g',...
    time_eight,accz,':b',...
    'LineWidth',1.6),...
    grid, title('Accelerations vs Time'),...
```

```matlab
    xlabel('Time (ms)'), ylabel('Accelerations (Gs)'),...
    legend('Raw x-axis',' Raw y-axis','Raw z-axis')


%    time_eight,accx_butter,'-c',...
%    time_eight,accy_butter,'-k',...
%    time_eight,accz_butter,'-m',...

% % Analysis of Encoder and Time differences.  Good check for lossy data
% figure(4), subplot(2,1,1), plot(encoder_diff,'.k'), axis([0,length(encoder_diff),-3,3]),...
%    grid, title('Difference in Encoder Values (0 OR 1 Desirable)'),...
%    legend(['Percentage of position errors ',num2str(encoder_error),'%'])
% subplot(2,1,2), plot(time_diff,'.k'), axis([0,length(time_diff),-3,3]), grid,...
%    title('Difference in Time (ms) (0 OR 1 Desirable)'),...
%    legend(['Percentage of timing errors ',num2str(time_error),'%'])

% Pitch vs time (eight)
figure(5), plot(...
    time_eight,pitch,'--r',...
    time_eight,pitch_butter,'-b',...
    time_eight,pitch_areo,'m',...
    'LineWidth',2.1),...
    grid, title('Pitch vs Time'),...
    xlabel('Time (ms)'), ylabel('Degrees'),...
    legend('Unfiltered Pitch','Pitch Filtered with Butterworth Filter',...
    'Pitch determined via accelerometer')

% % Roll vs time (eight)
% figure(6), plot(...
%    time_eight,roll,'--m',...
%    time_eight,roll_butter7,'-c',...
%    time_eight,roll_butter2,':k','LineWidth',2.1),...
%    grid, title('Roll vs time (Filtered with 7th and 2nd Order Butterworth Filters with w_c =
(0.02)w_s)'),...
%    xlabel('Time (ms)'), ylabel('Degrees'),...
%    legend('Raw','7th Filtered','2nd Filtered')

% % Surface Profile based on Pitch
% figure(7), plot(x_profile_pitch,y_profile_pitch,'-k','LineWidth',2.3), grid,...
%    title('TEST Surface Profile: Filtered Laser Rotated by Filtered Pitch vs Encoder'),...
%    xlabel('Encoder Position (mm run)'), ylabel('Surface Profile by Pitch (mm rise)')

% % Surface Profile based on Roll
% figure(8), plot(x_profile_roll,y_profile_roll,'-k','LineWidth',2.3), grid,...
%    title('TEST Surface Profile: Filtered Laser Rotated by Filtered Roll vs Encoder'),...
%    xlabel('Encoder Position (counts)'), ylabel('Surface Profile by Roll (units TBD)')
```

```matlab
figure(9), surf(laser_3d), shading interp, axis off, alpha(1.0),...
   colorbar, title('Laser Perturbation Visualization')

% % Plot linear velocity with filtered linear velocity
% figure(10), plot(time_eight(1:length(raw_vel)),raw_vel,'-b',...
%    time_eight(1:length(lin_vel)),lin_vel,'-r',...
%    'LineWidth',2.1), grid,...
%    title('Linear Velocity (derivative of encoder data)'),...
%    xlabel('Time (ms)'),...
%    ylabel('Linear Velocity (ms)')

% figure(11), plot(time_eight(1:length(raw_acc)),raw_acc,'-g',...
%    time_eight(1:length(lin_acc)),lin_acc,'-k',...
%    'LineWidth',2.1), grid,...
%    title('Linear Acceleration (second derivative of encoder data)'),...
%    xlabel('Time (ms)'),...
%    ylabel('Linear Acceleration (ms)')

% figure(12), plot(...
%    time_eight,accx_butter,'-c',...
%    time_eight(1:length(lin_acc)),lin_acc*1000/9.81,'-k',...
%    'LineWidth',2.3),...
%    grid,

%====COMPLETE PLOTTING ALL DATA====

%====REMOVE THE FALSE PITCH READINGS IN THE BEGINNING AND END====
% pitch_butter = fixInclinometer(time_eight,pitch_butter);
% figure(99), plot(time_eight,pitch_butter)

% % Export filtered laser and encoder data to text file.
% % Laser in first column, encoder in second columnm, separated by SPACE
% % (' '). The data can be imported into the WheelPath Algorithm Program.
% % Note that the laser data is exported with negative values to give the
% % more literal "appearance" of the surface.  Therefore, make sure that the
% % data is not being flipped again in the WheelPath Algorithm Program.
% outFileRaw = [laser;encoder]';
% outFileRawString = strcat(file,'_out.txt');
% dlmwrite(outFileRawString,outFileRaw,' ');

% % Export the Accelerometer data [position \t accx \t accy \t accz]
% outAccFile = [time_eight;accx;accy;accz]';
% outAccFileString = strcat(file,'_timeAcc.txt');
% dlmwrite(outAccFileString,outAccFile,'\t');

% % Export the Inclinometer data [position \t pitch \t roll]
% outIncFile = [time_eight;pitch;roll]';
```

```
% outIncFileString = strcat(file,'_timeInc.txt');
% dlmwrite(outIncFileString,outIncFile,'\t');



%====CHECK THROUGH FILTERS, REAPPLY NEW FILTERS====%



% while(1)
%    filt_opts = menu('Want to make any new filters?','Laser','Pitch','Roll','Accelerations','No');
%
%    if (filt_opts ~= 5 && filt_opts ~= 0)
%        butter_order = input('What order for Butterworth Filter? ');
%        butter_wc = input('What cuttoff frequency? ');
%    end
%
%    switch(filt_opts)
%        case(0)
%           break
%        case(1)
%           [num,denom] = butter(butter_order,butter_wc);
%           laser_butter_opts = filtfilt(num,denom,laser);
%           laser_error = laser_butter_opts - laser;
%           laser_mse = mean(laser_error.^2);
%
%           figure(10), plot(encoder,laser,':r',encoder,laser_butter_opts,'-b',...
%           'LineWidth',1.6),grid,...
%           title('Laser vs Encoder (Custom Butterworth Filter)'),...
%           xlabel('Encoder Position (mm)'), ylabel('Laser (mm from PathMeT'),...
%           legend('Raw',['1st Filtered (MSE = ',num2str(laser_mse),')'])
%
%        case(5)
%           break
%    end
%
% end
```

linFunctTransform.m

% Perform the continuous linear transformation on a function y(x) when the
% function theta defines the angles of vector rotation. This script takes
% three inputs: the independent variables x, the dependent variables y, and
% the angles of vector rotation theta. The script first creates the
% rotation matrix transform using the function linRotateTransform(), a
% function which takes a single angle in degrees. The script performs
% iterations by calculating the vector between two points, rotating the
% vector by the angle theta through LH matrix multiplication, and adding
% the new vector to the overall transformed function. The script outputs
% the transformed x and transformed y data as separate arrays.

% Ian McIntyre 01102014 for use with PathMeT postprocessing

```
function [x_transform, y_transform] = linFunctTransform(x,y,theta)

for i = 1:length(theta)
   transform = linRotateTransform(theta(i));
   if i == 1
      f_transform(:,i) = transform*[x(i);y(i)];
   else
      f_transform(:,i) = transform*[x(i)-x(i-1);y(i)-y(i-1)]+...
         [f_transform(1,i-1);f_transform(2,i-1)];
    end
end

x_transform = f_transform(1,:);
y_transform = f_transform(2,:);

end
```

linRotateTransform.m

% Create the rotation matrix cooresponding to an angle theta in degrees

% Ian McIntyre 01102014 for use with PathMeT postprocessing, specifically
% called in linFunctTransform.m

function mat = linRotateTransform(theta)

mat = [cosd(theta) -sind(theta); sind(theta) cosd(theta)];

end

A.1.2 Java

Form1.Designer.cs

```
namespace PostProcessing
{
  partial class Form1
  {
    /// <summary>
    /// Required designer variable.
    /// </summary>
    private System.ComponentModel.IContainer components = null;

    /// <summary>
    /// Clean up any resources being used.
    /// </summary>
    /// <param name="disposing">true if managed resources should be disposed;
otherwise, false.</param>
    protected override void Dispose(bool disposing)
    {
      if (disposing && (components != null))
      {
        components.Dispose();
```

```
        }
        base.Dispose(disposing);
    }

    #region Windows Form Designer generated code

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
        this.components = new System.ComponentModel.Container();
        this.buttonReadHexFile = new System.Windows.Forms.Button();
        this.openFileDialog1 = new System.Windows.Forms.OpenFileDialog();
        this.richTextBox1 = new System.Windows.Forms.RichTextBox();
        this.labelBytesRead = new System.Windows.Forms.Label();
        this.progressBar1 = new System.Windows.Forms.ProgressBar();
        this.backgroundWorker1 = new System.ComponentModel.BackgroundWorker();
        this.zedLazerEncoder = new ZedGraph.ZedGraphControl();
        this.buttonPlot = new System.Windows.Forms.Button();
        this.zedLaserTime = new ZedGraph.ZedGraphControl();
        this.zedEncoderTime = new ZedGraph.ZedGraphControl();
        this.zedEncoderDiff = new ZedGraph.ZedGraphControl();
        this.tabControl1 = new System.Windows.Forms.TabControl();
        this.tabPage1 = new System.Windows.Forms.TabPage();
        this.tabPage2 = new System.Windows.Forms.TabPage();
        this.tabControl2 = new System.Windows.Forms.TabControl();
        this.tabPage3 = new System.Windows.Forms.TabPage();
        this.zedAccx = new ZedGraph.ZedGraphControl();
        this.tabPage4 = new System.Windows.Forms.TabPage();
        this.zedRoll = new ZedGraph.ZedGraphControl();
        this.zedPitch = new ZedGraph.ZedGraphControl();
        this.tabControl3 = new System.Windows.Forms.TabControl();
        this.tabPage5 = new System.Windows.Forms.TabPage();
        this.tabPage6 = new System.Windows.Forms.TabPage();
        this.tabControl1.SuspendLayout();
        this.tabPage1.SuspendLayout();
        this.tabPage2.SuspendLayout();
        this.tabControl2.SuspendLayout();
        this.tabPage3.SuspendLayout();
        this.tabPage4.SuspendLayout();
        this.tabControl3.SuspendLayout();
        this.tabPage5.SuspendLayout();
        this.tabPage6.SuspendLayout();
        this.SuspendLayout();
        //
```

```
// buttonReadHexFile
//
this.buttonReadHexFile.Location = new System.Drawing.Point(16, 33);
this.buttonReadHexFile.Name = "buttonReadHexFile";
this.buttonReadHexFile.Size = new System.Drawing.Size(100, 44);
this.buttonReadHexFile.TabIndex = 0;
this.buttonReadHexFile.Text = "Read";
this.buttonReadHexFile.UseVisualStyleBackColor = true;
this.buttonReadHexFile.Click                              +=                              new
System.EventHandler(this.buttonReadHexFile_Click);
//
// openFileDialog1
//
this.openFileDialog1.FileName = "openFileDialog1";
this.openFileDialog1.FileOk                              +=                              new
System.ComponentModel.CancelEventHandler(this.openFileDialog1_FileOk);
//
// richTextBox1
//
this.richTextBox1.Font = new System.Drawing.Font("Microsoft Sans Serif",
9.75F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.richTextBox1.Location = new System.Drawing.Point(3, 0);
this.richTextBox1.Name = "richTextBox1";
this.richTextBox1.Size = new System.Drawing.Size(957, 194);
this.richTextBox1.TabIndex = 1;
this.richTextBox1.Text = "";
//
// labelBytesRead
//
this.labelBytesRead.AutoSize = true;
this.labelBytesRead.Font = new System.Drawing.Font("Microsoft Sans Serif",
12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.labelBytesRead.Location = new System.Drawing.Point(213, 7);
this.labelBytesRead.Name = "labelBytesRead";
this.labelBytesRead.Size = new System.Drawing.Size(37, 20);
this.labelBytesRead.TabIndex = 2;
this.labelBytesRead.Text = "      ";
//
// progressBar1
//
this.progressBar1.Location = new System.Drawing.Point(16, 12);
this.progressBar1.Name = "progressBar1";
this.progressBar1.Size = new System.Drawing.Size(114, 15);
this.progressBar1.TabIndex = 3;
//
// backgroundWorker1
//
```

```csharp
this.backgroundWorker1.WorkerReportsProgress = true;
this.backgroundWorker1.DoWork                        +=                        new
System.ComponentModel.DoWorkEventHandler(this.backgroundWorker1_DoWork);
this.backgroundWorker1.ProgressChanged               +=                        new
System.ComponentModel.ProgressChangedEventHandler(this.backgroundWorker1_ProgressCh
anged);
//
// zedLazerEncoder
//
this.zedLazerEncoder.Location = new System.Drawing.Point(6, 6);
this.zedLazerEncoder.Name = "zedLazerEncoder";
this.zedLazerEncoder.ScrollGrace = 0D;
this.zedLazerEncoder.ScrollMaxX = 0D;
this.zedLazerEncoder.ScrollMaxY = 0D;
this.zedLazerEncoder.ScrollMaxY2 = 0D;
this.zedLazerEncoder.ScrollMinX = 0D;
this.zedLazerEncoder.ScrollMinY = 0D;
this.zedLazerEncoder.ScrollMinY2 = 0D;
this.zedLazerEncoder.Size = new System.Drawing.Size(1530, 241);
this.zedLazerEncoder.TabIndex = 4;
//
// buttonPlot
//
this.buttonPlot.Location = new System.Drawing.Point(122, 34);
this.buttonPlot.Name = "buttonPlot";
this.buttonPlot.Size = new System.Drawing.Size(87, 43);
this.buttonPlot.TabIndex = 5;
this.buttonPlot.Text = "Plot";
this.buttonPlot.UseVisualStyleBackColor = true;
this.buttonPlot.Click += new System.EventHandler(this.buttonPlot_Click);
//
// zedLaserTime
//
this.zedLaserTime.Location = new System.Drawing.Point(3, 3);
this.zedLaserTime.Name = "zedLaserTime";
this.zedLaserTime.ScrollGrace = 0D;
this.zedLaserTime.ScrollMaxX = 0D;
this.zedLaserTime.ScrollMaxY = 0D;
this.zedLaserTime.ScrollMaxY2 = 0D;
this.zedLaserTime.ScrollMinX = 0D;
this.zedLaserTime.ScrollMinY = 0D;
this.zedLaserTime.ScrollMinY2 = 0D;
this.zedLaserTime.Size = new System.Drawing.Size(1516, 241);
this.zedLaserTime.TabIndex = 6;
//
// zedEncoderTime
//
```

