The University Of Sheffield.

Closing the Performance Gap in Building Energy Modelling through Digital Survey methods and Automated Reconstruction

By: Tom Garwood

A thesis submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

The University of Sheffield

Faculty of Engineering

Department of Mechanical Engineering

June 2019

Acknowledgements

This research would not have been possible without the financial contributions of EPSRC and the industrial sponsor BMW. I would like to thank the following departments from the University of Sheffield for their technical support as well as access to facilities and equipment; Department of Mechanical Engineering, the Advanced Manufacturing Research Centre and Factory 2050. Special mention must also be made to IES who provided access to the IES VE software that was critical in conducting this research.

I would like to thank my primary supervisor Professor Ben Hughes for his advice and guidance throughout the process to ensure that this PhD research achieved its potential. I wish him all the success in his new post at the University of Strathclyde. I'd also like to thank Professor Bill Nimmo for his support in the final stages of this research.

A large host of characters deserve a mention for their support during this research from reviewing research paper submissions, co-authoring papers, reviewing research progress, assisting in the laboratories and providing assistance in the correct use of survey equipment; Dr Dominic O'Connor, Dr John K Calautit, Dr Michael R Oates, Dr Ruby Hughes, Tom Hodgson and Dmitri Govorukhin.

Sebastian Ochmann, from the University of Bonn, must be thanked for the provision of source code produced through his own research that enabled me to conduct my own research in Building Energy Modelling and Simulation applications.

Thank you to my fellow PhD researchers Saeed, Victoria and Sheen for your help and support throughout the research. I wish you all the best in your own programmes.

I'd like to thank my family, especially my Dad, for their endless support throughout my PhD research as well as always taking an interest in the subject matter.

Finally, the biggest thank you goes to my little home team unit Dr Stephanie Llewelyn and Zoe. Taking the decision to leave my comfortable career and pursue a PhD could have been very difficult but the two of you made it so much easier than I thought possible. Whether it was talking your ears off or throwing sticks and chasing pigeons around the park you kept me on track. Thank you for listening and your unwavering support throughout, I couldn't have done it without you.

Research Outputs

The following research outputs were produced during the course of this PhD programme:

Journal Publications (Published)

1. Garwood TL, Hughes BR, Oates MR, O'Connor D, Hughes R. A review of energy simulation tools for the manufacturing sector. Renewable and Sustainable Energy Reviews 2018;81:895–911.
2. Garwood TL, Hughes BR, O'Connor D, Calautit JK, Oates MR, Hodgson T. A framework for producing gbXML building geometry from Point Clouds for accurate and efficient Building Energy Modelling. Applied Energy 2018;224:527–37.

Conference Proceedings

1. Garwood TL, Hughes BR, O'Connor D, Calautit JK, Oates MR, Hodgson T. Geometry Extraction for High Resolution Building Energy Modelling Applications from Point Cloud Data: A Case Study of a Factory Facility. In: Yan J, Wu J, Li H, editors. Energy Procedia, Cardiff: Elsevier Ltd.; 2017, p. 1805–10.
2. Garwood TL, Hughes BR, O'Connor D, Oates MR. Automatic Building Geometry Creation From Point Clouds For Simulating Industrial Building Energy Use. Proc. BSO 2018 4th Building Simulation and Optimisation Conference. Cambridge, UK 11-12 Sept. 2018, Cambridge, UK: 2018, p. 396–401.
3. Garwood TL, Hughes BR. Proposed Framework for Improving the Workflow of Retrofit Building Energy Modelling and Simulation (In Press). Proceddings of the International Conference on Construction Futures, Wolverhampton: 2018.

Abstract

Against the backdrop of increasing global efforts to mitigate the effects of climate change there has been a large focus on the Built Environment. The low level of building stock turnover in the UK, estimated between 1-3% per annum, has reinforced the importance of robust retrofit programmes to meet legislated targets [1,2]; Experts predict that the majority of existing UK buildings will still be in use in 2050 [3]. Residential and commercial buildings account for approximately 20% of energy end use globally with UK industry building services such as space heating and lighting account for between 6-56% of overall building energy use, depending on sector [4].

Building Energy Modelling and Simulation (BEMS) software is used to assess the energy performance of a building based on knowledge of its construction, design, use and location. While design data is readily available for new buildings, existing buildings, that are in need of retrofit, tend to have limited as-built building data. This requires a collection of data through site surveys and manual creation of building models; This is a time consuming and expensive activity. The aim of this research was "*Develop a scientific method to remove barriers to urban scale Building Energy Modelling and Simulation (BEMS) using pattern recognition software to extract built forms from large data sets*".

This research has developed a process of rapid geometry generation for BEMS applications to substantially improve this workflow. Following an internal site survey, a Point Cloud was produced of a case-study building. This was automatically processed to create recognisable building geometry for BEMS applications that achieved time savings of 85% over traditional methods. It was identified that internal survey methods present limitations to the automated reconstruction process and that existing offerings for UAV mounted survey equipment required high capital expenditure. A low-cost prototype for external scanning underwent initial development and identified areas for further development.

The geometry that was reconstructed via internal survey data was simulated in BEMS and compared against measured energy data. The annual energy use was predicted to within 6% of the measured energy data. Limitations to a full reconstruction led to a hybrid approach being conducted. The hybrid approach predicted annual energy use to within 4% of measured data and met industrial validation requirements. The research conducted has demonstrated that improvements to the BEMS workflow can be achieved and in doing so it can contribute to the reduction in emissions from the Built Environment.

Acronyms

AEC	Architecture, Engineering and Construction
API	Application Programming Interface
BIM	Building Information Modelling
BEMS	Building Energy Modelling and Simulation
BLOB	Binary Large OBject
CAD	Computer Aided Design
CityBES	City Building Energy Saver
COP	Coefficient Of Performance
CVRMSE	Coefficient of Variation of the Root Mean Square Error
DES	Discrete Event Simulation
DOE	Department of Energy (U.S)
DSY	Design Summer Year
DURAARK	Durable Architectural Knowledge
EC	European Commission
EEPROM	Electrically Erasable Programmable Read-Only Memory
EMF	Electro Magnetic Force
EPBD	Energy Performance of Buildings Directive
EWY	Example Weather Year
F2050	Factory 2050
gbXML	green building Extensible Mark-up Language
GND	Ground Connection
GPR	Ground Penetrating Radar
GPS	Global Positioning System
GUI	Graphical User Interface
HVAC	Heating, Ventilation and Cooling
I2C	Inter-Integrated Circuit
IBPT	International Building Physics Toolbox
IDE	Integrated Development Environment
IES	Integrated Environmental Solutions
IFC	Industry Foundation Classes
IMAS	Indoor Multi-sensor Acquisition System

IMU	Inertial Measurement Unit
ISR	Interrupt Service Routine
KDE	Kernel Density Estimation
LBNL	Lawrence Berkeley National Laboratory
LED	Light Emitting Diode
LEED	Leadership in Energy and Environmental Design
LiDAR	Light Detection And Ranging
LL3HP	LiDAR-Lite v3 High Performance
NMBE	Normalized Mean Bias Error
OOP	Object Orientated Programming
OOPM	OOP Modelling
PCB	Printed Circuit Board
POE	Post-Occupancy Evaluation
PWM	Pulse Width Modulation
QA	Quality Assurance
RANSAC	Random Sample Consensus
RGB	Red, Green and Blue
RoI	Return on Investment
RMS	Root Mean Square
RSS	Root Sum of Squares
SCA	Serial Clock Line
SDA	Serial Data Line
SDK	Software Development Kit
SLAM	Simultaneous Localisation and Mapping
SRAM	Static Random Access Memory
TRY	Test Reference Year
UAV	Unmanned Aerial Vehicle
USB	Universal Serial Bus
VE	Virtual Environment
XML	eXtensible Mark-up Language
XSLT	eXtensible Style sheet Language Transformation

Nomenclature

Greek

Δ	Difference
β	Gradient decent constant
ε	Emissivity of a surface
θ	Angle (°)
λ	Eigenvalue
ρ	Density ($kg\ m^{-3}$)
σ	Stefan-Boltzmann Constant ($5.67\ \times 10^{-8}\ W\ m^{-2}\ K^{-4}$)
$\sum$	Summation
Ψ	Linear Thermal Transmittance ($W\ m^{-1}\ K^{-1}$)
ω	Angular rate of rotation ($°\ s^{-1}$)