```
this.zedEncoderTime.Location = new System.Drawing.Point(3, 0);
this.zedEncoderTime.Name = "zedEncoderTime";
this.zedEncoderTime.ScrollGrace = 0D;
this.zedEncoderTime.ScrollMaxX = 0D;
this.zedEncoderTime.ScrollMaxY = 0D;
this.zedEncoderTime.ScrollMaxY2 = 0D;
this.zedEncoderTime.ScrollMinX = 0D;
this.zedEncoderTime.ScrollMinY = 0D;
this.zedEncoderTime.ScrollMinY2 = 0D;
this.zedEncoderTime.Size = new System.Drawing.Size(1513, 457);
this.zedEncoderTime.TabIndex = 7;
//
// zedEncoderDiff
//
this.zedEncoderDiff.Location = new System.Drawing.Point(3, 250);
this.zedEncoderDiff.Name = "zedEncoderDiff";
this.zedEncoderDiff.ScrollGrace = 0D;
this.zedEncoderDiff.ScrollMaxX = 0D;
this.zedEncoderDiff.ScrollMaxY = 0D;
this.zedEncoderDiff.ScrollMaxY2 = 0D;
this.zedEncoderDiff.ScrollMinX = 0D;
this.zedEncoderDiff.ScrollMinY = 0D;
this.zedEncoderDiff.ScrollMinY2 = 0D;
this.zedEncoderDiff.Size = new System.Drawing.Size(1516, 237);
this.zedEncoderDiff.TabIndex = 8;
//
// tabControl1
//
this.tabControl1.Controls.Add(this.tabPage1);
this.tabControl1.Controls.Add(this.tabPage2);
this.tabControl1.Location = new System.Drawing.Point(6, 200);
this.tabControl1.Name = "tabControl1";
this.tabControl1.SelectedIndex = 0;
this.tabControl1.Size = new System.Drawing.Size(1530, 519);
this.tabControl1.TabIndex = 9;
//
// tabPage1
//
this.tabPage1.Controls.Add(this.zedLaserTime);
this.tabPage1.Controls.Add(this.zedEncoderDiff);
this.tabPage1.Location = new System.Drawing.Point(4, 22);
this.tabPage1.Name = "tabPage1";
this.tabPage1.Padding = new System.Windows.Forms.Padding(3);
this.tabPage1.Size = new System.Drawing.Size(1522, 493);
this.tabPage1.TabIndex = 0;
this.tabPage1.Text = "Debug1";
this.tabPage1.UseVisualStyleBackColor = true;
```

```
//
// tabPage2
//
this.tabPage2.Controls.Add(this.zedEncoderTime);
this.tabPage2.Location = new System.Drawing.Point(4, 22);
this.tabPage2.Name = "tabPage2";
this.tabPage2.Padding = new System.Windows.Forms.Padding(3);
this.tabPage2.Size = new System.Drawing.Size(1522, 493);
this.tabPage2.TabIndex = 1;
this.tabPage2.Text = "Debug2";
this.tabPage2.UseVisualStyleBackColor = true;
//
// tabControl2
//
this.tabControl2.Controls.Add(this.tabPage3);
this.tabControl2.Controls.Add(this.tabPage4);
this.tabControl2.Location = new System.Drawing.Point(6, 253);
this.tabControl2.Name = "tabControl2";
this.tabControl2.SelectedIndex = 0;
this.tabControl2.Size = new System.Drawing.Size(1533, 630);
this.tabControl2.TabIndex = 10;
//
// tabPage3
//
this.tabPage3.Controls.Add(this.zedAccx);
this.tabPage3.Location = new System.Drawing.Point(4, 22);
this.tabPage3.Name = "tabPage3";
this.tabPage3.Padding = new System.Windows.Forms.Padding(3);
this.tabPage3.Size = new System.Drawing.Size(1525, 604);
this.tabPage3.TabIndex = 0;
this.tabPage3.Text = "Accelerometer";
this.tabPage3.UseVisualStyleBackColor = true;
//
// zedAccx
//
this.zedAccx.Location = new System.Drawing.Point(5, 3);
this.zedAccx.Name = "zedAccx";
this.zedAccx.ScrollGrace = 0D;
this.zedAccx.ScrollMaxX = 0D;
this.zedAccx.ScrollMaxY = 0D;
this.zedAccx.ScrollMaxY2 = 0D;
this.zedAccx.ScrollMinX = 0D;
this.zedAccx.ScrollMinY = 0D;
this.zedAccx.ScrollMinY2 = 0D;
this.zedAccx.Size = new System.Drawing.Size(1505, 450);
this.zedAccx.TabIndex = 5;
//
```

```
// tabPage4
//
this.tabPage4.Controls.Add(this.zedRoll);
this.tabPage4.Controls.Add(this.zedPitch);
this.tabPage4.Location = new System.Drawing.Point(4, 22);
this.tabPage4.Name = "tabPage4";
this.tabPage4.Padding = new System.Windows.Forms.Padding(3);
this.tabPage4.Size = new System.Drawing.Size(1525, 604);
this.tabPage4.TabIndex = 1;
this.tabPage4.Text = "Inclinometor";
this.tabPage4.UseVisualStyleBackColor = true;
//
// zedRoll
//
this.zedRoll.Location = new System.Drawing.Point(6, 215);
this.zedRoll.Name = "zedRoll";
this.zedRoll.ScrollGrace = 0D;
this.zedRoll.ScrollMaxX = 0D;
this.zedRoll.ScrollMaxY = 0D;
this.zedRoll.ScrollMaxY2 = 0D;
this.zedRoll.ScrollMinX = 0D;
this.zedRoll.ScrollMinY = 0D;
this.zedRoll.ScrollMinY2 = 0D;
this.zedRoll.Size = new System.Drawing.Size(817, 216);
this.zedRoll.TabIndex = 7;
//
// zedPitch
//
this.zedPitch.Location = new System.Drawing.Point(5, 13);
this.zedPitch.Name = "zedPitch";
this.zedPitch.ScrollGrace = 0D;
this.zedPitch.ScrollMaxX = 0D;
this.zedPitch.ScrollMaxY = 0D;
this.zedPitch.ScrollMaxY2 = 0D;
this.zedPitch.ScrollMinX = 0D;
this.zedPitch.ScrollMinY = 0D;
this.zedPitch.ScrollMinY2 = 0D;
this.zedPitch.Size = new System.Drawing.Size(817, 196);
this.zedPitch.TabIndex = 6;
//
// tabControl3
//
this.tabControl3.Controls.Add(this.tabPage5);
this.tabControl3.Controls.Add(this.tabPage6);
this.tabControl3.Location = new System.Drawing.Point(16, 83);
this.tabControl3.Name = "tabControl3";
this.tabControl3.SelectedIndex = 0;
```

```
this.tabControl3.Size = new System.Drawing.Size(1550, 789);
this.tabControl3.TabIndex = 11;
//
// tabPage5
//
this.tabPage5.Controls.Add(this.richTextBox1);
this.tabPage5.Controls.Add(this.tabControl1);
this.tabPage5.Location = new System.Drawing.Point(4, 22);
this.tabPage5.Name = "tabPage5";
this.tabPage5.Padding = new System.Windows.Forms.Padding(3);
this.tabPage5.Size = new System.Drawing.Size(1542, 763);
this.tabPage5.TabIndex = 0;
this.tabPage5.Text = "Debug";
this.tabPage5.UseVisualStyleBackColor = true;
//
// tabPage6
//
this.tabPage6.Controls.Add(this.zedLazerEncoder);
this.tabPage6.Controls.Add(this.tabControl2);
this.tabPage6.Location = new System.Drawing.Point(4, 22);
this.tabPage6.Name = "tabPage6";
this.tabPage6.Padding = new System.Windows.Forms.Padding(3);
this.tabPage6.Size = new System.Drawing.Size(1542, 763);
this.tabPage6.TabIndex = 1;
this.tabPage6.Text = "Output";
this.tabPage6.UseVisualStyleBackColor = true;
//
// Form1
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.ClientSize = new System.Drawing.Size(1596, 874);
this.Controls.Add(this.tabControl3);
this.Controls.Add(this.buttonPlot);
this.Controls.Add(this.progressBar1);
this.Controls.Add(this.labelBytesRead);
this.Controls.Add(this.buttonReadHexFile);
this.Name = "Form1";
this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
this.Text = "Form1";
this.WindowState = System.Windows.Forms.FormWindowState.Maximized;
this.tabControl1.ResumeLayout(false);
this.tabPage1.ResumeLayout(false);
this.tabPage2.ResumeLayout(false);
this.tabControl2.ResumeLayout(false);
this.tabPage3.ResumeLayout(false);
this.tabPage4.ResumeLayout(false);
```

```
            this.tabControl3.ResumeLayout(false);
            this.tabPage5.ResumeLayout(false);
            this.tabPage6.ResumeLayout(false);
            this.ResumeLayout(false);
            this.PerformLayout();

        }

        #endregion

        private System.Windows.Forms.Button buttonReadHexFile;
        private System.Windows.Forms.OpenFileDialog openFileDialog1;
        private System.Windows.Forms.RichTextBox richTextBox1;
        private System.Windows.Forms.Label labelBytesRead;
        private System.Windows.Forms.ProgressBar progressBar1;
        private System.ComponentModel.BackgroundWorker backgroundWorker1;
        private ZedGraph.ZedGraphControl zedLazerEncoder;
        private System.Windows.Forms.Button buttonPlot;
        private ZedGraph.ZedGraphControl zedLaserTime;
        private ZedGraph.ZedGraphControl zedEncoderTime;
        private ZedGraph.ZedGraphControl zedEncoderDiff;
        private System.Windows.Forms.TabControl tabControl1;
        private System.Windows.Forms.TabPage tabPage1;
        private System.Windows.Forms.TabPage tabPage2;
        private System.Windows.Forms.TabControl tabControl2;
        private System.Windows.Forms.TabPage tabPage3;
        private System.Windows.Forms.TabPage tabPage4;
        private ZedGraph.ZedGraphControl zedAccx;
        private ZedGraph.ZedGraphControl zedRoll;
        private ZedGraph.ZedGraphControl zedPitch;
        private System.Windows.Forms.TabControl tabControl3;
        private System.Windows.Forms.TabPage tabPage5;
        private System.Windows.Forms.TabPage tabPage6;
    }
}
```

```csharp
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;
using System.Threading;
using ZedGraph;

namespace PostProcessing
{
    public partial class Form1 : Form
    {
        private Byte[] rawBytes;
        StringBuilder st;

        GraphPane laserEncoderPlane;
        GraphPane TimeDiffPlane;
        GraphPane encoderTimePlane;
        GraphPane encoderDiffPlane;

        GraphPane AccxEncoderPlane;


        GraphPane PitchEncoderPlane;
        GraphPane RollEncoderPlane;



        PointPairList listLaseEnc;
        PointPairList listTimeDiff;
        PointPairList listEncodertime;
        PointPairList listEncDiff;

        PointPairList AccxEncoder;
        PointPairList AccyEncoder;
        PointPairList AcczEncoder;

        PointPairList PitchEncoder;
        PointPairList RollEncoder;
```

```csharp
public Form1()
{
    InitializeComponent();
    laserEncoderPlane = zedLazerEncoder.GraphPane;
    laserEncoderPlane.Title.Text = "Laser vs Encoder pos";
    laserEncoderPlane.XAxis.Title.Text = "Encoder Pos";
    laserEncoderPlane.YAxis.Title.Text = "Laser(inch)";


    TimeDiffPlane = zedLaserTime.GraphPane;
    TimeDiffPlane.Title.Text = "Time vs Time difference";
    TimeDiffPlane.XAxis.Title.Text = "Time(ms)";
    TimeDiffPlane.YAxis.Title.Text = "diff(ms)";

    encoderTimePlane = zedEncoderTime.GraphPane;
    encoderTimePlane.Title.Text = "Encoder vs Time";
    encoderTimePlane.XAxis.Title.Text = "Time(ms)";
    encoderTimePlane.YAxis.Title.Text = "Encoder Pos";

    encoderDiffPlane = zedEncoderDiff.GraphPane;
    encoderDiffPlane.Title.Text = "Encoder vs Encoder Diff";
    encoderDiffPlane.XAxis.Title.Text = "Encoder Position";
    encoderDiffPlane.YAxis.Title.Text = "ticks";

    AccxEncoderPlane = zedAccx.GraphPane;
    AccxEncoderPlane.Title.Text = "x-green,y-yellow,z-blue";
    AccxEncoderPlane.XAxis.Title.Text = "Encoder Position";
    AccxEncoderPlane.YAxis.Title.Text = "g";


    PitchEncoderPlane = zedPitch.GraphPane;
    PitchEncoderPlane.Title.Text = "Pitch vs Encoder";
    PitchEncoderPlane.XAxis.Title.Text = "Encoder Position";
    PitchEncoderPlane.YAxis.Title.Text = "deg";
    RollEncoderPlane = zedRoll.GraphPane;
    RollEncoderPlane.Title.Text = "Roll vs Encoder";
    RollEncoderPlane.XAxis.Title.Text = "Encoder Position";
    RollEncoderPlane.YAxis.Title.Text = "deg";
```

```csharp
        }

        private void openFileDialog1_FileOk(object sender, CancelEventArgs e)
        {
            backgroundWorker1.RunWorkerAsync();

        }

        private void buttonReadHexFile_Click(object sender, EventArgs e)
        {
            openFileDialog1.ShowDialog();

        }

        private void backgroundWorker1_DoWork(object sender, DoWorkEventArgs e)
        {
            StringBuilder laz=new StringBuilder();
            StringBuilder enc=new StringBuilder();
StringBuilder output=new StringBuilder();

            st = new StringBuilder();
            float laserMeasurement = 0;
            long encoderPos = 0;
            double timeStamp = 0;
            double roll = 0;
            double pitch = 0;
            double accX = 0;
            double accY = 0;
            double accZ= 0;


            int cycles = 0;
            int sectorCounts = 0;
            int measurementCounts = 0;

            listLaseEnc = new PointPairList();
            listTimeDiff = new PointPairList();
            listEncodertime = new PointPairList();
            listEncDiff = new PointPairList();
             AccxEncoder = new PointPairList();
             AccyEncoder = new PointPairList();
             AcczEncoder = new PointPairList();

             PitchEncoder = new PointPairList();
             RollEncoder = new PointPairList();
```

```
rawBytes = File.ReadAllBytes(openFileDialog1.FileName);
double tempTimeStamp = timeStamp = (rawBytes[8] << 16) | (rawBytes[9] << 8)
| (rawBytes[10]);
long tempencoderPos = encoderPos = (rawBytes[4] << 24) | (rawBytes[5] << 16) |
(rawBytes[6] << 8) | (rawBytes[7]);

int size = rawBytes.Length;
progressBar1.Invoke((MethodInvoker)(() => progressBar1.Maximum = size));
int i;
for (i = 0; i < size - 512; )
{
    if (rawBytes[i] == 0 & rawBytes[i + 1] == 0 & rawBytes[i + 2] == 0
&rawBytes[i + 3] == 0 &rawBytes[i + 4] == 0 & rawBytes[i + 5] == 0 & rawBytes[i + 6] == 0
&rawBytes[i + 7] == 0)
    {
        break;
    }
    laserMeasurement = (((rawBytes[i]) << 12) | ((rawBytes[i + 1]) << 8) |
(((rawBytes[i + 2])) << 4) | (((rawBytes[i + 3])))) / 16463f * 19.5f + 5f;
    encoderPos = (rawBytes[i + 4] << 24) | (rawBytes[i + 5] << 16) | (rawBytes[i +
6] << 8) | (rawBytes[i + 7]);
    timeStamp = (rawBytes[i + 8] << 16) | (rawBytes[i + 9] << 8) | (rawBytes[i +
10]);
    cycles++;
    if (cycles == 8)
    {
        pitch = (((rawBytes[i + 11] << 24) | (rawBytes[i + 12] << 16) | (rawBytes[i +
13] << 8) | (rawBytes[i + 14])) / 1000.0);
        roll = (((rawBytes[i + 15] << 24) | (rawBytes[i + 16] << 16) | (rawBytes[i +
17] << 8) | (rawBytes[i + 18])) / 1000.0);
        accX = (((rawBytes[i + 19] << 8) | (rawBytes[i + 20]))*3.3/1023-1.65)/0.3;
        accY = (((rawBytes[i + 21] << 8) | (rawBytes[i + 22])) * 3.3 / 1023 - 1.65) /
0.3;
        accZ = (((rawBytes[i + 23] << 8) | (rawBytes[i + 24])) * 3.3 / 1023 - 1.65) /
0.3;

        i+=14;
        cycles = 0;

    }
    i += 11;
    listEncodertime.Add(timeStamp, encoderPos);
    listTimeDiff.Add(timeStamp, timeStamp - tempTimeStamp); tempTimeStamp
= timeStamp;
    listLaseEnc.Add(encoderPos, laserMeasurement);
    listEncDiff.Add(encoderPos,encoderPos-tempencoderPos);
```

```csharp
                    AccxEncoder.Add(encoderPos,accX);
                    AccyEncoder.Add(encoderPos, accY);
                    AcczEncoder.Add(encoderPos, accZ);

                    PitchEncoder.Add(encoderPos,pitch) ;
                    RollEncoder.Add(encoderPos, roll);


                // laz.Append(laserMeasurement.ToString());
                //enc.Append(encoderPos.ToString());
                output.Append(laserMeasurement.ToString() + " " + encoderPos.ToString() +"
"+ timeStamp.ToString()+ "\r\n");

                if (pitch == 16777.174)
                {
                    MessageBox.Show(rawBytes[i + 11].ToString() + " " + rawBytes[i +
12].ToString() + " " + rawBytes[i + 13].ToString() + " " + rawBytes[i + 14].ToString(), "My
Application",
                        MessageBoxButtons.OKCancel, MessageBoxIcon.Asterisk);
                }


                tempencoderPos =encoderPos;
                measurementCounts++;
                if (i > 0 && measurementCounts % 40 == 0) { sectorCounts++; i += 2; }

                st.Append("Laser: " + laserMeasurement.ToString("0.000") + "      enc: " +
encoderPos.ToString("00000000") + "     time: " + timeStamp.ToString("00000000") + "   Pitch:
" + pitch.ToString("+0.000;-0.000;0") + "   Roll: " + roll.ToString("+0.000;-0.000;0") + "  Accx:
" + accX.ToString("+0.000;-0.000;0") + "   Accx: " + accY.ToString("+0.000;-0.000;0") + "
Accx: " + accZ.ToString("+0.000;-0.000;0") + "\n");



                if (i % 2500 == 0) { backgroundWorker1.ReportProgress(i); }

                //Thread.Sleep(100);
            }

            labelBytesRead.Invoke((MethodInvoker)(() => labelBytesRead.Text = "file size:
" + (rawBytes.Length).ToString() + " bytes\n" + " " + sectorCounts.ToString() + " sectors
translated\naverage sample rate is\n" + measurementCounts / (timeStamp / 1000) + "
samples/sec"));
            //richTextBox1.Invoke((MethodInvoker)(()      =>      richTextBox1.Text      =
st.ToString()));
```

```
            richTextBox1.Invoke((MethodInvoker)(()   =>   richTextBox1.SelectionStart   =
rawBytes.Length));

            progressBar1.Invoke((MethodInvoker)(() => progressBar1.Value = 0));

            laserEncoderPlane.CurveList.Clear();
            TimeDiffPlane.CurveList.Clear();
            encoderTimePlane.CurveList.Clear();
            encoderDiffPlane.CurveList.Clear();
            AccxEncoderPlane.CurveList.Clear();

            PitchEncoderPlane.CurveList.Clear();
            RollEncoderPlane.CurveList.Clear();



            laserEncoderPlane.AddCurve("Porsche",
         listLaseEnc, Color.Red, SymbolType.Diamond);
            TimeDiffPlane.AddCurve("Porsche",
        listTimeDiff, Color.Blue, SymbolType.Diamond);
            encoderTimePlane.AddCurve("Porsche",
         listEncodertime, Color.Green, SymbolType.Diamond);
            encoderDiffPlane.AddCurve("Porsche",
         listEncDiff, Color.Black, SymbolType.Diamond);

            AccxEncoderPlane.AddCurve("Porsche",
        AccxEncoder, Color.Green, SymbolType.Diamond);
            AccxEncoderPlane.AddCurve("Porsche",
        AccyEncoder, Color.Yellow, SymbolType.Diamond);
            AccxEncoderPlane.AddCurve("Porsche",
        AcczEncoder, Color.Blue, SymbolType.Diamond);

            PitchEncoderPlane.AddCurve("Porsche",
        PitchEncoder, Color.Green, SymbolType.Diamond);
            RollEncoderPlane.AddCurve("Porsche",
        RollEncoder, Color.Green, SymbolType.Diamond);
            FileStream   fs1   =   new   FileStream(   openFileDialog1.FileName+"P.txt",
FileMode.OpenOrCreate, FileAccess.Write);
            StreamWriter writer = new StreamWriter(fs1);
        //writer.Write(openFileDialog1.FileName);

            writer.Write(output.ToString());
            writer.Close();


        }
```

```csharp
        private        void        backgroundWorker1_ProgressChanged(object        sender,
ProgressChangedEventArgs e)
        {
          progressBar1.Value = e.ProgressPercentage;
        }

        private void buttonPlot_Click(object sender, EventArgs e)
        {
          zedLazerEncoder.AxisChange(); zedLazerEncoder.Refresh();
          zedLaserTime.AxisChange(); zedLaserTime.Refresh();
          zedEncoderTime.AxisChange(); zedEncoderTime.Refresh();
          zedEncoderDiff.AxisChange(); zedEncoderDiff.Refresh();
          zedAccx.AxisChange(); zedAccx.Refresh();

          zedPitch.AxisChange(); zedPitch.Refresh();
          zedRoll.AxisChange(); zedRoll.Refresh();