Roman

$°C$	Celsius ($273.15K$)
a	Linear acceleration ($m\ s^{-2}$)
$\boldsymbol{a}$	Acceleration vector
ach	air changes per hour (h^{-1})
a_t	Terrain coefficient
A	Area (m^2) / Constant
$\mathcal{A}$	Ampere
$\boldsymbol{b}$	Vector
B	Constant
$\mathcal{B}$	Earth's Magnetic Field
c_p	Specific heat capacity ($J\ kg^{-1}\ K^{-1}$)
c_{wp}	Wind Pressure Coefficient
C	Covariate matrix / Constant
C_f	Flow Coefficient of Air Infiltration Path
CDD	Cooling Degree Days ($24h$)
d	differential
D	Constant
err	Error term

E	Constant
f	Objective function
F	Farad ($s^4\ \mathcal{A}^2\ m^{-2}\ kg^{-1}$)
g	Acceleration due to gravity ($9.81\ m\ s^{-1}$)
$\boldsymbol{g}$	Gravitational vector
G	Gauss ($10^{-4}\ kg\ \mathcal{A}^{-1}\ s^{-2}$)
h	hour ($3600\ s$)
$\boldsymbol{h}$	Quaternion of magnetic data
$\hat{h}$	Convection heat transfer coefficient ($W\ m^{-2}\ K^{-1}$)
Hz	Hertz (s^{-1})
HDD	Heating Degree Days ($24h$)
$\boldsymbol{i}$	First imaginary component of a quaternion
$\boldsymbol{j}$	Second imaginary component of a quaternion
J	Joule ($kg\ m^2\ s^2$)
$\boldsymbol{J}$	Jacobian Matrix
k	Thermal Conductivity ($W\ m^{-1}\ K^{-1}$)
$\boldsymbol{k}$	Third imaginary component of a quaternion
k_{nn}	Number of nearest neighbours selected for normal estimation
k_p	Proportional gain constant
k_t	Terrain coefficient
kg	kilogram
K	Kelvin
KB	Kilobytes
l	Length (m)
m	metre
$\boldsymbol{m}$	Magnetic vector
$\dot{m}$	Mass flow rate ($kg\ s^{-1}$)
mm	millimetre
n	Number of data points or periods in the baseline period
n_f	Flow exponent
$\vec{n}$	Normal direction vector
N	Number of RANSAC iterations

$okta$	Extent of cloud cover ($0okta$ is clear sky, $8okta$ is full cloud cover)
O	Origin point in three dimensional space
p	Number of parameters or terms in the baseline model, as developed by a mathematical analysis of the baseline data
p_{pc}	Point in a Point Cloud
$\overline{p_{pc}}$	3D centroid of the nearest k_{nn} neighbour set
P	Pressure (Pa)
P_s	Stack Pressure (Pa)
P_w	Wind Pressure (Pa)
Pa	Pascal $(J\ m^{-3})$
q	Air Permeability $(m^3\ h^{-1}\ m^{-2})$
$\boldsymbol{q}$	Quaternion
Q	Heat Flux $(W\ m^{-2})$
$\dot{Q}$	Rate of heat transfer (W)
$\vec{r}$	Ray vector
$\overrightarrow{r_O}$	Position vector from an origin point O
R	Thermal Resistance $(W^{-1}\ m^2\ K)$
s	second
t	Time (s)
T	Temperature (K)
$\hat{T}$	Transpose
$\mathcal{T}$	Tesla $(kg\ s^{-2}\ \mathcal{A}^{-1})$
TLC	Total Loss Coefficient $(W\ K^{-1})$
$U\text{-}value$	Thermal Transmittance $(W\ m^{-2}\ K^{-1})$
v	Linear velocity $(m\ s^{-1})$
$\boldsymbol{v}$	Quaternion vector $(\boldsymbol{q_1 i + q_2 j + q_3 k})$
$\vec{v}$	Eigenvector
$\overrightarrow{vr}$	Ray direction vector
v_p	3D Scanner Viewpoint
V	Volume (m^3)
$\dot{V}$	Volumetric flow rate $(m^3\ s^{-1})$
$\mathcal{V}$	Volt $(kg\ m^2\ s^{-3}\ \mathcal{A}^{-1})$

W	Watt ($J\ s^{-1}$)
x	Linear coordinate (m)
$\vec{x}$	Linear vector in x direction
y	Linear coordinate (m)
$\vec{y}$	Linear vector in y direction
$\dot{y}$	Dependent variable of some function of the independent variable
$\bar{y}$	Arithmetic mean of the sample of n observations
$\hat{y}$	Regression model's predicted value of y
z	Linear coordinate (m)
$\vec{z}$	Linear vector in z direction

Subscripts

0	Properties at $0°C$
50	Value of parameter at $50Pa$
avg	Average
c	Cross-section properties
$cond$	Conduction
$conv$	Convection
$emit$	Radiation emission
ext	External properties of a volume
f	Fluid properties
fab	Related to building fabric
G	Global property
i	The index of a value from a set of values
in	Input
int	Internal properties of a volume
j	The index of a value from a set of values
L	Local property
$\dot{m}$	Mass flow rate properties
max	Maximum
mod	Modified value
O	Origin coordinate

rad	Radiation
raw	Raw data readings
s	Surface properties
$source$	Properties from a source
$surr$	Properties of surrounding surfaces
t	Properties at time t
$therm$	Related to the thermal envelope of a building
$vent$	Related to ventilation
x	Component in x direction
y	Component in y direction
z	Component in z direction

Table of Contents

List of Figures

List of Tables

List of Equations

Chapter 1.

Introduction

1.1. Background

The European Commission (EC) roadmap highlighted that the Built Environment sector is capable of reducing its emissions by 90% in 2050 compared with 1990 levels [5]. The roadmap stated this reduction can be achieved through the implementation of passive technology, retrofitting existing buildings and switching from the use of fossil fuels to sustainable alternatives. The UK's Climate Change Act specified a legally binding target of 80% reduction against 1990 levels in the same time frame across all sectors [6].

The low level of building stock turnover in the UK, estimated between 1-3% per annum, has reinforced the importance of robust retrofit programmes to meet legislated targets [1,2]; Experts predict that the majority of existing UK buildings will still be in use in 2050 [3].

Residential and commercial buildings account for approximately 20% of energy end use globally with industrial energy end use accounting for 54% [7]. In the UK these proportions have been reported as 30% and 17% respectively [4]. UK industry building services such as space heating and lighting account for between 6-56% of overall building energy use, depending on sector [4].

To maximise the efficacy of any retrofit programme it was identified that it is critical for practitioners to be provided with tools and technologies that enable efficient workflows with reliable outputs. Digital tools and technologies available to practitioners within the Built Environment sector rapidly increased in number and sophistication since the advent of Building Information Modelling (BIM) in 1974 [8].

In modern terms BIM has been considered as a central project store of data relating to the many facets of a construction project. BIM has been used to store Computer Aided Design (CAD) models, construction plans, project costs, bill of materials and results from a plethora of different engineering simulations such as structural, daylighting and energy use assessments. It is increasingly common to find that these engineering simulations are conducted by 3rd party software but interoperability between BIM and these 3rd party software packages has been highlighted as a barrier to seamless data transfer and fully integrated digital documentation of construction projects [9,10].

Building Energy Modelling and Simulation (BEMS) has been used to conduct some of these 3rd party engineering simulations and it has performed an important role in building retrofit projects. BEMS has been used to quantify expected energy savings and Return on Investment (RoI) for different retrofit options at an early stage of projects. BEMS has been employed to predict the amount of energy used by a building over a period of time by considering the geometry, use, construction, location and orientation of a building. This has been conducted at different stages of the building's lifecycle from design through to building operation. This has helped to inform energy use orientated decisions from building design or retrofit options.

In recent years the commercial market has grown to contain several options available to BEMS practitioners such as Design Builder [11], EnergyPlus [12], eQuest [13], Green Building Studio [14], International Building Physics Toolbox (IBPT) [15] and Integrated Environmental Solutions (IES) Virtual Environment (VE) [16]. While differences have been found to exist between the software packages the underlying calculations used are all an assessment of the thermal balance in the building envelope.

Within the library of digital engineering toolsets that have been available, BEMS has been a specific analysis tool and, in the majority of cases, doesn't handle other data that may be needed across the spectrum of engineering applications within a full construction project. BIM fulfils this function and is used to capture, centrally, all information related to the design or construction of a building. BIM has served many uses across the Architecture, Engineering and Construction (AEC) industry and has been used to store large levels of information related to the wide possible application of the data. A large proportion of this data was found to not be relevant for BEMS, see Figure 1.1.