        }
      }
    }
```

Program.cs

```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace PostProcessing
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}
```

A.2   ELECTRONICS

Link to electronics source code

A.3   KML GENERATOR

Link to KML Generator

EARLY DESIGN SKETCHES
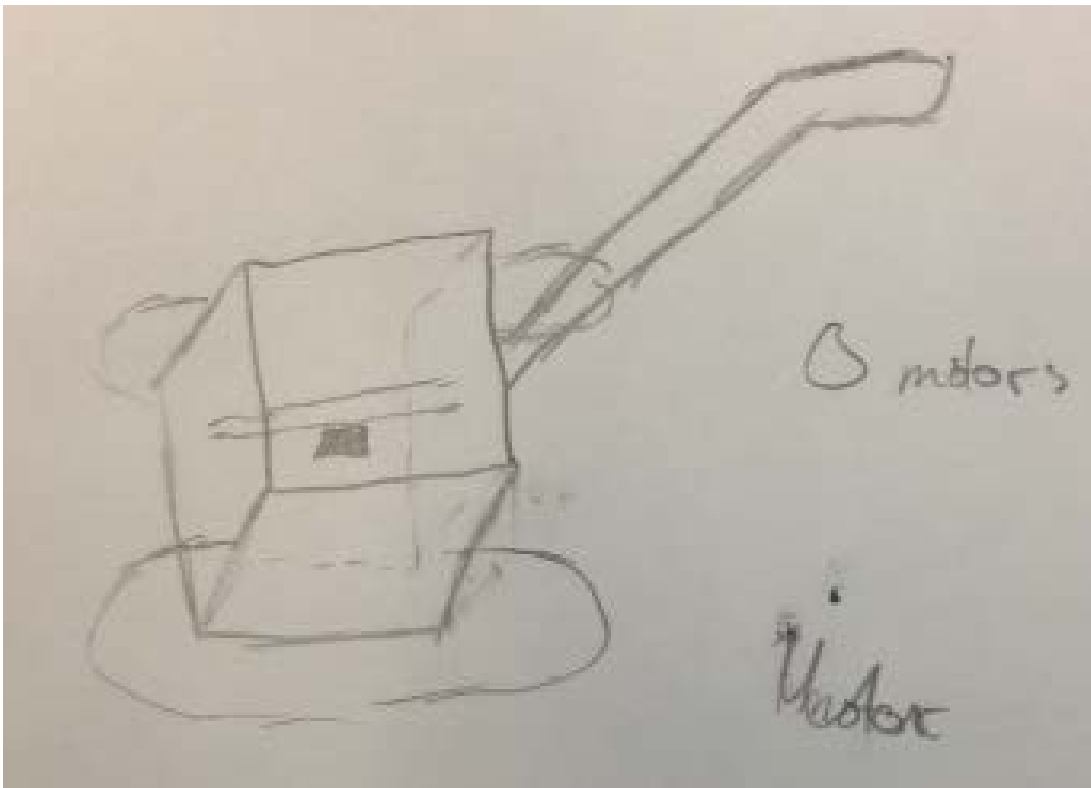
Figure 41: Push device concept similar to PathMeT
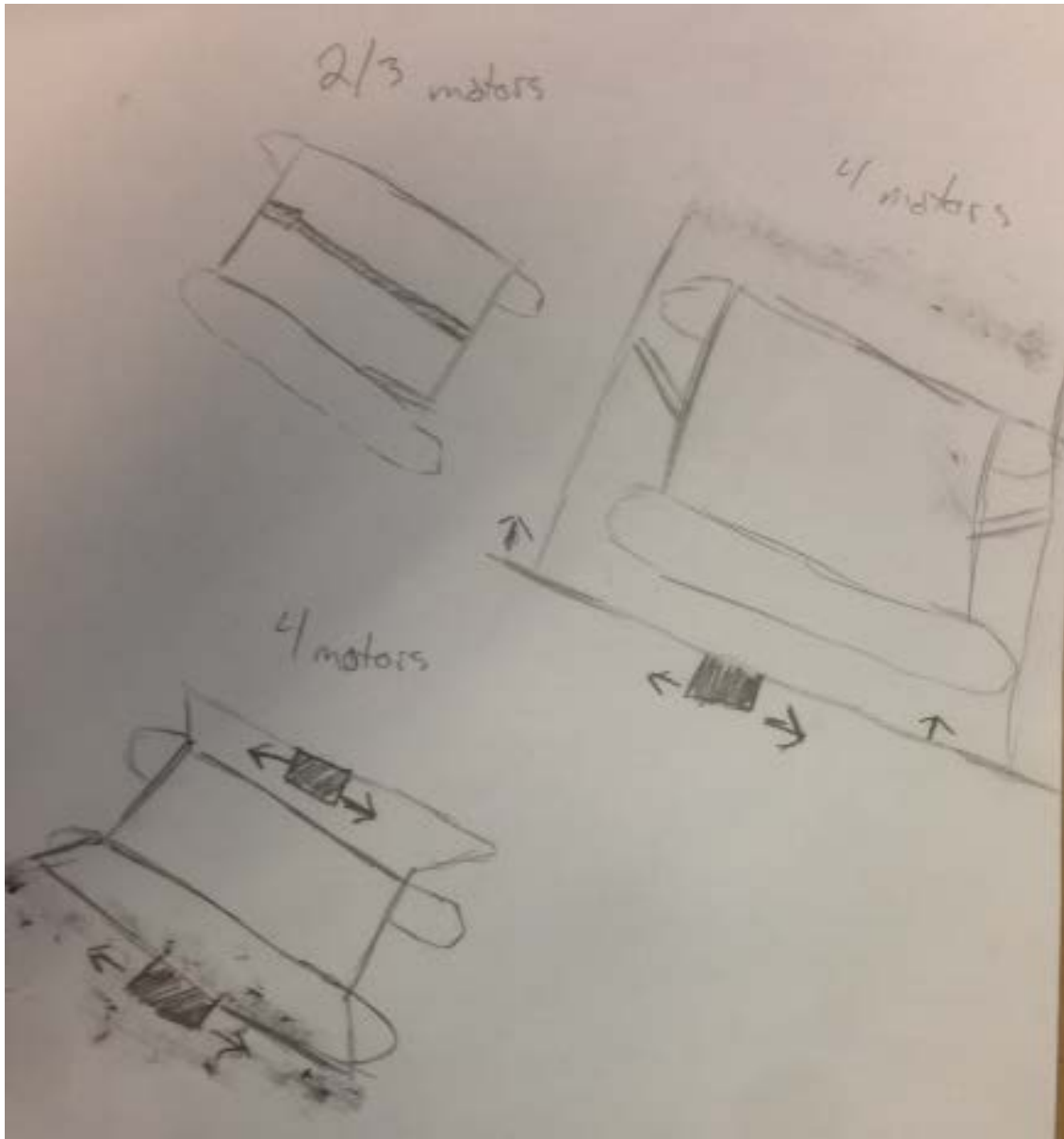
Figure 42: Robotic device concept with varying number of motors to move laser
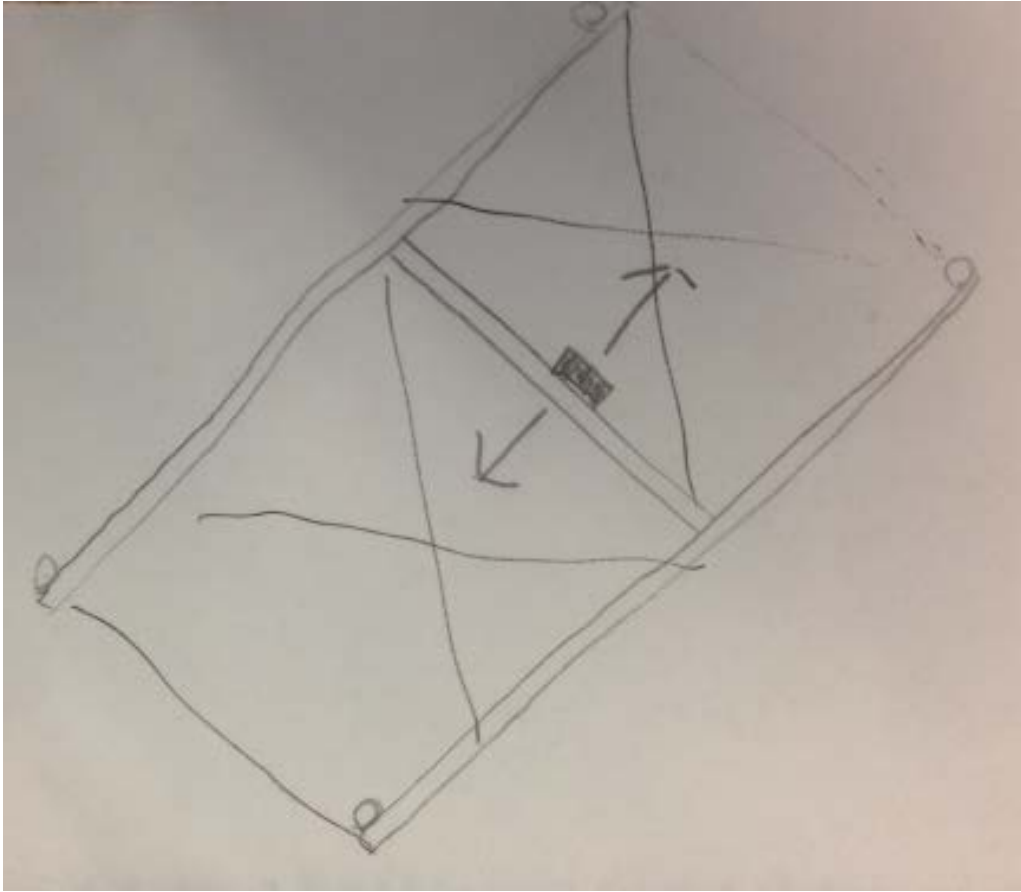
Figure 43: "Bedframe" concept, similar to tracks

ELECTRONICS

C.1   BILL OF MATERIALS

Table 21: Electronics Bill of Materials

			PATHMET BUILD 7	

Bill of Materials

Source Data From:
Project: **PCB7_IanProject.PrjPcb**
                                                                 PCB7_IanProject.PrjPcb
Variant: **None**

Creation Date:	3/26/2014	10:44:55 AM
Print Date:	41726	41726.38886
	Ian McIntyre	IPM4@pitt.edu

Footprint	LibRef	Designator	Description	Qty
Mate-N-Lok_2P	Battery	2	Multicell Battery	1
Voltage Regulator - OKI-78SR Series	Volt Reg	3.3VR, 5VR	Voltage Regulator	2
HDR1X5	Header 5	AccHeader, buffChipProg, CamChipProg, Header, IEncHeader1, IncHeader, MasterProg, REncHeader1, smpProg	Header, 5-Pin	9

Table 21 (continued)

MCHP-SPDIP-SP28	DSPIC33EP128GP502-I/SP	BufferChip	16-Bit Microcontroller and Digital Signal Controller with High-Speed PWM, Op Amps and Advanced Analog, 70 MIPS, 21 I/O, GPIO, -40 to 85 degC, 28-pin SPDIP (SP28), Tube	1
HDR1X2	Header 2	BufferChipLed, LEDCamChip, MotorLimitSwitch, SampChipLED	Header, 2-Pin	4
CAPC1608L	Cap	C1, C2, C9, C10, C11, C12, C13, C14, C15, C18, C19, C20, C21, C22	Capacitor	14
CAPC3225L	Cap	C3, C4, C8, C17	Capacitor	4
1608[0603]	Cap	C5, C6	Capacitor	2
C1210_L	Cap	C7	Capacitor	1
CAPC3225N	Cap	C16	Capacitor	1
C1210_N	Cap Pol1	C23, C24, C25, C26	Polarized Capacitor (Radial)	4
MCHP-SPDIP-SP28	DSPIC33EP128GP502-I/SP	CameraChip	Polarized Capacitor (Radial)	1
HDR1X4	Header 4	CameraHeader, GPSHeader, LaserHeader	Polarized Capacitor (Radial)	3
HDR2X13	Header 13X2	four, one, three, two	Polarized Capacitor (Radial)	4
SOIC127P600-16N	MAX232ACPE	M1, M2	Polarized Capacitor (Radial)	2
MCHP-SPDIP-SP28	DSPIC33EP128MC202-I/SP	Master	Polarized Capacitor (Radial)	1
HDR1X4	Header 4H	MotorHeader	Polarized Capacitor (Radial)	1
AXIAL-0.5	Res2	R1, R2, R3	Polarized Capacitor (Radial)	3
AXIAL-0.3	Res1	R4	Polarized Capacitor (Radial)	1
HDR1X8	Header 8	SD1, SD2, SD3, SD4	Polarized Capacitor (Radial)	4
HDR2X5	MHDR2X5	TFT2x5Header	Polarized Capacitor (Radial)	1
Crystal-AT49	XTAL	Y1, Y2, Y3	Polarized Capacitor (Radial)	3
				67

C.2   SCHEMATICS
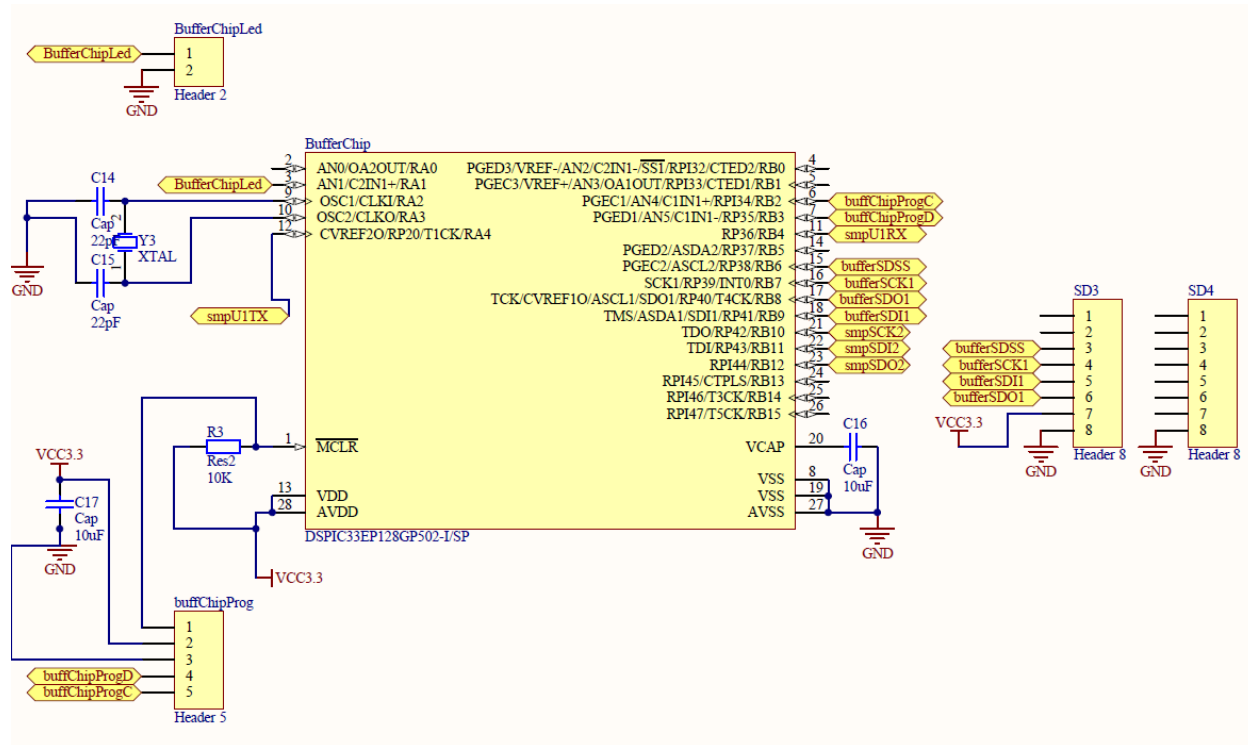
Figure 44: Buffer chip schematic
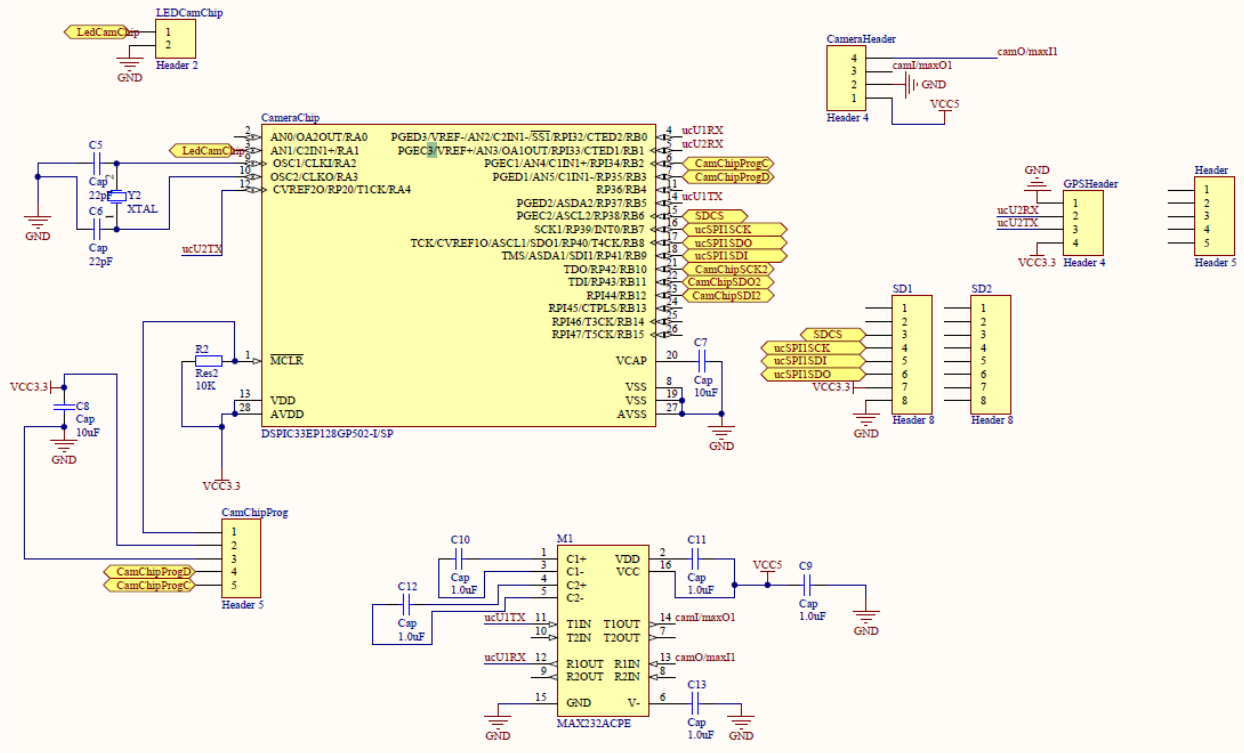
Figure 45: Camera chip schematic
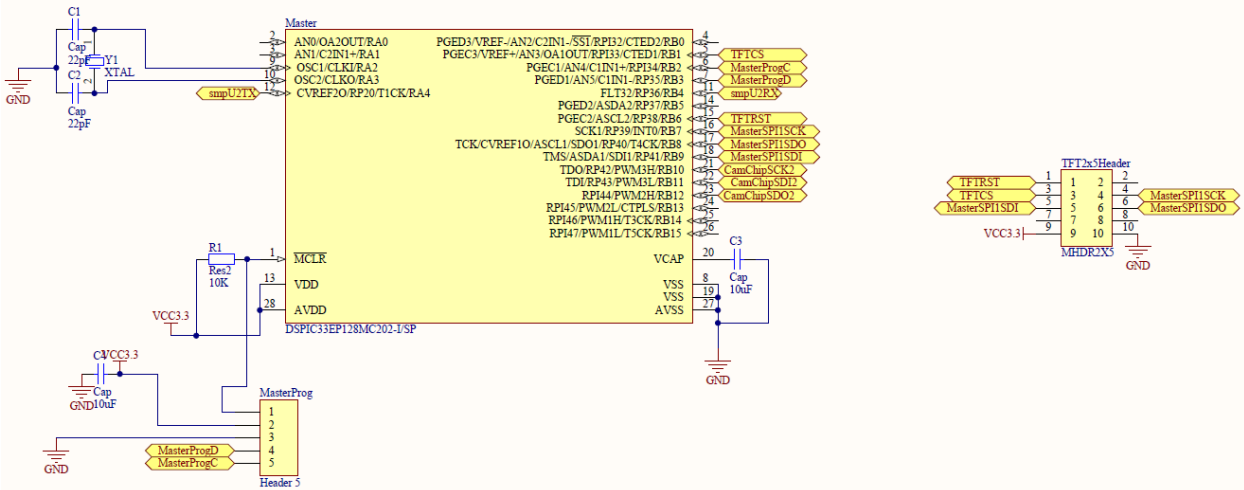
Figure 46: Master chip schematic
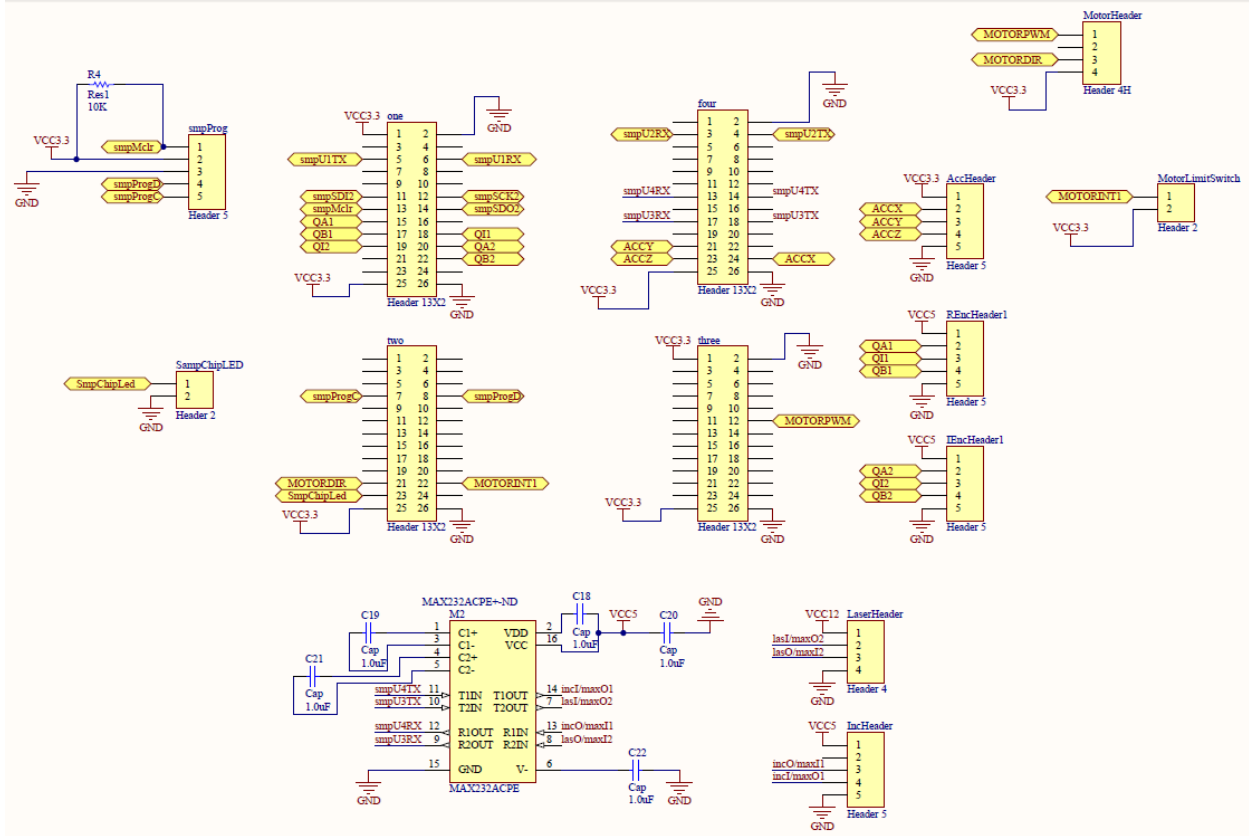
Figure 47: Sensor chip schematic
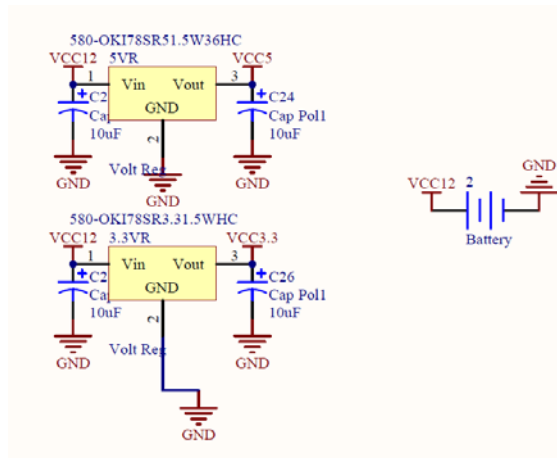
Figure 48: Power supply schematic
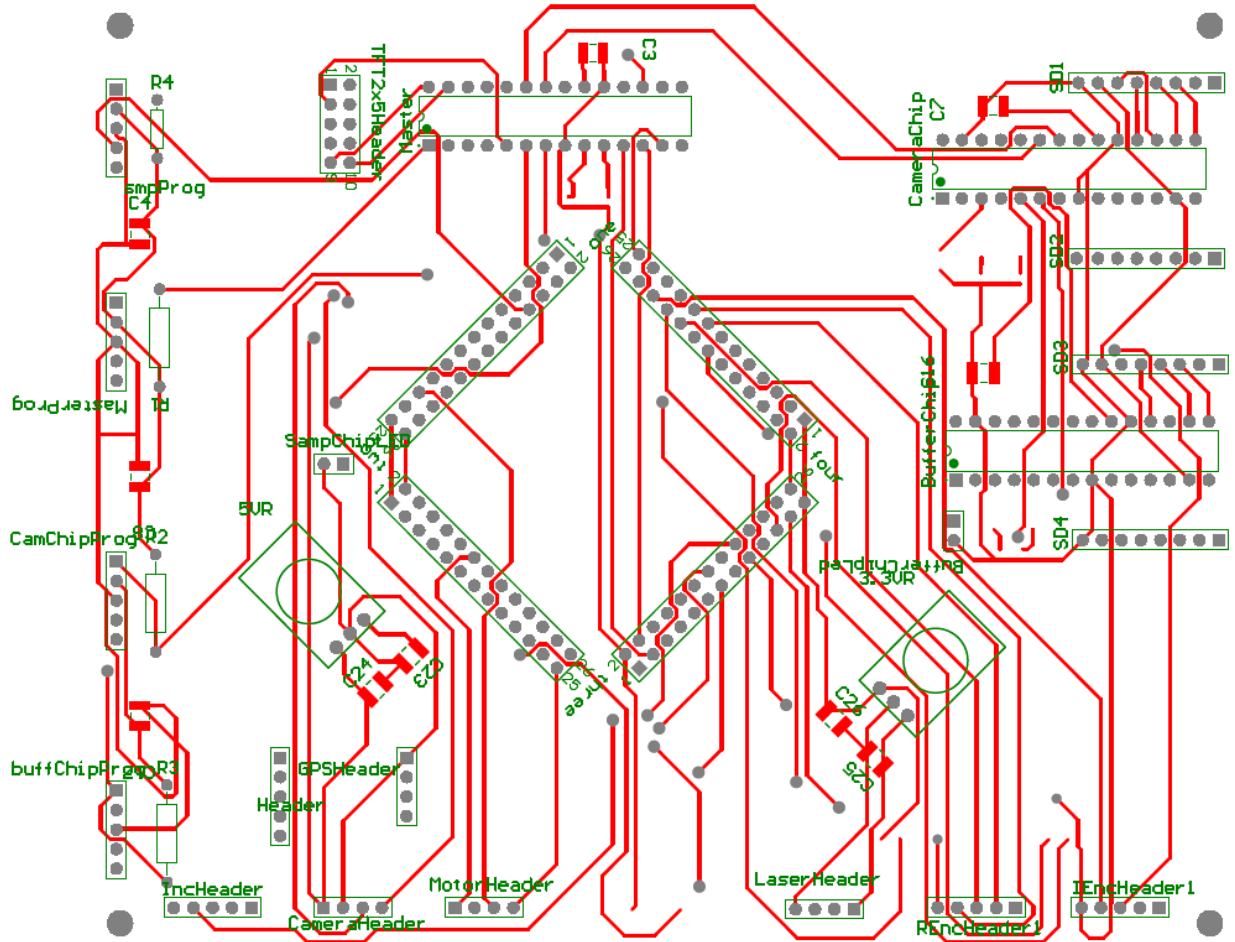
Figure 49: Top layer of printed circuit board
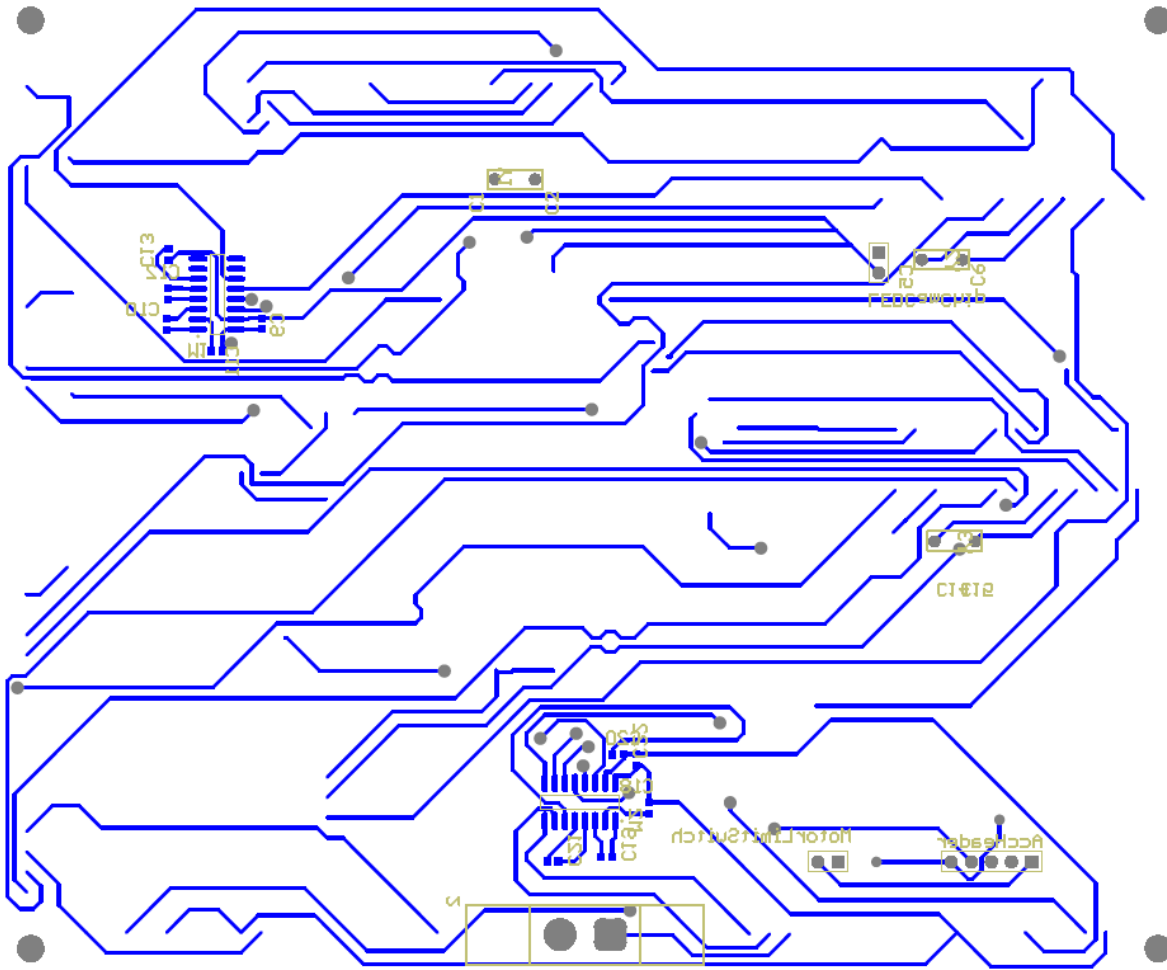
Figure 50: Bottom layer of printed circuit board
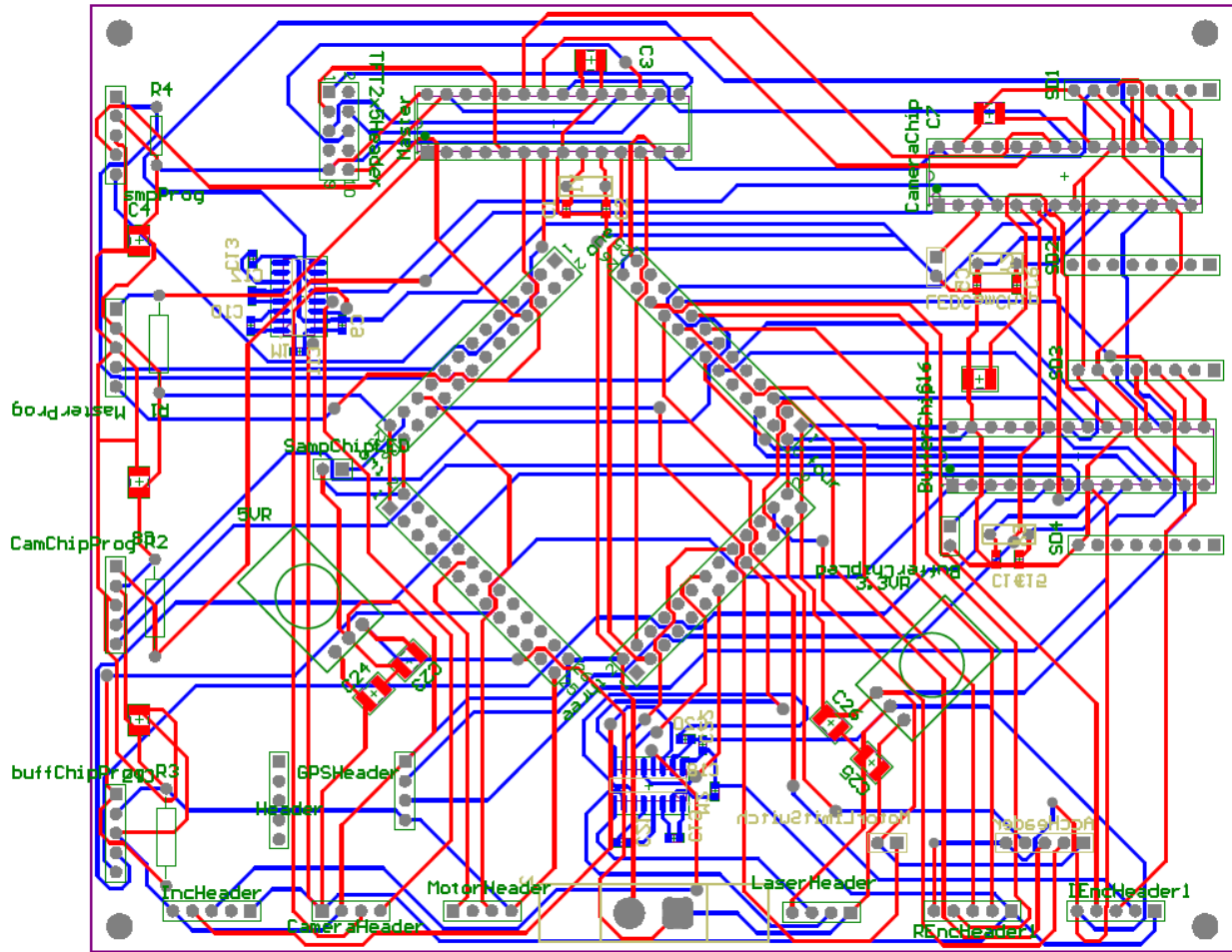
Figure 51: Top (red) and bottom (blue) layer of printed circuit board

APPENDIX D

MECHANICAL DRAWINGS AND MATERIALS

D.1   3D MODEL
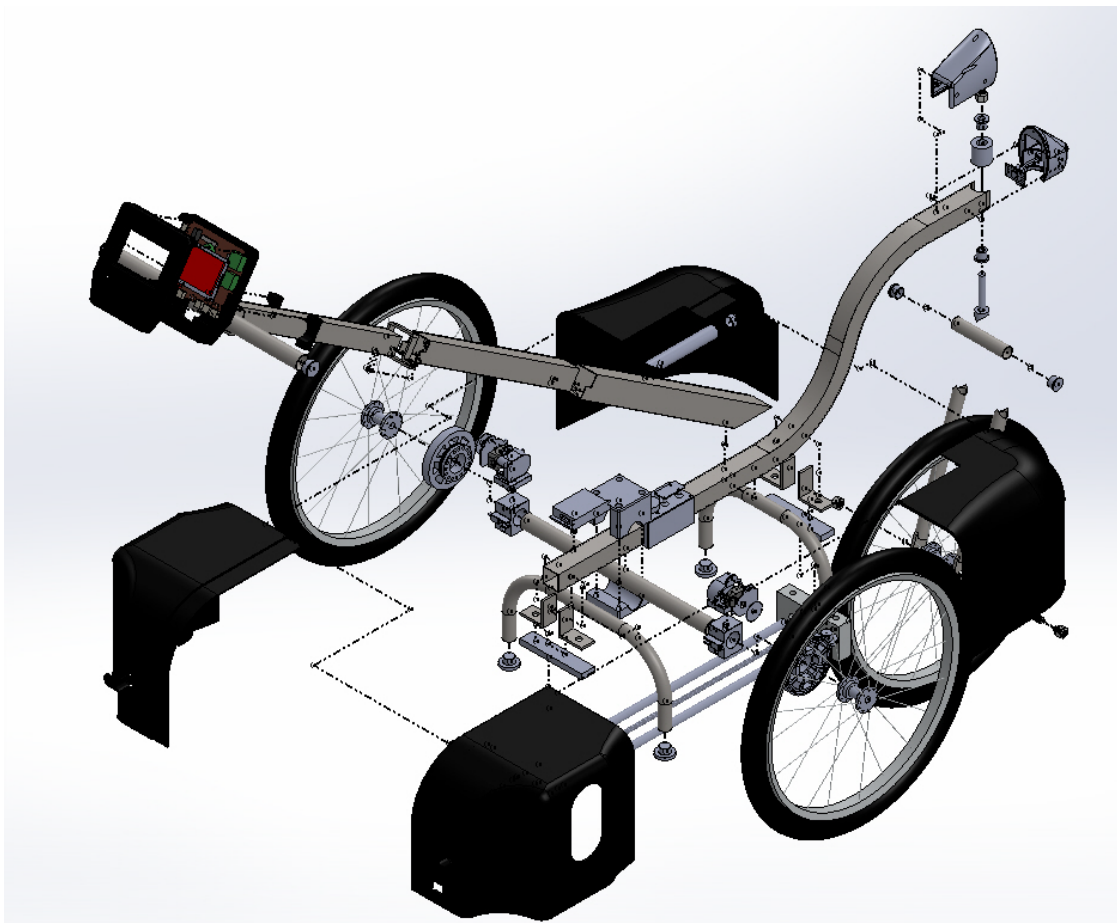
Figure 52: Exploded view of PathMeT

D.2 BILL OF MATERIALS

Table 22: Materials, hardware, and sensor Bill of Materials

Item	Price per unit		Price per 50 or Long Parts		Qty	Subtotal at Volume Price	
PaveTesting Laser	$	980.00	$	980.00	1	$	980.00
S5 Optical Shaft Encoder	$	108.18	$	81.67	2	$	163.34
X3M Multi-Axis Absolute MEMS Inclinometer	$	245.70	$	154.53	1	$	154.53
Venus GPS Logger	$	59.95	$	53.96	1	$	53.96
Color JPEG Camera w/ Infrared	$	49.00	$	44.10	1	$	44.10
MMA7260Q Accelerometer	$	45.00	$	40.00	1	$	40.00
3.4" TFT Proto Board	$	31.00	$	27.90	1	$	27.90
dsPIC33EP512MU810 Microcontroller MCU	$	25.90	$	23.31	2	$	46.62
22" Wheelchair Wheels	$	51.85	$	51.85	3	$	155.55
STR2 Stepper Drive	$	99.00	$	74.25	1	$	74.25
NEMA 17 Step Motor	$	53.00	$	39.75	1	$	39.75
Battery	$	142.00	$	142.00	1	$	142.00
Battery Charger	$	21.95	$	21.95	1	$	21.95
4 ft 1.5"x1.5"x0.65" Square Steel Tube	$	13.28	$	6.84	1	$	6.84
4 ft 2" Square Telescoping Tube	$	20.97	$	19.46	1	$	19.46
4 ft 1.75" Square Telescoping Tube	$	19.57	$	17.61	1	$	17.61
4 ft 1.5" Square Telescoping Tube	$	19.89	$	16.30	1	$	16.30
6 ft 1"OD x 0.065 wall 1020 DOM Steel	$	11.78	$	11.78	1	$	11.78
6 ft 1.25" OD x 0.065 Wall DOM Steel	$	29.70	$	9.90	0.5	$	4.95
3/8"-8 Fast Travel Lead Screw	$	29.90	$	29.90	1	$	29.90
3/8"-8 Plastic Nut	$	26.70	$	26.70	1	$	26.70
24" Lg, 5/8" Diameter Hardened Shaft	$	16.32	$	15.74	2	$	31.48
Nylon Plastic Sleeve Bearings	$	4.43	$	4.43	0.5	$	2.22
Ultra-Flex Double-Loop Shaft Coupling	$	13.98	$	13.98	1	$	13.98
Double Sealed Ball Bearing	$	8.38	$	8.38	1	$	8.38
2 ft 2"x2"x1/8" Steel Angle	$	4.94	$	2.96	0.2	$	0.59
1 ft 6" x 6" Aluminum Square	$	258.16	$	152.42	0.083	$	12.65
Total						**$**	**2,146.79**

D.3   RAPID PROTOTYPED DRAWINGS

114

Top drawing (Encoder_Gear):

.433
.123
.014
Ø.089 X3
Ø.508
.014
R.031
R.771
R.723
R.008
Ø.508
.235
Ø.235

Title block (top):

UNLESS OTHERWISE SPECIFIED:		NAME	DATE		
DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL ± ANGULAR: MACH ± BEND ± TWO PLACE DECIMAL ± THREE PLACE DECIMAL ±	DRAWN			TITLE:	
	CHECKED				
	ENG APPR				
	MFG APPR				
	Q.A				
INTERPRET GEOMETRIC TOLERANCING PER:	COMMENTS:				
MATERIAL					
PROPRIETARY AND CONFIDENTIAL THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.	FINISH		SIZE	DWG. NO.	REV
	NEXT ASSY	USED ON	A	Encoder_Gear	
APPLICATION	DO NOT SCALE DRAWING	SCALE: 1:1	WEIGHT:	SHEET 1 OF 1	

Bottom drawing (Wheel_Gear):

R.920
R1.300
R1.799
R1.950
Ø4.500
1.200
.700
.550
.250
.200
120X .024
R.025
Ø.235
R1.150
Ø2.750

Title block (bottom):

UNLESS OTHERWISE SPECIFIED:		NAME	DATE		
DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL ± ANGULAR: MACH ± BEND ± TWO PLACE DECIMAL ± THREE PLACE DECIMAL ±	DRAWN			TITLE:	
	CHECKED				
	ENG APPR.				
	MFG APPR.				
	Q.A.				
INTERPRET GEOMETRIC TOLERANCING PER:	COMMENTS:				
MATERIAL					
PROPRIETARY AND CONFIDENTIAL THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.	FINISH		SIZE	DWG. NO.	REV
	NEXT ASSY	USED ON	A	Wheel_Gear	
APPLICATION	DO NOT SCALE DRAWING	SCALE: 1:2	WEIGHT:	SHEET 1 OF 1	

115

Ø.492
Ø.745
Ø.555
Ø.745
R.010
3.474 1.407 1.250
R.100
Ø.142
.598
3.512
6.000