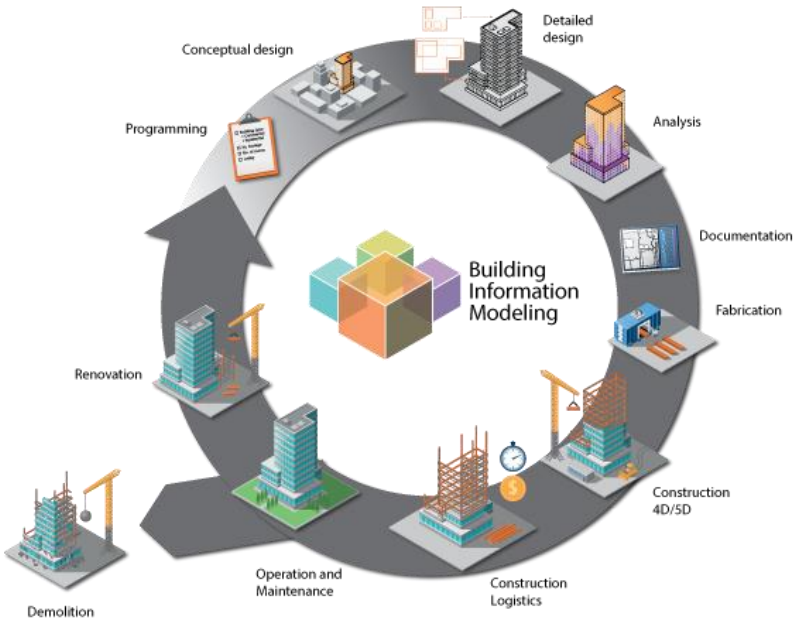


Figure 1.1. BIM lifecycle [17]

Where possible, there has been a clear benefit of being able to re-use existing BIM data for the purposes of BEMS via an automated data transfer method; especially for building geometry, for example. In early design stages, BEMS has been used to conduct energy analysis for differing design concepts. In some limited circumstances this has allowed energy modelling to be used as a decision tool for selecting the best design instead of a single regulatory check on a preselected design to ensure energy and efficiency requirements are met. It has been established that during the operation stage of a building's lifecycle the use of as-built BIM data could have been utilised in BEMS to make the most appropriate retrofit decisions for reducing the building energy use and improving energy efficiency.

However, poor interoperability between BIM and BEMS often leads practitioners to remodel the entire geometry of a building manually. Considering older buildings, where BIM has not been created, or has not been kept up to date, lengthy and disruptive site surveys are performed to establish the as-built building geometry for creation in BEMS software. This has led to poor value for money in conducting BEMS for a building operator/owner who either may have already invested in BIM or didn't want disruptive surveys taking place. It also means that BEMS model creation has formed a disproportionately large fraction of the entire retrofit BEMS workflow compared with high value engineering and energy analysis.

1.2. As-Built Building Modelling

In recent years the technology available to building survey practitioners has advanced substantially. Some of the recent developments included digital survey techniques such as LiDAR, Photogrammetry, Thermal Imaging and Radar. These surveys generate large amounts of raw data however the native support for automatic processing of data streams has been limited. It is common for manual operations to be conducted using survey data as a guide such as tracing CAD geometry over survey data.

1.2.1. LiDAR

Light Detection and Ranging (LiDAR) sensors, more commonly known are laser scanners, are used to detect the distance from the sensor to surfaces surrounding the sensor. When the LiDAR units rotate the sensor during distance measurements, Point Clouds are created that provide a visual representation of the environment surrounding a LiDAR sensor. The available commercial sensors require high capital expenditure to purchase. This has pushed the accessibility of the technology out of reach of a majority of existing and potential survey practitioners for both static terrestrial scanners and scanners designed for use on an Unmanned Aerial Vehicle (UAV).

Further effort has been made in recent years to develop LiDAR systems for UAVs however the cost of the commercially available systems has limited the technology accessibility for LiDAR surveys [18]. Indicative costs of several commercially available UAV LiDAR systems have been tabulated in Table 1.1.

Table 1.1. Example costs of commercial UAV compatible LiDAR systems (correct in 2018)

Company	System	Cost
Clickmox Solutions Inc.	VERSA3D	US$58,000.00
3Dlasermapping	ROBIN-MINI	€140,000.00 ~US$161,340.00
Velodyne	VLP-16	US$7,900.00
Velodyne	VLP-16 Lite	US$9,500.00
Velodyne	VLP-16 HI-Res	US$12,000.00
Velodyne	VLP-32	US$49,950.00
Hokuyo	UTM-30LX	US$4,500.00

1.2.2. Photogrammetry

The Photogrammetry technique is used to capture multiple overlapping photos of an object from different positions or orientations. The captured images are then subsequently processed by image recognition algorithms to identify common features in any photo overlap regions. Trigonometry is then used with these identifying markers to map the object being photographed into a 3D environment. The produced 3D maps are found to be more detailed and higher quality when a higher image resolution and larger overlap area among a collection of photos is used during data capture.

Similar to LiDAR, site surveys are conducted with scan targets, see Figure 4.1, to aid post-processing but several commercial photogrammetry software packages have been observed to achieve a high success rate without the need for targets. Photogrammetry is used to solve a computer vision problem in reconstruction but as distances are not recorded in the acquired data set any reconstruction, including the dimension between any two points, are determined indirectly. This is contrasted against LiDAR which is seen as a direct measurement method.

Three beneficial features were identified for Photogrammetry over LiDAR;

1. Hardware costs were identified as much lower which has led to the higher use of photogrammetry in UAV applications compared with LiDAR. For example, the cost of high quality cameras supplied by a world leading UAV commercial company has been found to range from £1,969-£2,699 depending on the camera specifications [19].
2. Photogrammetry is known to support a wider field of view when compared to single beam LiDAR. Photogrammetry has been able to capture details of the side elevations of objects during a flyover due to the wider field of view, see Figure 1.2.

Figure 1.2. (left) 2D Google Maps image, (right) 3D Google Maps image

3. The level of noise captured during a Photogrammetry survey is reduced. For example, any reflections captured in one image will not look the same when captured in a second image from a different position. This has allowed common features to be retained and noise to be ignored easily by detection algorithms.

1.2.3. Thermal Imaging

In construction surveys infrared images are taken of construction elements (e.g. a building façade) and used to determine the relative conductivity across the element under inspection. The readings captured from the infrared sensors are subsequently depicted in the visible light spectrum to enable interpretation by human practitioners, see Figure 1.3.

Figure 1.3. Example thermographic image of building façade [20]

Thermal images are used to illustrate the contrast in energy emitted by different areas of a building element. Large contrasts identified in a thermal image allow the detection of thermal leakage from a building envelope, in particular, through any thermal bridges (e.g. steel girders) that transport heat from inside to outside with little resistance. Thermal imaging is used for spot assessments to identify weaknesses in a building's envelope that enable the determination of which areas would benefit from retrofit improvements.

Unfortunately, thermal imaging data is based upon the site conditions at the time of the survey and therefore environmental conditions (e.g. air temperature and humidity) can

significantly affect the identification of thermal weak spots in a thermal envelope. This has led to building assessments based upon a limited snapshot in time which may have differed if the survey was conducted over a long time period. Photogrammetry techniques have also been applied to thermal imaging in order to render a 3D visualisation of a building and the thermal performance of its exterior, see Figure 1.4.