UNLESS OTHERWISE SPECIFIED:		NAME	DATE		
DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL ± ANGULAR: MACH ± BEND ± TWO PLACE DECIMAL ± THREE PLACE DECIMAL ±	DRAWN			TITLE:	
	CHECKED				
	ENG APPR				
	MFG APPR				
INTERPRET GEOMETRIC TOLERANCING PER:	Q.A				
	COMMENTS:				
MATERIAL					
				SIZE A	DWG. NO. laser	REV
NEXT ASSY	USED ON	FINISH		SCALE:12	WEIGHT:	SHEET 1 OF 1
APPLICATION		DO NOT SCALE DRAWING			

.025
.500
R.010
R.010
.250
R.010

Ø.819 Ø.869 Ø.889 Ø1.480 Ø1.500

UNLESS OTHERWISE SPECIFIED:		NAME	DATE		
DIMENSIONS ARE IN INCHES TOLERANCE: FRACTIONAL± ANGULAR: MACH± BEND± TWO PLACE DECIMAL ± THREE PLACE DECIMAL ±	DRAWN			TITLE:	
	CHECKED				
	ENG APPR.				
	MFG APPR.				
	Q.A.				
	COMMENTS:				
				SEE DWG. NO. Kickstand Leg Cap A	REV
NEXT ASSY	USED ON	FINISH		SCALE:2:1	WEIGHT:	SHEET 1 OF 1
APPLICATION		DO NOT SCALE DRAWING			

D.4   METALS DRAWINGS

R8.750

6.000

R7.250

.992

20.500

1.500 1.370

1.500 1.370

R.750

41.874

UNLESS OTHERWISE SPECIFIED:		NAME	DATE	
DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL ± ANGULAR: MACH ± BEND ± TWO PLACE DECIMAL ± THREE PLACE DECIMAL ±	DRAWN			TITLE:
	CHECKED			
	ENG APPR.			
	MFG APPR.			
INTERPRET GEOMETRIC TOLERANCING PER:	Q.A.			
MATERIAL	COMMENTS:			
FINISH				SIZE DWG. NO. REV

NEXT ASSY   USED ON
APPLICATION   DO NOT SCALE DRAWING

ASquare_Tube

SIZE DWG. NO.   REV
SCALE: 1:12 WEIGHT: SHEET 1 OF 1

5   4   3   2   1

6.500

4.989

Ø1.000

Ø.870

33.50°

UNLESS OTHERWISE SPECIFIED:		NAME	DATE	
DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL ±	DRAWN			
	CHECKED			TITLE:

1/2-13 Tapped Hole

Ø.625
Ø.266
Ø.100
Ø1.000

118.00°

.980 .750

3.000

1.500
2.390
3.000
3.610
4.500

Ø.463
Ø.266

.356

1.500
2.117
2.369
3.631
3.883
4.500

.165
.250
.900
6.000
5.100
1.250 1.000

.165
.250
6.000
1.250 .910

Ø.422
Ø.500

Ø.625
Ø.266
Ø.688
118.00°

.980 .750

3.000

1.500
4.500

Ø.463
Ø.266

1.500
4.500

1/2-13 Tapped Hole

UNLESS OTHERWISE SPECIFIED:		NAME	DATE			
DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL± ANGULAR: MACH± BEND ± TWO PLACE DECIMAL ± THREE PLACE DECIMAL ±	DRAWN					
	CHECKED			TITLE:		
	ENG APPR					
	MFG APPR					
INTERPRET GEOMETRIC TOLERANCING PER:	Q.A.					
MATERIAL	COMMENTS:					
				SIZE	DWG. NO.	REV
FINISH	NEXT ASSY	USED ON		A	End_Block1	

118

Ø1.000
Ø.870
11.125
R.500

.375
.625
.375
Ø1.000

UNLESS OTHERWISE SPECIFIED:		NAME	DATE			
DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL ± ANGULAR: MACH ± BEND ± TWO PLACE DECIMAL ± THREE PLACE DECIMAL ±	DRAWN			TITLE:		
	CHECKED					
	ENG APPR.					
	MFG APPR.					
INTERPRET GEOMETRIC TOLERANCING PER:	Q.A.					
MATERIAL	COMMENTS:					
FINISH			SIZE A	DWG. NO. front yoke side		REV
DO NOT SCALE DRAWING			SCALE: 1:4 WEIGHT:		SHEET 1 OF 1	

NEXT ASSY   USED ON
APPLICATION

5   4   3   2   1

119

TITLE:

Kickstand_Leg

SCALE: 1:4   WEIGHT: SHEET 1 OF 1

TITLE:

Kickstand_Leg_Back

SCALE: 1:4   WEIGHT: SHEET 1 OF 1

120

Square tube clamp drawing.

Dimensions: 3.070, 2.320, 2.170, .150, R.180, 2.215, 2.065, .250, .150, .400, R.030, R.040, 2.695, .625, Ø.250

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ±   BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH

NAME   DATE
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:

TITLE:

SIZE DWG. NO.   REV
A
square tube clamp, 2.00
SCALE: 1:1 WEIGHT: SHEET 1 OF 1

NEXT ASSY   USED ON
APPLICATION   DO NOT SCALE DRAWING

Yoke wheel part drawing.

Dimensions: 2.000, 1.606, 1.106, R.150, .513, 1.300, R.150, .250

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ±   BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH

NAME   DATE
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:

TITLE:

SIZE DWG. NO.   REV
A
yoke wheel part
SCALE: 2:1 WEIGHT: SHEET 1 OF 1

NEXT ASSY   USED ON
APPLICATION   DO NOT SCALE DRAWING

121

APPENDIX E

EARLY DESIGN BRIEF

PathMeT

OVERVIEW

Our goal is to develop a Pathway Measurement Tool (PathMeT) that characterizes the flatness, running slope, cross slope, lippage and roughness of pedestrian pathways. PathMeT is being designed for stakeholders involved in designing, constructing, and evaluating pathways such as city planners, engineers, contractors and architects. Our design objectives are as follows:

1. PathMeT will be portable enough so that it can be transported in the trunk of a typical automobile.
2. PathMeT will be accurate enough to measure pathway profiles at a length scale of 1mm.
3. In addition to measuring pathway profiles, PathMeT will also record photographs of the surface and GPS coordinates, allowing for photographic and geospatial locating of pathways for the purposes of infrastructure management.
4. PathMeT data will be compatible with Google maps and ProVAL.

Table 23 shows a timeline for product development.

Table 23: Development timeline

Task	Sep-Dec	Jan-Apr	May-Jun	July-Aug
Design	X			
Fabrication		X		
Testing			X	
Final Validation				X

The following is a detailed description of the mechanical, software, and electrical design of PathMeT:

The mechanical design of PathMeT, shown in Figure 53 and Figure 54, includes a light-weight tubular frame with three wheels. The two rear wheels are attached to the back axle but can be removed for increased portability. The front wheel is a caster, which allows PathMeT to easily make sharp turns and navigate corners without lifting any wheels off the ground. This front wheel can also be removed for increased portability. The frame will be constructed of two pieces of round aluminum tubing. Round aluminum tubing is light and is easily bent with our current tooling, which allows for a highly customized design. Figure 55 shows a laser that is located in a box in the center of the frame. PathMeT has two modes—rolling (RM) and inch-worm (IM). When PathMeT is in the RM, a user will push it along the pathway to be characterized while data is being collected. In the IM, PathMeT remains stationary while a laser moves along a track housed within the tubular frame of PathMeT. An encoder will be attached at both rear wheels to record how far PathMeT has traveled and whether it is being pushed straight or around a turn. An inclinometer and camera will also be attached to the laser enclosure (Figure 55); this will help determine the cross slope and running slope of the pathway PathMeT is traveling on. A GPS will be attached on the platform above the laser. The battery that operates all of the sensors and on-board computer is located toward the rear of the frame on a platform above the laser. The attachment pivot of the push-handle will be located toward the middle of the device, as shown in Figure 53. This location will help prevent the user from accidentally lifting the front wheel, which may induce errors in PathMeT's measurements. This location also allows the handle to be retracted in a direction that reduces the length and height of PathMeT, helping with portability. A digital touch-screen (Figure 54) is attached to the push-handle and is the user interface for PathMeT's data collection system, which is described below. The frame and all of

the sensors will be enclosed by a resilient shroud that is not shown in the figures. The shroud will keep sunlight away from the laser and prevent inclement weather from damaging any of the sensors. PathMeT is designed to fit in the trunk of a car. Table 24 shows target specifications for PathMeT.

Table 24: PathMeT mechanical specifications

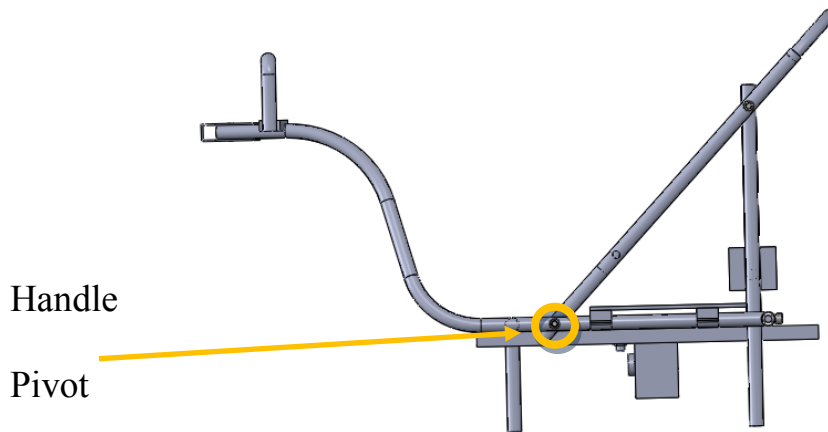| **Material Type**	Aluminum
Target Weight (Disassembled)	25 lbs.
Target Weight (Assembled)	40 lbs.
Target Physical Dimensions (disassembled & collapsed push-handle)	36" L x 25" W x 20" H
Target Physical Dimensions (assembled)	60" L x 25" W x 36" H

Figure 53: Side view of PathMeT without wheels, ready to transport

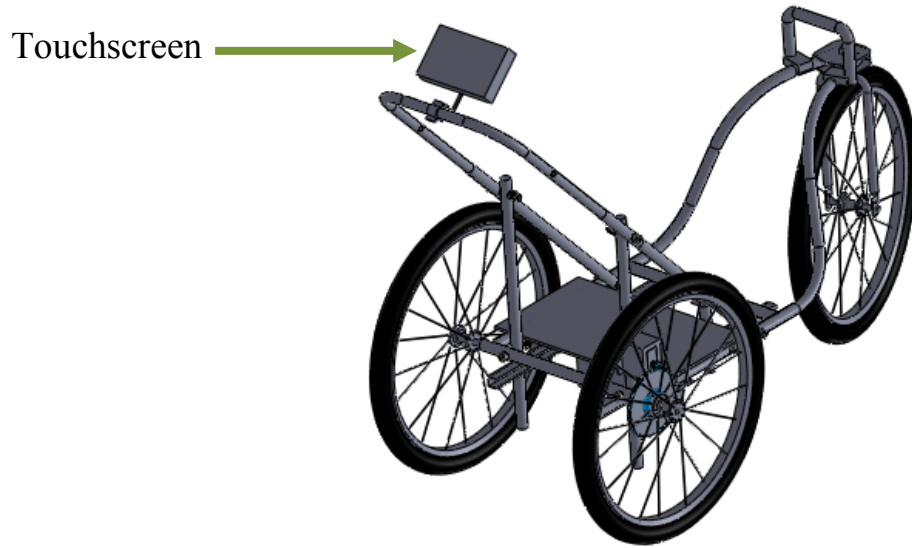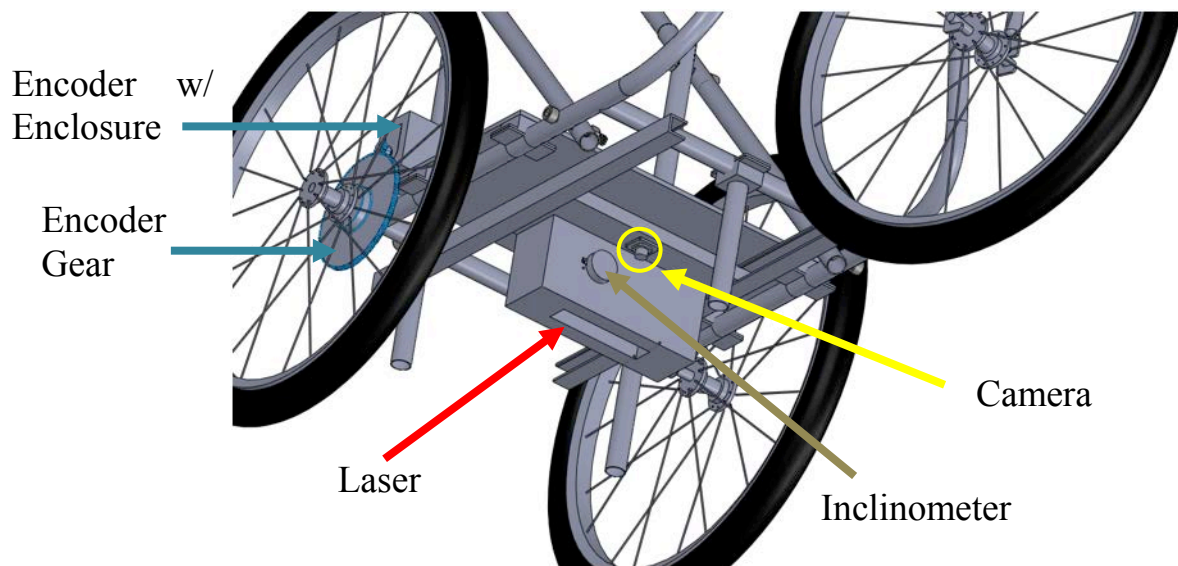**Figure 54: Isometric view of PathMeT assembled and ready for data collection**

Figure 55: Underneath view of PathMeT components

PathMeT will be programmed by using the C# language, which is the most common programming language for PC-users. A JRexPlus single-board computer made by Kontron will be used. It has numerous ports that accommodate PathMeT's many sensors. The operating system that will be used has not been finalized. However, it will likely be Windows 7 or Windows XP, which are both widely available on the commercial market.

General Graphical User Interface (GUI)

The GUI (Figure 56) will allow for easy operation while providing real-time feedback on sensor data. The GUI will allow the user to switch between the RM and IW modes of operation, which are described in detail below. Speed, profile, odometer, Pathway Roughness Index (PRI), running slope, cross slope, and pathway photograph will be displayed on the GUI. A list of the inputs and outputs are displayed in Table 25. The PRI is a roughness index that is based on a computer-simulated path of a solid 2.5-inch wheel as it travels over a surface profile. The roughness index is calculated as the cumulative sum of vertical deviations of the wheel path normalized by the distance travelled.

Table 25: Computer inputs and outputs

User Inputs:	Sensor Inputs:	Real Time Outputs:	Delayed Outputs:
Mode	Encoder (x2)	Distance	Roughness Index
Comments	GPS	Speed	Roughness Profile
Initialize	Laser	Surface Photograph	Pathway Data File
Start/Stop	Inclinometer	Cross Slope	
Date/Time	Camera	Running Slope	
		Roughness Profile	

Speedometer: The speedometer is designed as a semicircular array of lights. The center of the array represents the ideal speed at which PathMeT should travel. Only the green light is lit when travelling at that ideal speed. The far left and right represent going too slow and too fast, respectively. These lights will light up red when travelling at these limits. All other lights represent an acceptable speed in between the ideal speed and the limits. Only one light is lit at a time.

Profile: As the user propels PathMeT over a surface, a real-time down-sampled plot of the pathway profile is displayed. Once a trial has been completed, a full profile of the pathway surface will be displayed.

Odometer: In RM, the odometer will count up until data collection has stopped. In IM, the odometer is not needed. Furthermore, the odometer is used to count down to the next flagged spot of interest. These flagged spots are described below in the RM. The odometer will tell the user how far back to the spot of interest. The user will travel backwards along the initially travelled path until the odometer reads "0 ft" and turns green. (A further explanation of use is explained in the Rolling mode).

PRI, Running Slope, Cross Slope: At the end of a run, the screen will display the maximum and mean PRI, running slope, and cross slope measurements.

Photograph: A photograph of the surface will be displayed on the screen to assure accurate data. The picture will change periodically as PathMeT is propelled over the surface.

The date and time not only provide the user with the current date and time, but this is the method by which each file will be saved. The comments section is available to add any extra

information about the surface, location, etc. The initialize button tells the computer to begin streaming data, but without saving it to a specific file. This part of the process is done only after the mode is selected, comments are added, and the user is ready to start collecting data. Pressing initialize will create the file to which that specific trial will be saved. In addition, after PathMeT is initialized, the user will be able to see if any errors are present and check that each appropriate sensor is collecting data. Finally, the start/stop button tells the computer to prepare to begin data collection. Data collection does not begin until the user begins propelling PathMeT.
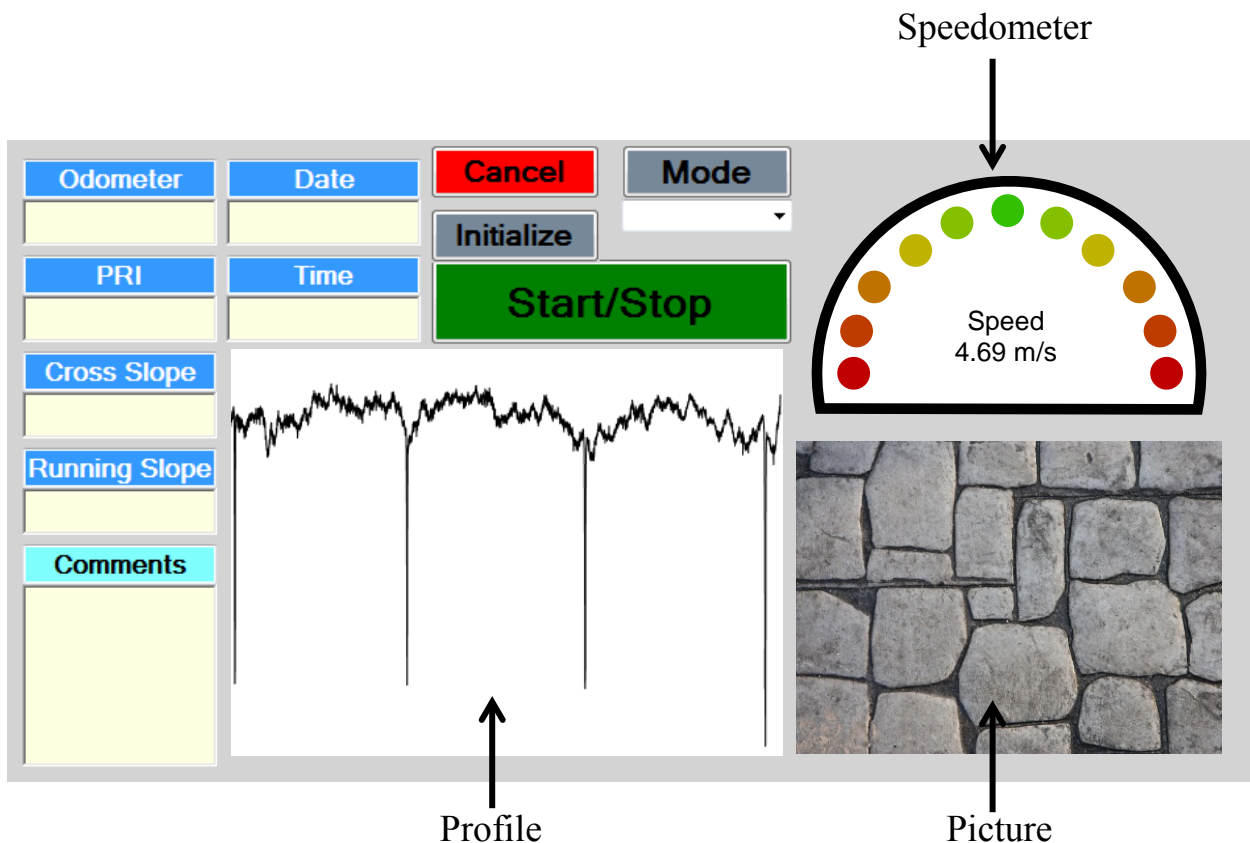

Figure 56: PathMeT Graphical User Interface (GUI)

Basic Operation:

PathMeT is operated via the touch screen mounted to its handle (Figure 54). A main power switch (not shown) will be used to start the system. Once the system is started, the user will select the mode he or she wishes to use, and any comments about the path they are about to

evaluate. When the user is ready to start data collection, he or she presses initialize. This signals the computer to create the file and folder, save the comments, and begin streaming the data onto the screen. If there are any problems with the creation of the file or the streaming of the data, all buttons on the GUI will turn red signifying an error. The user fixes the problem and then presses the initialize button again. If there are no errors, the screen remains unchanged and data collection can begin. At the end of the run, the PRI, cross slope, and running slope data will be displayed.

Mode 1: Rolling Mode (RM)