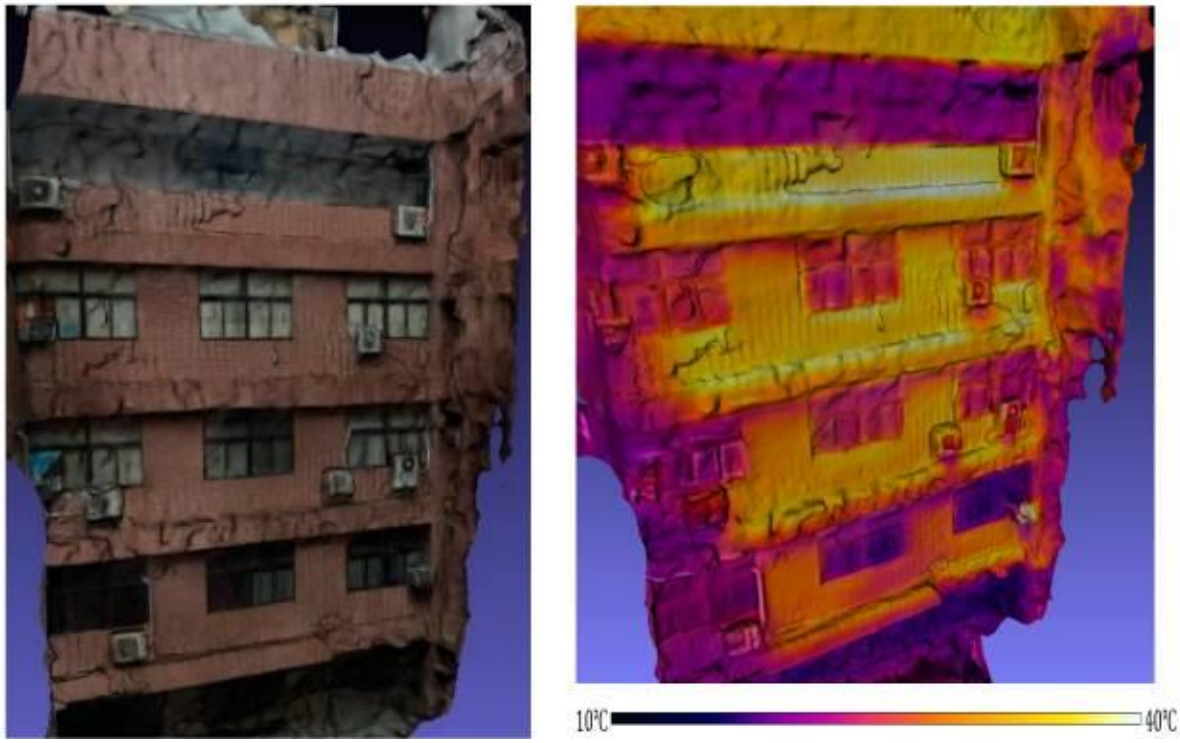


Figure 1.4. 3D model and thermal 3D model of building exterior wall [21]

1.2.4. Radar

In a similar fashion to LiDAR that relies on emitting, reflecting and receiving a laser beam signal, Radar is used to emit, reflect and receive radio waves. Radar has been utilised across many applications which have included distance measuring and object tracking.

While original Radar systems required large antenna and were primarily used for military purposes recent developments have produced systems for the detection of subterranean utilities and structures. Unlike LiDAR, Radar has been used to penetrate through opaque objects during a survey. This has been especially useful during construction, retrofit or site assessment works as the location and depth of critical objects (e.g. electric cables or water pipes) have been identified prior to any drilling or cutting to avoid unwanted damage.

A range of Ground Penetrating Radars (GPR) have been developed for the commercial market that primarily focus on ground sub-surface inspections. The same technology has also been applied to wall sub-surface inspections however wall applications tend to be more focused on identifying rebar in concrete with a much lower depth resolution. Recent research has investigated the use of GPR and image recognition techniques to identify and reconstruct water pipes using GPR data [22,23]. Figure 1.5 illustrates an example of raw GPR data outputs.

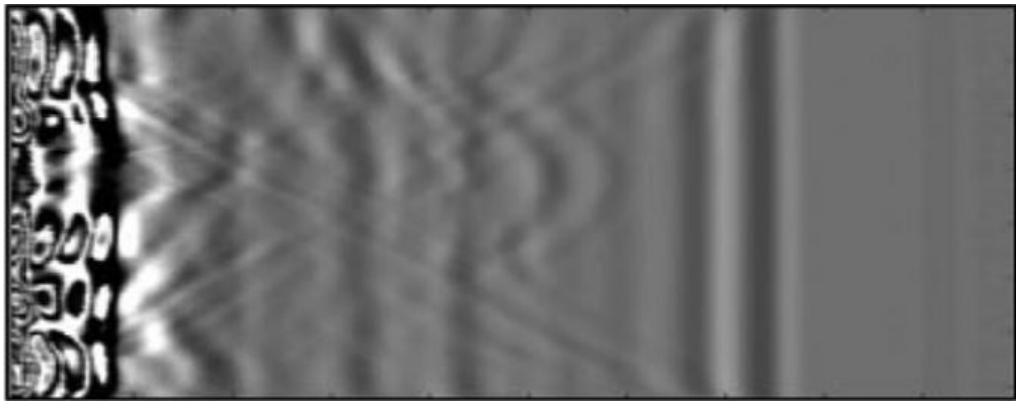


Figure 1.5. Raw GPR image [22]

Of these four technologies; LiDAR, Photogrammetry, Thermal Imaging and Radar, the data provided by thermal imaging and radar is used to provide useful information but workflows for building geometry generation have been found to be immature or unknown. The capabilities of LiDAR and Photogrammetry have captured building geometry quickly with mature workflows. When the two techniques have been compared it has been observed that LiDAR provides a direct measurement instead of the indirect estimation generated through Photogrammetry. However, LiDAR data has been shown to be more susceptible to include noise generated from reflective surfaces and cluttered environments. In addition, a high capital cost has been associated with LiDAR equipment.

As previously described, LiDAR data capture techniques have stored data in a mature and well-developed data format that has been used throughout the survey industry. This format is known as a Point Cloud that represents the distance measurements made from a laser scanner as a large set of discrete coordinates points. These points are usually rendered to provide a visualisation of the environment that surrounded the scanner during a survey.

1.3. Point Cloud Data

Point Clouds were created to store a large dataset consisting of multiple points within a 3D coordinate system; this is usually with a Cartesian coordinate system (i.e. X, Y and Z). This type of dataset is used to virtually represent the surface geometry of objects within the defined coordinate system. Point Clouds have been used in a range of applications to represent a range of objects such as landscape topography, building features (e.g. floors, walls, roofs, windows and doors) and equipment. Point Clouds are acquired when sophisticated LiDAR equipment is deployed that is used to record data points in a 3D volume. The resolution of the scan between adjacent data points is varied depending on the object surveyed but it reaches a resolution of a few millimetres if required. A high-resolution Point Cloud of a building, for example, usually has millions of data points in the data set. Most commercial laser scanners are supplied with software allowing the 3D coordinates to be mapped as illustrated in Figure 1.6.

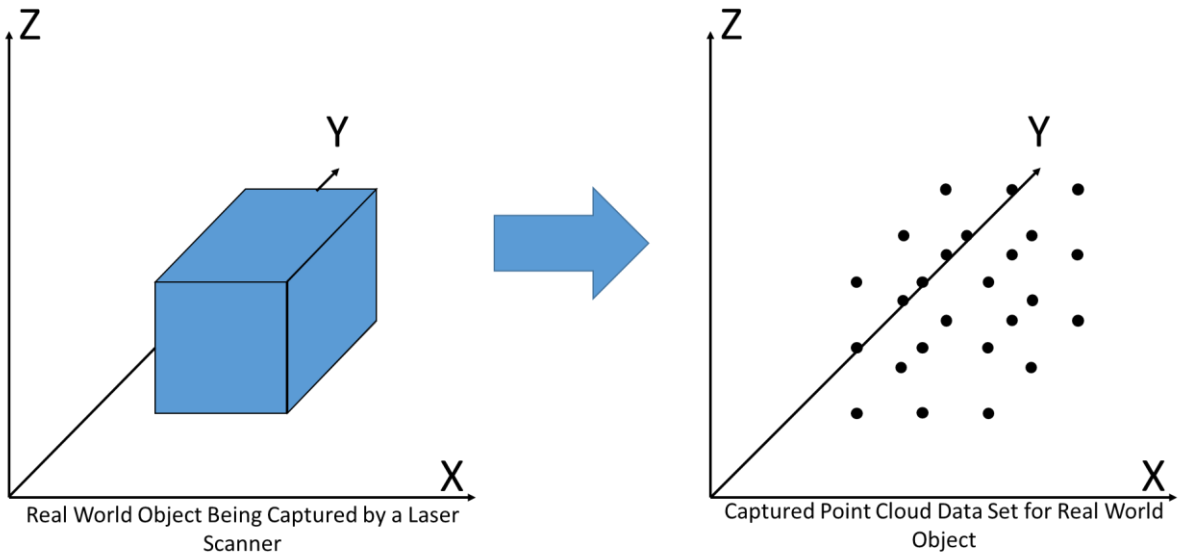


Figure 1.6. Representation of Point Cloud production

Many benefits have been identified in using Point Clouds for mapping real-world objects. For example, inspections of manufacturing or construction tolerances, rapid geometry mapping of large objects or land areas, the use of as-built information to inform future design decisions and further enhancements such as virtual reality plug-ins. It has been observed that simplistic geometry, as illustrated in Figure 1.6, with a low scan resolution is difficult to visualise when presented as a Point Cloud.

Having produced a Point Cloud dataset, very little comprehensive native support for data processing was found. In the majority of cases where Point Clouds are used in the AEC industry, CAD geometry was found to be traced over Point Cloud data. This has been highlighted as a workflow inefficiency when pattern recognition techniques have shown the potential to rapidly detect, for example, that a cluster of points lie in a single plane and represent the same wall surface. Surveys that have been conducted over large topographical areas have captured datasets covering large urban areas that have consisted of multiple buildings. Pattern recognition and reconstruction methods would have allowed for a much greater utility of the survey technology if it was available as a tool to industry practitioners.

While BIM would have been a high value target for applying these techniques the interoperability between BIM and other applications such as BEMS would have persisted. The ability to accurately measure and capture building geometric information for the purposes of BEMS would have been highly beneficial in the retrofit BEMS industry. The use of such a tool would have increased value in the BEMS workflow and informed effective sustainable retrofit decisions while avoiding the interoperability issues associated with BIM. The automated processing of Point Clouds would have offered a method to rapidly generate as-built building geometry in a VE that included all potential renovations that have taken place since first construction but missing from current building plans or models. The use of a pattern recognition method to identify building forms in large digital datasets for retrofit BEMS workflows would have aided in the removal of workflow barriers that have persisted within the industry.

While BEMS has been identified as a tool that can contribute to reduced emissions across the built environment substantial disincentives exist in its application to retrofit projects. The lack of BIM and interoperability with BEMS requires geometry to be manually created for energy assessment purposes which is time consuming and expensive. Methods of digital surveying of buildings have the potential to help in this regard but no simple solution currently exists.

1.4. Aim

The research detailed within this thesis explores two main strands of investigation following a critical review of relevant literature; 1) methods of automatically processing internal digital

survey data of a case study building to improve the efficiency and reliability of the BEMS retrofit workflow, and 2) development of a low-cost and mobile LiDAR unit to facilitate external building surveys.

The overarching aim of this research is thus;

"*Develop a scientific method to remove barriers to urban scale Building Energy Modelling and Simulation (BEMS) using pattern recognition software to extract built forms from large data sets*".

1.5. Objectives

The objectives of this research that need to be fulfilled to achieve the stated aim in subsection 1.4 are as follows;

1. Conduct a critical review of existing research and current state-of-the-art techniques to identify gaps in the existing knowledge base for this research to fill.
2. Use a case study building to establish a benchmark energy model using current industry accepted workflows and validation.
3. Generate and process large digital survey dataset of case study building to automatically reconstruct geometry for BEMS.
4. Compare BEMS results using extracted building geometry against measured energy data from case study building to establish performance gap of technique.
5. Establish computational code to repeatedly carry out the processes on similar large digital survey datasets carried as defined in objective 3.
6. Based on objectives 3, 4 and 5, establish a method of digital survey data collection at an urban scale through the development of accessible hardware.

1.6. Thesis Structure

Chapter 1 introduces the subject matter and provides context for the research detailed within this thesis. The overarching research aim, and corresponding objectives are provided followed by a summary of the content of each thesis Chapter.

Chapter 2 documents the critical literature review that was conducted for this research that identified gaps in existing knowledge that formed the focus of the subsequent research described in the remainder of the thesis.

Chapter 3 explains the theory behind BEMS and details the research undertaken on a case study facility via a traditional BEMS workflow. The results of this benchmark BEMS were validated against the applicable industrial standards to provide confidence in the boundary conditions used.

Chapter 4 describes the experimental software research that took place which included site survey methods, data registration and automated building geometry extraction from survey data for use in BEMS.

Chapter 5 presents the research undertaken in hardware development towards a low-cost LiDAR for external scan applications. The research included identification of prototype requirements and iterative development towards achieving these requirements.

Chapter 6 outlines the results of BEMS that were conducted on reconstructed building geometry using site survey data. The results of validation against industry standards were provided to identify the performance gap against measured data. A comprehensive discussion on the findings of the research contained within earlier Chapters of this thesis is also provided. The discussion includes identification of the research novelty and limitations, as well as recommendations on the direction of future research to overcome these limitations.

Chapter 7 concludes by detailing how the conducted research has achieved all objectives listed within subsection 1.5 of this thesis in pursuit of the overarching aim of the research as specified in subsection 1.4. The closing remarks summarise the areas for future research to build upon the conducted research.

A complete set of references used throughout this research are provided at the end of this thesis. The list is numbered according to order of appearance within the thesis. Where referenced in the main body of the text the corresponding reference number is included in square brackets (e.g. [X]).

A set of appendices has been provided at the end of this thesis that contains complementary information to the research conducted in Chapter 3, Chapter 4 and Chapter 5.

Chapter 2.

Literature Review

2.1. Introduction

The critical literature review that was undertaken to identify the research gap is documented in this Chapter. The latest state-of-the-art research in BEMS with a specific focus on BIM interoperability, the Performance Gap and BEMS Calibration is summarised in subsection 2.2. An overview of state-of-the-art in using Point Cloud data to assess the energy use of buildings is provided in subsection 2.3. This review was split into two categories of Point Cloud use and energy assessment; 1) Urban scale, and 2) Building scale. Existing research into the development of low-cost and mobile survey hardware relevant to Point Cloud data capture has been reviewed in subsection 2.4. A summary of the literature review is provided in subsection 2.5 followed by a concise description of the identified research gap in subsection 2.6.

2.2. Building Energy Modelling and Simulation (BEMS)

BEMS is used to conduct thermal simulations of a building that considers the geometry, construction materials, use and local environmental conditions of the building under assessment. Several reviews on the use of BEMS have been identified in the academic literature as summarised below.

Garwood *et al.* [9] highlighted the challenges of BEMS in manufacturing environments through a review of existing literature on combining the traditional BEMS with Discrete Event Simulation (DES) that is used to simulate manufacturing processes. The review identified that progress has been made in simulating the energy use across different system levels within a manufacturing facility including interdependencies; machines, process lines, Technical Building Services (TBS) and the building shell. However, the progress to-date has generally been simplistic and "proof of concept" in nature resulting in possible solutions towards a holistic energy simulation but requiring further development to obtain a comprehensive simulator. The paper reviewed the developed modelling approaches and the tools available for use in future research. Requirements were identified for the development of a holistic energy simulation tool for use in a manufacturing facility, that is capable of simulating interdependencies between different building layers and systems, as well as a

rapid method of 3D building geometry generation from site data or existing BIM in an appropriate format for energy simulations of existing factory buildings.

Reviews by Volk *et al.* [24], Gao *et al.* [25], Farzaneh *et al.* [26] and De Wilde [27] covered a broad range of subject matter within the BEMS research field. It was identified that the majority of BIM tools currently in use are intended for new design work rather than existing buildings. More comprehensive requirements are needed to improve the data capture process when BIM is created for existing buildings. The use of correct BIM data capture techniques in combination with new technology is not regularly found in practice requiring a more comprehensive framework on BIM creation. Existing BIM to BEMS data exchanges are inefficient and offer poor results. gbXML was identified as useful data storage format in the BEMS domain. Studies investigating the Performance Gap highlight difference between first principle simulations and measured data, differences between machine learning techniques and measured data the difference between energy rating in compliance with display certificates.

2.2.1. BIM Interoperability

BIM was designed as a central digital data store for an entire construction project. The practicalities of ensuring data transfer between BIM and the large number of available analysis software packages, such as BEMS, is a large problem in the industry [9,10]. Several researchers have investigated effective methods of translating BIM data into BEMS software. The work undertaken has shown promising results although there are still areas for development.

Gourlis and Kovacic [28] considered the use of BIM data for direct use in BEMS. BIM metadata is extensive and likely to include useful information for BEMS; conversely there is a lot of information stored within BIM data structures that is not relevant to BEMS (e.g. construction timelines). The authors identified that the transfer of BIM data into BEMS software was not a mature process. This resulted in a large amount of adjustments and manual remodelling in order to make the BIM data useable in the BEMS environment. The paper presented a case study of two factories; a historic metal cutting and forming production facility, and a new construction for the food industry. The paper discussed the challenges in the simplification of Architecture, Engineering and Construction (AEC) models

such as modelling thermal zones using “air walls” to obtain a series of discrete thermal volumes, see Figure 2.1. The authors concluded that the full potential of using BIM data across the AEC industry remain largely unexplored due to data protection and industrial secrecy.