RM allows the user to push PathMeT continuously. This mode allows for a large amount of data to be collected in a relatively short period of time. While data is continuously collected, the user is alerted of pathway segments that fall out of compliance with standards or are rough enough to introduce errors to the continuous data collection; these data from these segments will need to be recollected using the IM mode outlined below. When PathMeT is stopped, each rough segment is displayed in a queue. The queue displays the distance the user must backtrack in order to recollect the data for each segment. The user can either decide to recollect the data or ignore the error.

Figure 58 shows a flow chart for operation within the RM. If the user decides to recollect that data, he/she will need to backtrack the original path travelled. As the user moves backwards, the odometer counts down until it read "0 ft" for the first error listed. The errors are ordered from newest to oldest. The user then applies the break and runs the IM. Then, the user has the option to ignore the next error, measure the next flagged segment, or continue data collection as normal. This process is repeated for each segment that displayed an error in the queue.

Mode 2: Inchworm Mode (IM)

IM is an autonomous method of data collection. This mode is used for pathway segments that appear to be unacceptable or if PathMeT is unable to collect high quality data using RM. IM is characterized by the laser moving while PathMeT remains stationary. While PathMeT sits on a rough surface segment, the laser collects data by moving parallel to PathMeT on a motorized track. The laser collects data as it travels in both directions. Once the laser returns to its original position, data collection is complete.

A conceptual diagram of the difference between the Rolling and Inchworm modes is displayed in Figure 57.

Data File and Analysis

After a successful trial is completed, the following information will be saved to the respective pathway file: laser data, encoder data, inclination data, total distance travelled, average speed, GPS data (longitude, latitude, altitude), and picture/pictures of the surface while PathMeT was moving. After data collection, ProVal will be used to analyze the data and display the final roughness index. Data will be stored in such a way that allows for immediate analysis by ProVal. Similarly, data will be uploaded to Google Maps to store pathway roughness. On Google Maps, one can add pinned locations that provide information about pathways, including pictures and roughness. This can be useful for disseminating information about accessible pathways, and support infrastructure management.
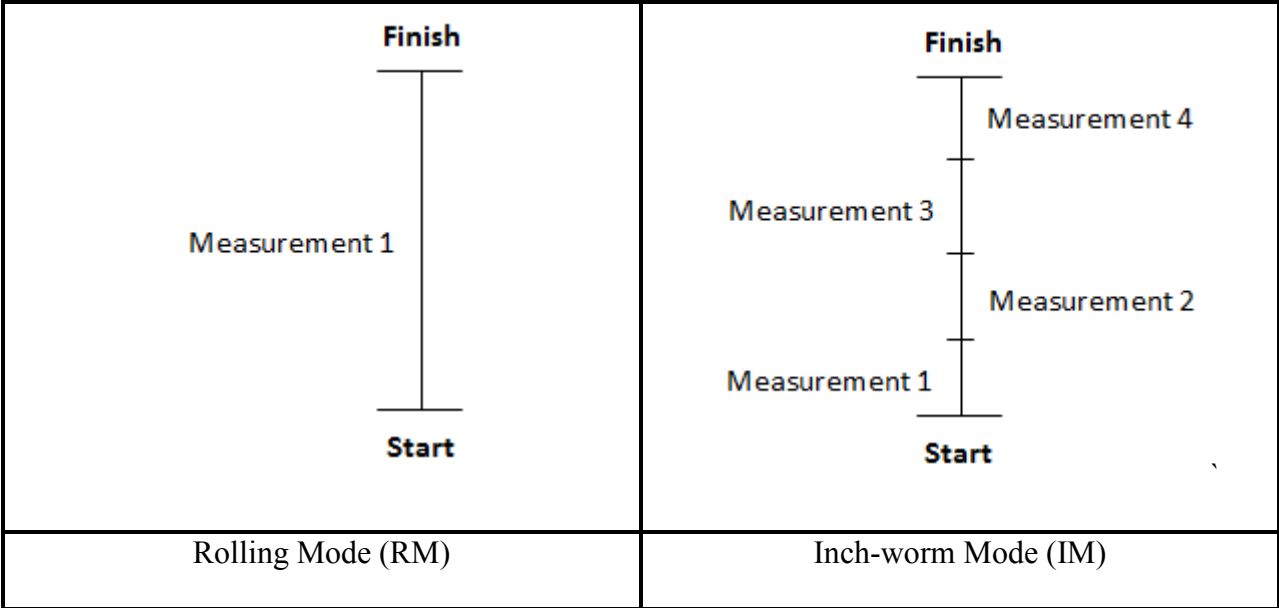
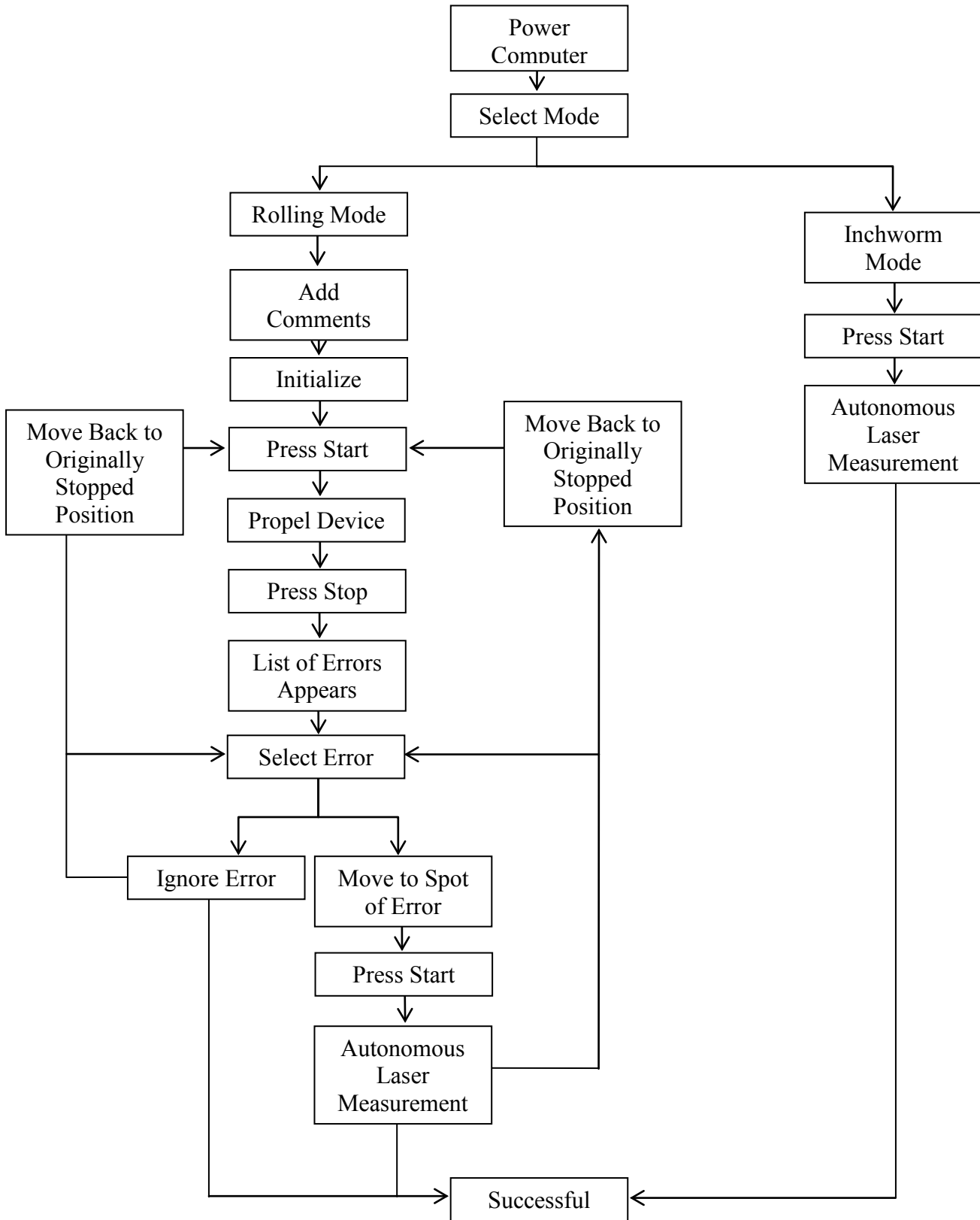**Figure 57: Comparison between Rolling and Inchworm modes**

Figure 58: Flow chart of user operation within each mode

Numerous sensors are included in the design of PathMeT. A list of these sensors, along with the computer and power source can be seen in Table 26. The current power source is a 12V, 18Ah battery from Braille Battery that only weighs 2.3 pounds. All components will be powered from this source. PathMeT will run continuously with all sensors running for an estimated five hours with this battery selection.

Figure 59 shows a schematic of all the sensors and their connections to the computer. The touchscreen is not included in this schematic, but it connects to a VGA and USB port in the computer. The schematic shows seven sensors circled in red. In addition, there are five connectors circled in blue. These are the connections from the sensors to the computer via USB or RS-232 Serial port. The connections that are circled in green are general-purpose input/output (GPIO) connections to the computer. The Bill of Materials can be found in Table 27.

Table 26: List of electrical design components

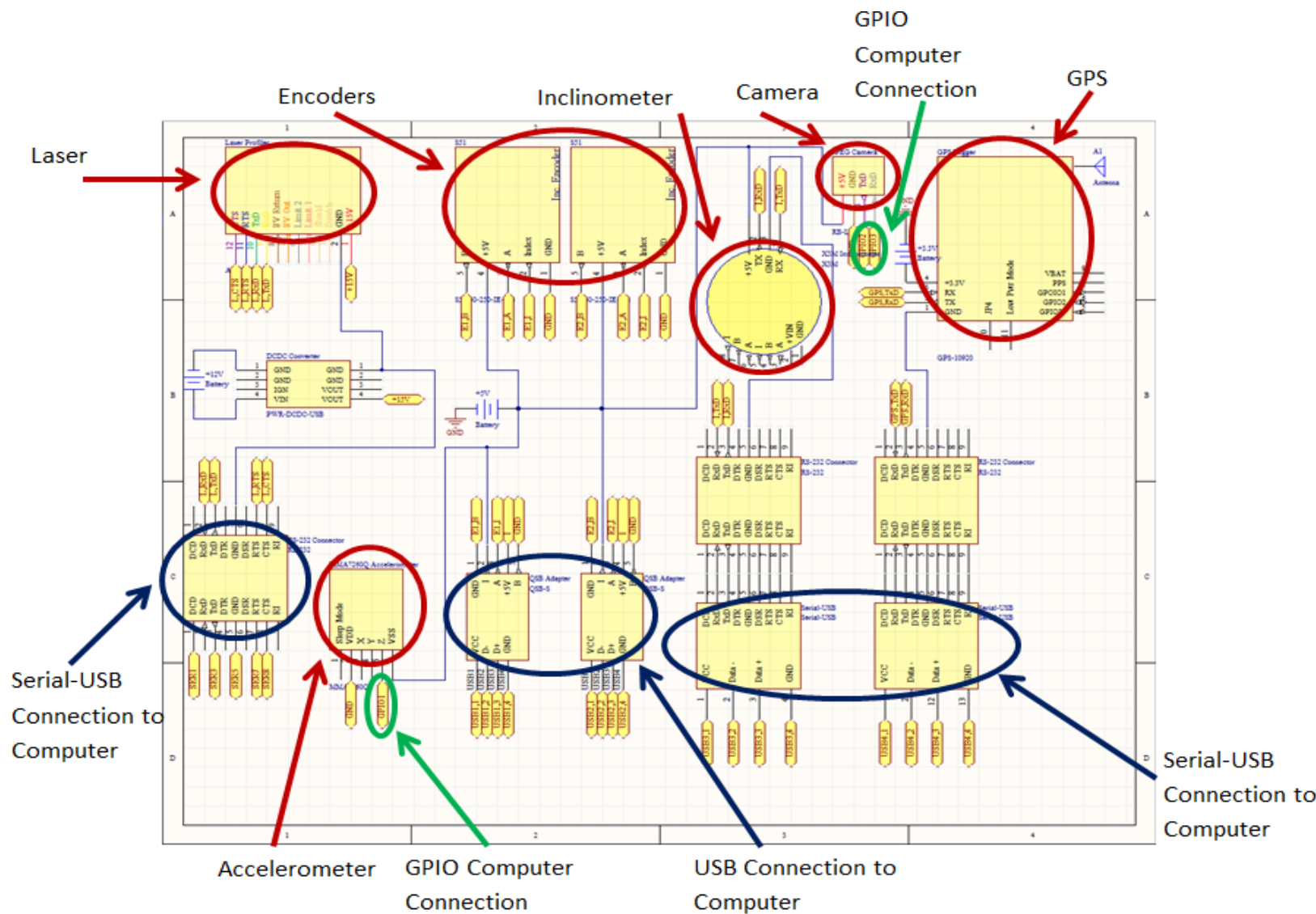| Component	Vendor
AR700-RP Laser	Acuity
S5 Optical Shaft Encoder	US Digital
X3M Multi-Axis Absolute MEMS Inclinometer	US Digital
Venus GPS Logger	Sparkfun
Color JPEG Camera w/ Infrared	RobotShop
MMA7260Q Accelerometer	Freescale
7" TFT LCD Touchscreen Monitor with VGA & AV Inputs	XENARC
JRexplus-DC Single Board Computer	Kontron
G9 Green Lite Lithium 12 V 18 Ah Battery	Braille

Figure 59: Schematic of PathMeT sensors and their connections to the computer

BILL OF MATERIALS

Table 27: Bill of Materials for PathMeT components.
Volume pricing is based on 50 units.

Component	Vendor	Unit Price	Volume Price	Qty	Subtotal at Volume Price
AR700-RP Laser	Acuity	$995.00	$796.00	1	$796.00
S5 Optical Shaft Encoder	US Digital	$108.18	$81.67	2	$163.34
X3M Multi-Axis Absolute MEMS Inclinometer	US Digital	$245.70	$154.53	1	$154.53
Venus GPS Logger	Sparkfun	$59.95	$53.96	1	$53.96
Color JPEG Camera w/ Infrared	RobotShop	$49.00	$44.10	1	$44.10
MMA7260Q Accelerometer	Freescale	$45.00	$40.00	1	$40.00
7" TFT LCD Touchscreen Monitor with VGA & AV Inputs	XENARC	$315.00	$315.00	1	$315.00
JRexplus-DC Single Board Computer	Kontron	$294.00	$279.00	1	$279.00
24" Wheelchair Wheels	New Solutions	$55.00	$55.00	3	$165.00
Flat Free Tire Inserts	Kenda	$15.00	$15.00	3	$45.00
Linear Actuator with Encoder	Thomson Linear	$350.00	$350.00	1	$350.00
G9 Green Lite Lithium 12 V 18 Ah Battery	Braille	$209.00	$209.00	1	$209.00
1236L Battery Charger	Braille	$149.00	$149.00	1	$149.00
DCDC Converter w/ USB	Mini-Box	$59.95	$54.95	1	$54.95
Picopsu-80 DCDC Power Supply	Mini-Box	$25.00	$19.95	2	$39.90
Total					**$2858.78**

BIBLIOGRAPHY

ADA. (1990). Americans with Disabilities Act. Washington, D.C.

ADAAG. (2002). Americans with Disabilities Act Accessibility Guidelines. Washington, D.C.

Bath, P.A., & Morgan, K. (1999). Differential risk factor profiles for indoor and outdoor falls in older people living at home in Nottingham, UK. *Eur J Epidemiol.*, **15**, 65–73.

Bergland, A., Jarnlo, G.B., & Laake, K. (2003). Predictors of falls in the elderly by location. *Aging Clin Exp Res.*,15, 43–50.

Bergland, A., Pettersen, A.M., & Laake, K. (1998). Falls reported among elderly Norwegians living at home. *Physiother Res Int.,* 3, 164–74.

Blank, S. & Dorf, B. (2012). *The startup owners manual: The step-by-step guide for building a great company.* Pescadero, CA: K and S Ranch Inc.

Bloomekatz, A. (2012). Suits could force L.A. to spend huge sums on sidewalk repair. *L.A. Times*.

Boninger, M.L., Cooper, R.A., Fitzgerald, S.G., Cooper, R., Dicianno, B., & Liu, B. (2003). Investigating neck pain in wheelchair users. *Am J Phys Med Rehabil*, **82**(3), 197-202.

Brault, M. (2012). Americans with disabilities: 2010 (Current Population Reports, P70-131). *Washington, DC: US Census Bureau*.