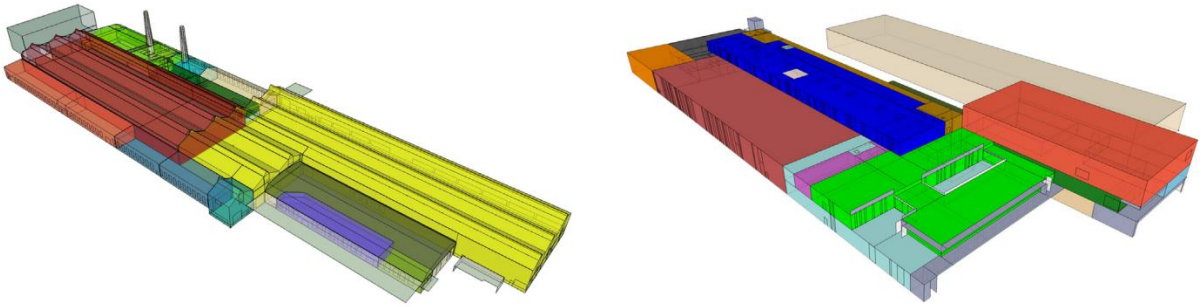


Figure 2.1. Defined thermal zones in two BEMS case studies [28]

Kamel and Memari [29] performed a review of BIM applications of energy simulation which highlighted the interoperability issues that remain within the research area. They conducted three cases studies and the results highlighted that data required for BEMS, which is included in BIM, is often omitted from the data transfer. The authors developed a corrective middleware between BIM and BEMS to correct a gbXML file for use in BEMS. The lack of standards available for data exchange has resulted in interoperability problems of varying degrees and the authors recommended that standards should be developed further.

O’Donnell *et al.* [30] identified that with BEMS it is becoming increasingly common that the results are often questionable and may lead to incorrect design decisions. The authors developed a semi-automated process that enabled reproducible conversions of Archicad [31] BIM data building geometry into a format that is required by the BEMS engine EnergyPlus [12]. The authors speculated that there is the potential to develop the process to provide geometry definitions for BEMS tools in addition to EnergyPlus. The benefits of this development were categorised as reduced BEMS time and cost, rapid generation of design alternatives, improved accuracy of BEMS and considerably improved building performance. It was identified that adequate training in the design of BIM processes for the purposes of BEMS could resolve the majority of the issues encountered to avoid remodelling as part of the BIM2BEMS process. The authors also developed a customised rule set to support validation of models that have undergone a BIM2BEMS process.

Reeves *et al.* [32] developed guidelines for the evaluation and selection of BEMS tools for use in specific building lifecycle phases. The objectives of the research were to evaluate existing BEMS tools, illustrate the application of the top three BEMS tools and develop guidelines for evaluation, selection and application of BEMS tools in the design, construction and operation/maintenance phases of buildings. In total twelve BEMS tools were evaluated using the criteria; Interoperability, Usability, Available inputs, and Available outputs. The top three tools were evaluated in a case study consisting of two separate buildings for energy usage, daylighting performance, and natural ventilation. All three tools were re-evaluated, post case study, against the initial criteria plus the addition of speed and accuracy. IES VE [16] was selected as the most appropriate BEMS software when criteria were weighted evenly. The authors concluded that the developed guidelines can help potential BEMS practitioners to identify the most appropriate BEMS tool for application in specific building lifecycle phases.

Jeong *et al.* [33] acknowledged that the BIM2BEMS process is labour intensive, error prone and cumbersome. Existing tools are only semi-automated requiring input/modification along the process. The authors speculated that problems arise as BEMS tools do not take advantage of Object Orientated Programming (OOP); the foundation of BIM data structures. A mapping tool was created for BIM data, from Autodesk Revit [34], to BEMS data, using Modelica [35] models, that utilise OOP, see Figure 2.2. The BIM2BEMS development enables interdisciplinary data exchange between BIM and BEMS for energy simulations which significantly reduced the effort required for BEMS definition. Validation was conducted against other models, but no real-world validation was recorded.

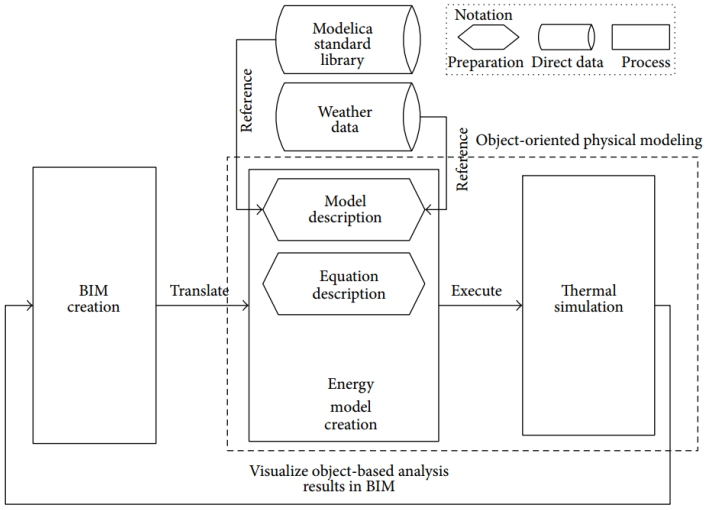


Figure 2.2. Translation process between BIM and ModelicaBEMS. [33]

Kim, Jeong *et al.* [36] presented the development of a Modelica library for BIM-based BEMS using an OOP Modelling (OOPM) approach and Modelica. The aims of the paper were to enhance interoperability among BIM and BEMS tools, enabling more reliable BEMS, enhancing the integration and the coordination among multi-domain BEMS and enabling BIM as a common user interface for the multi-domain energy simulations. The developed workflow translated BIM data into OOPM in Modelica, semi-automatically generated BEMS data from BIM data, immediately performed building analyses, and reported simulation results. The BIM2BEMS disparities such as semantic mismatches of building components and behaviour mismatches between AEC design and thermal modelling were addressed. The approach was validated against the Lawrence Berkeley National Laboratory (LBNL) Buildings Library (not real world); comparable results were obtained. Validation also took place against the requirements of the ANSI/ASHRAE Standard 140-2007 [37]. The authors concluded that BIM data can be translated into Modelica BEMS models using their proposed method. It was identified that coherent model structures between BIM and BEMS allow direct feedback from energy analyses to building design without complex result interpretation.

Jeong *et al.* [38] presented a framework for integrating BIM and OOPM-based BEMS with a focus on thermal simulation to support decision making in the design process. The authors presented the developed prototype "*Revit2Modelica*". The capability of topology creation facilities in Modelica facilitates a natural mapping from object based architectural modelling

(i.e. BIM) to OOPM-based BEMS. The framework was validated using a benchmark model and comparing simulation results for heating and cooling which achieved comparable results with LBNL's results which have been previously validated. The authors concluded that 1) the developed BIM Application Programming Interface (API) allows direct access to BIM data for direct use in BEMS, 2) parametric modelling capability can be preserved in BIM data, and 3) employing BIM and OOPM can enhance the interoperability between architectural models and energy simulation models. Areas for further beneficial work were identified to include testing of diverse topologies and complex building geometry, the use of real-world data to enhance and validate the process, development of a pre-processing program and an algorithm to recognise relationships among building objects (e.g. floor, walls and roof).

Jeong and Kim [39] used case studies to investigate the feasibility of automatically translating a building topology from BIM to OOPM-based BEMS. The authors presented the proposed method and case studies were used to evaluate the process. Each case study increased the complexity of building geometry and objects. Results were compared with the verified OOPM-based models using the LBNL Modelica Buildings Library [35]. The authors concluded that the approach allowed seamless model translations from BIM to OOPM-based BEMS with test cases having validated the automatic model translations which provided time savings and error reductions. Potential future work was identified as the use of building operational data that highlights the advantages automatic BIM to OOPM based BEMS mapping can offer such as reducing time and elimination of error-prone manual processes. The authors noted that the research should be expanded to incorporate BIM authoring tools other than Revit [34]. More complicated and diverse real-world building projects are needed as there is scope to apply this to as-built buildings for lifecycle decision making.

Garcia and Zhu [40] presented an automated data exchange solution through the design of a novel eXtensible Style sheet Language Transformation (XSLT). Their approach included a series of instructions to facilitate the information exchange between BIM and BEMS. The paper focused on information exchange between gbXML and Department of Energy (DOE)-2 schemas and was implemented as an Autodesk Revit [34] plug-in. Validation occurred using the Oxygen eXtensible Mark-up Language (XML) Editor v15.2 on three case studies; a classroom, a multi-story office and a retail store. The conversion process was able to correct most modelling mismatches that occurred during traditional conversion, see Figure 2.3.

Remaining geometry mismatches required manual intervention. The whole process resulted in a time saving of approximately 79%. The proposed XSLT-based converter was able to simulate electric and gas consumption of the case studies, but this was not validated against measured data. Further work areas were highlighted as interoperability issues between the gbXML schema and the open standard schemas adopted in other BEMS software applications.

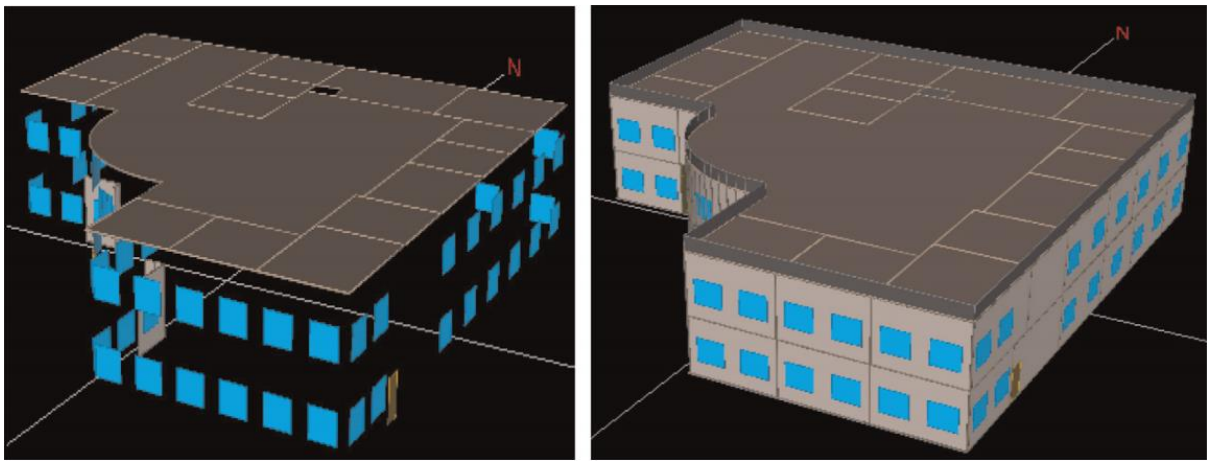


Figure 2.3. Comparison of BIM2BEMS exchange results (left) Green Building Studio, (right) proposed XSLT process [40]

Kim, Shen *et al.* [41] focused on one of the major limitations of BIM2BEMS data exchanges; inaccuracies through simplifications in construction and material data. The authors aimed to improve accuracy of the BIM2BEMS process by developing an object-based approach of materials definition in which the user may change and expand various material properties. The process included establishing automatic data parsing using the IFC format and development of an object-based approach in BIM material properties. The results were validated against a case study using a multi-room and multi-storey office building which were compared with a different BEMS tools achieving a 6.9% difference. The authors claimed improved accuracy however there was no obvious evidence for this claim; only a differing energy use when compared to a different BEMS tool was reported. The proposed approach, however, allowed for more accurate material properties to be used so an improved accuracy would be expected. The efficiency of the BEMS process was improved through time savings and reduced manual inputs.

Bahar *et al.* [10] focused their investigation on thermal simulation tools that provided functionalities to exchange data with other BIM tools. This provided an understanding of

basic working principles and identified thermal tools native to certain BIM software, see Figure 2.4. The authors identified that BEMS tools are traditionally stand-alone tools and are used as independent 3rd party tools separate from BIM tools. Such software applications are required to understand the native formats of the other applications to achieve improved interoperability. Improved interoperability requires information structures that are standardised through industry such as IFC and gbXML. The authors considered gbXML to be better suited for thermal simulations. The authors concluded that 1) interoperability through the BIM platform offers one solution to the problem of software integration, 2) interoperability between BIM and BEMS can improve workflow and analysis, 3) the potential of BIM in terms of built-in energy analysis has not been completely optimised yet, and 4) more detailed analyses rely on stand-alone thermal software that requires its own geometry. The authors determined that further work was needed to determine if the appropriate design information used in thermal analyses can be adequately captured in BIM.

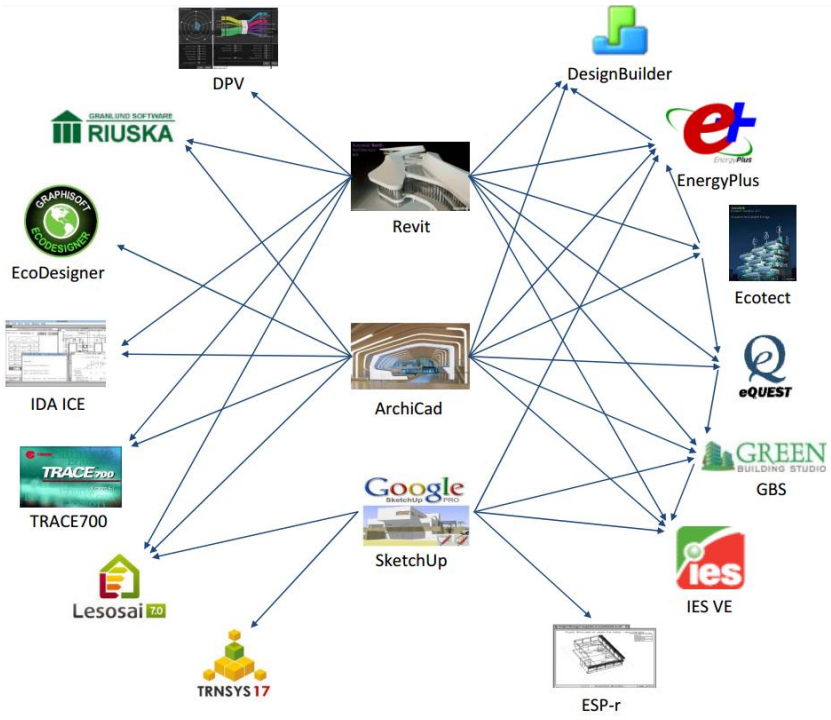


Figure 2.4. Summary of data exchange between BIM and BEMS tools [10]

Abanda *et al.* [42] carried out an appraisal of different BIM software systems for managing construction project information. The authors identified that interoperability and communication between BEMS and BIM is a key issue and the "*skyrocketing*" number of new software, in both fields, means that common management between the two domains is

becoming increasingly difficult. A major challenge that was identified was one of trust between BIM users and vendors as it is only upon purchase that the full functionalities of software become apparent. This places a large risk on users in investing in BIM when interoperability is such a prevalent issue.

Ahn *et al.* [43] identified that no practical solution of interoperability between BIM and BEMS had yet been developed. The authors developed a full and semi-automated method of transferring BIM data, in IFC format, into BEMS. The methods differed by assuming default values in the fully automated version and allowing user inputs for the semi-automated version. The fully automated version created large variation in results when assuming default values while the semi-automated method allowed for experienced judgement of a suitably qualified practitioner on BEMS inputs.

Pezeshki *et al.* [44] reviewed and classified applications of BEMS within the literature and identified where BIM databases have been used to inform BEMS. The authors stated that EnergyPlus was the software of choice within the literature, but it appeared that only a limited number of studies (only 58 references in paper) were considered in their review so it may not be a reliable finding. The number of studies that considered BIM to BEMS workflow was a small proportion of the overall literature. Interoperability was highlighted as being responsible for creating barriers to the development and delivery of projects.

2.2.2. BEMS and the Performance Gap

The “Performance Gap”, with respect to BEMS, has been identified as the observation that across many studies the results from BEMS do not accurately predict the energy use of the buildings under assessment following occupancy upon completion of a construction or retrofit works. There are many variables associated with BEMS and the majority of commercial BEMS software simplify all analyses by assuming constant values for BEMS inputs. This uncertainty in simulation inputs and lack of training for practitioners are key reasons for the Performance Gap as described in this subsection.