Census, U.S. (2012). The Older Population in the United States: 2012. Available from: http://www.census.gov/population/age/data/2012.html.

Centers for Disease Control and Prevention, National Center for Injury Prevention and Control. Web–based Injury Statistics Query and Reporting System (WISQARS) [online]. Accessed August 15, 2013.

Control, C.f.D. *Falls among older adults: an Overview*. 2013; Available from: http://www.cdc.gov/homeandrecreationalsafety/Falls/adultfalls.html

Cooper, R.A., & MacLeish, M. (1992). Racing wheelchair roll stability while turning : a simple model. *J Rehabil Res Dev*, **29**(2), 23-30.

Cooper, R.A., Thorman, T., Cooper, R., Dvorznak, M.J., Fitzgerald, S.G., Ammer, W., … Boninger, M.L. (2002). Driving Characteristics of Electric Powered Wheelchair Users: How Far, Fast, and Often Do People Drive? *Arch Phys Med Rehabil*, **83**(2), 250-5.

DiGiovine, C.P., Cooper, R.A., Fitzgerald, S.G., Boninger, M.L., Wolf, E.J., & Guo, S.. (2003). Whole-body vibration during manual wheelchair propulsion with selected seat cushions and back supports. *IEEE Trans Neural Syst Rehabil Eng,* **11**(3), 311-22.

Duvall, J. "Development of surface roughness standard for wheelchair pathways." Master's Thesis, University of Pittsburgh, 13 June 2013.

Duvall, J., Cooper, R., Sinagra, E., Stuckey, D., Brown, J., & Pearlman, J. (2013). Development of Surface Roughness Standards for Pathways Used by Wheelchairs. *Transportation Research Record*, **2387,** 149-56.

Duvall, J., Sinagra, E., Stuckey, D., Pearlman, J., & Cooper, R. (2014). Proposed Surface Roughness Standard for Pathways used by Wheelchairs. Transportation Research Board 93nd Annual Meeting, (14-4494).

Faul, M, Xu, L., Wald, M.M., & Coronado, V.G. (2010). Traumatic brain injury in the United States: emergency department visits, hospitalizations, and deaths, 2002-2006. U.S. Department of Health and Human Services, Centers for Disease Control and Prevention, National Center for Injury Prevention and Control.

FHWA. (2014). Designing Sidewalks and Trails for Access - Part II of II: Best Practices Design Guide. Retrieved from http://www.fhwa.dot.gov/environment/bicycle_pedestrian/publications/sidewalk2/sidewalks204.cfm.

Gaal, R.P., Rebholtz, N., Hotchkiss, R.D., & Pfaelzer, P.F. (1997). Wheelchair rider injuries: Causes and consequences for wheelchair design and selection. *Journal of Rehabilitation Research and Development,* **34** (1), 58-71.

Garcia-Mendez, Y, Pearlman, J.L., Cooper, R.A., & Boninger, M.L. (2012). Dynamic stiffness and transmissibility of commercially available wheelchair cushions using a laboratory test method. *Journal of Rehabilitation Research & Development*, **49**(1), 7-22.

Greenwald, T. (2013, December 4). *Calling out America's terrible sidewalks with a Magic Cart.* Retrieved from http://www.wired.com/autopia/2013/12/magiccart/.

International, A. (2008). *Standard Practice for Computing International Roughness Index of Roads from Longitudinal Profile Measurements*.

ISO, I.S.O. (1997). Mechanical Vibration and Shock-- Evaluation of human exposure to whole-body vibrations (2631-1). *International Standards Organization*.

Justice, Department of. (2010). 2010 ADA Standards for Accessible Design. Retrieved from http://www.ada.gov/regs2010/2010ADAStandards/2010ADAstandards.htm#c4.

Khambatta, A. & Loewenherz, F. (2011). Pedestrian infrastructure on the public right-of-way: Aging pedestrians and prioritizing sidewalk hazards using the Ultra Light Inertial Profiler (ULIP). Transportation Research Board 2012 Annual Meeting.

Kirby, R.L. & Ackroyd-Stolarz, S.A. (1995). Wheelchair safety--adverse reports to the United States Food and Drug Administration. *Am J Phys Med Rehabil*, **74**(4), 308-12.

Latif, A.B.A. (2009). *Relationship bewteen international roughness index (IRI) and present serviceability index (PSI)*. Universiti Teknologi Malaysia.

Li, W., Keegan, T.H.M., Sternfeld, B., Sidney, S., Quesenberry Jr., C.P., & Kelsey, J.L. (2006). Outdoor Falls Among Middle-Aged and Older Adults: A Neglected Public Health Problem. *American Journal of Public Health*., **96** (7), 1192-200.

Loewenherz, F. *Asset Management for ADA Compliance Using Advanced Technologies*. Available from: http://www.ite.org/Membersonly/annualmeeting/2010/AB10H4001.pdf.

Materials, A.S.f.T.a. (1996). Standard practice for safe walking surfaces, in F1637-95.

Osterwalder, A. & Pigneur, Y. (2010). *Business model generation*. Hoboken, NJ: John Wiley & Sons, Inc.

Sayers, M.W. & Karamihas, S.M. (1998). *The little book of profiling*. Ann Arbor: Transportation Research Institute, University of Michigan. http://www.umtri.umich.edu/erd/roughness/lit_book.pdf.

Seidel, H. & Heide, R. (1986). Long-term effects of whole-body vibration: a critical survey of the literature. *Int Arch Occup Environ Health*, **58**(1), 1-26.

Starodub, Inc. (2009). Retrieved from http://www.starodub.com/ulip.html.

Sterling, D.A., O'Connor, J.A., & Bonadies, J. (2001). Geriatric falls: injury severity is high and disproportionate to mechanism. *Journal of Trauma–Injury, Infection and Critical Care*, **50**(1), 116–9.

Stevens, J.A., Corso, P.S., Finkelstein, E.A., & Miller, T.R.. (2006). The costs of fatal and nonfatal falls among older adults. *Injury Prevention*, **12**, 290–5.

Stout, G. (1979). Some aspects of high performance indoor/outdoor wheelchairs. B*ull Prosthet Res*, **16**(2), 135-75.

SurPRO. (2011). Retrieved from http://www.surpro.com/index.php/product/product-specifications.

Tolerico, M.L., Ding, D., Cooper, R.A., Spaeth, D.M., Fitgerald, S.G., Cooper, R., … Boninger, M.L. (2007). Assessing mobility characteristics and activity levels of manual wheelchair users. *J Rehabil Res Dev*, **44**(4), 561-72.

Tromp, A.M., Pluijm, S.M.F., Smit, J.H., Deeg, D.J.H., Bouter, L.M., & Lips, P. (2001). Fall-risk screening test: a prospective study on predictors for falls in community-dwelling elderly. *J Clin Epidemiol*, **54**(8), 837–44.

Washington, University of. (2005). *Pavement Evaluation*. Available from: http://training.ce.washington.edu/PGI.

Weinberg, L.E., & Strain L.A. (1995). Community-dwelling older adults' attributions about falls. *Arch Phys Med Rehabil.*, **76**, 955–60.

Wolf, E., Cooper, R.A., Pearlman, J., Fitzgerald, S.G., & Kelleher, A. (2007). Longitudinal assessment of vibrations during manual and power wheelchair driving over select sidewalk surfaces. *J Rehabil Res Dev*, **44**(4), 573-80.

Works, C.o.S.P. (2014). Sidewalks, curbs, and gutters.