Gourlis and Kovacic [45] performed follow on research, from that described in subsection 2.2.1, that investigated the use of BEMS on an existing industrial facility for retrofit purposes; two retrofit scenarios were assessed. The authors also created a benchmark scenario however it was not validated against measured data. The authors identified the

large amount of uncertainty to be dealt when conducting retrofit BEMS with published literature and industrial guidance dictating many of the boundary conditions used in their study. The simulation results identified a potential reduction of 52% in heating demand through retrofit and a reduction in overheating temperature peaks by $6°C$ in some cases. Although the full building system, including production processes, was considered in the assessment no retrofit proposals for reducing process energy was provided.

Katunsky *et al.* [46] also investigated energy use in an industrial setting. The authors stated that there was no standard method and validation technique for BEMS within an industrial environment and set out to develop a practical solution. Heating energy loads were calculated through measurements, calculations and dynamic simulations. An important observation included the uncertainty around internal gains from occupants and machinery which required high levels of definition. The authors were not able to reproduce measured temperatures within the stratified volume of air through BEMS without a detailed knowledge of the HVAC systems being used. Recommendations of positive strategies of energy reduction via BEMS included integration of lighting, heat recovery and door automation.

Sun *et al.* [47] identified that input variables used in BEMS can have a huge variation. They developed a probabilistic risk assessment method to identify the most sensitive or uncertain BEMS inputs. Latin hypercube sampling was then deployed to generate BEMS inputs stochastically using the Monte Carlo method. 10,000 simulations were then run on the seven most sensitive parameters and best curve fitting techniques were applied to the results in comparison with measured energy use and energy cost. This provided additional confidence in the BEMS used to support retrofit assessments but required substantial computing infrastructure to be economical.

Eguaras-Martínez *et al.* [48] focused on the effect of occupancy and internal load schedules in BEMS by comparing results that utilised widely accepted schedule templates with measured schedules. They demonstrated that the annual energy use can vary by as much as 30% when measured occupant schedules are used compared to using industry accepted templates. It is noted that there is no comparison of the BEMS results with real-world energy data to establish if the proposed method is more accurate than using templates. The authors acknowledged that their article conveys preliminary results which need further research.

Tian *et al.* [49] investigated the levels of uncertainty present within BEMS from four viewpoints; 1) uncertainty sources, 2) forward and inverse BEMS methods, 3) application of uncertainty analysis, and 4) available software. Findings included the identification of a need for a transparent and open-source database to enable uncertainty quantification, occupancy behaviour was found to complicate BEMS and imposed a "*significant impact*" with no signs of changing in the near-future, Monte Carlo methods and Bayesian inverse computation were the most prevalent techniques for forward and inverse BEMS methods respectively, and the data available for quantising uncertainty was observed to be rapidly growing. The probabilistic Monte Carlo methods allow for a forward uncertainty analysis where sufficient data exists where as the non-probabilistic Bayesian techniques allow for an inverse uncertainty analysis to cope with gaps in available data.

Ascione *et al.* [50] utilised the EnergyPlus simulation engine to conduct BEMS for a historical building. The results were validated against ASHRAE Guideline 14-2002 [51]. Occupant behaviour was identified as a key factor on building energy use and recommendations on the most appropriate retrofit options were provided. The authors did not complete a post-occupancy assessment following the retrofit to verify their simulation findings.

Wang *et al.* [52] conducted an investigation on the energy use of industrial building types and demonstrated how the simulation of natural ventilation could be used to militate against overheating caused by roof lights. The simulation results showed that a 70% reduction in lighting gains could be achieved through the use of control strategies and, combined with natural ventilation in the roof, the overheating hours during the summer months were reduced. A 45% reduction in CO_2 emissions was demonstrated too. Unfortunately, the authors have not validated their base case against utility bills or measured energy data which raises questions on the reliability of their results.

Khoury *et al.* [53] reviewed 26 retrofit projects to understand the occurrence of the Performance Gap and how it can be bridged in BEMS. They cited a 30-65% Performance Gap that was found in their review of the literature. They then demonstrated their BEMS optimisation that reduced the simulated gap to 20%. The two main contributing factors that the authors identified included; 1) inaccuracies of input data, and 2) quality of input data. Energy simulations that were conducted tended to underestimate energy use compared

with measured site data across all 26 case studies when standard input parameter values were applied. The authors did not discuss how improved training and experience of BEMS practitioners should enable the use of engineering judgement to adjust standard published input values to account for the real site under assessment.

Burman *et al.* [54] highlighted the risk of national energy savings targets being missed because of the Performance Gap. The authors proposed a measurement and verification plan to calibrate a thermal model against measured data. Once the authors established a calibrated thermal model against ASHRAE Guideline 14-2002 [55] the BEMS inputs were reverted to default values proposed by the Energy Performance of Buildings Directive (EPBD), European Directive 2002/91/EC [56]. Comparison between the two sets of results demonstrated that the verified EPCD simulation underestimated the measured data by approximately 48-90%. The authors concluded that the use of standard operating conditions in energy performance makes comparison of actual performance with theoretical performance difficult. The author's case study demonstrated that a well calibrated BEMS could predict to within 10% of measured energy performance of a building.

Menezes *et al.* [57] considered the Performance Gap within non-domestic buildings by conducting Post-Occupancy Evaluations(POE). The authors identified that a performance gap of 50% was common in existing studies that ranged from 60-85%. The study identified that POE findings can be used to inform future predictions on the same or similar buildings via refinement of the BEMS. This finding was limited as it could not be conducted and early project design stages. Occupancy patterns and behaviour were identified as a key factor in the Performance Gap as people from a population don't all use buildings in the same way. Reinforced by Janda [58] with their paper titled, "*Buildings don't use energy: people do*".

Karlsson *et al.* [59] used several different BEMS software to assess low-energy houses in Sweden. The buildings were designed under the assumption of no space heating demand producing a design energy requirement of $5{,}400 kWh$. The measured energy use ranged from 3,900-5,300kWh when hot water demand and fan use was not considered. The terrace building design was assessed in three different BEMS software packages that predicted an energy use of $4{,}370\,kWh$, $4{,}490\,kWh$ and $4{,}465\,kWh$ across all three programs. These simulation results, on all three software packages, fell within the measured ranges of the

buildings. Once full building energy use was considered BEMS results predicted 50% less energy use than measured data. The root cause of this difference was established as occupant-driven behaviour such as higher indoor temperature, lower equipment efficiency and higher internal gains than assumed in simulations. This study highlighted how occupant behaviour can affect BEMS results.

De Wilde [27] conducted a pilot study to investigate the feasibility of quantifying a performance gap with consideration of uncertainties in prediction and measurement methods. This was deemed a complex process with many uncertainties excluded due to unknown distribution types meaning production of probability density functions was not straightforward. The author also made an important point that having access to measured data can lead practitioners to automatically calibrate a simulation and when attempting to quantify a performance gap the calibration should consider only the information that would be available at the design stage of a building.

Turner and Frankel [60] measured energy performance for 121 Leadership in Energy and Environmental Design (LEED) buildings to determine how building design matched the post-occupancy reality. The median Energy Use Intensity across measured building stock was found to be 24% lower in LEED buildings compared with the national commercial building stock. The Energy Star Rating was calculated at 68 compared with the national building stock of 50. BEMS had predicted 25% energy savings by using the LEED scheme and the savings were measured at 28% when compared with standard industrial baselines across the full LEED stock. It should be noted that this report has not been peer-reviewed and there is disagreement within the academic community of whether LEED buildings do save energy [61,62].

To address the prevalent deterministic approach that has created uncertainty in BEMS more studies have developed stochastic approaches that leverage cloud computing capabilities to run thousands of variations of a single simulation to identify which combination of input variables best match the measured energy data of a building but it remains very immature in its application with no set standards [43]. The papers reviewed in this subsection have identified a performance gap that can range up to 70% with a common gap quoted as 50% (i.e. measured energy use is twice that of simulated energy use).

2.2.3. Model Calibration

In using BEMS it is important that confidence in obtained results is ensured. As such model calibration has been highlighted for its importance and how it should not be overlooked. As discussed by Harish and Kumar [63], the effectiveness of a building energy control strategy depends on how the energy model has been developed and calibrated. This subsection summarises some of the latest research in energy modelling calibration techniques. Currently, an important industry standard defining BEMS calibration is the ASHRAE Guideline 14-2002 [51] which is used to provide validation maximum limits for two indices; 1) the Normalized Mean Bias Error (NMBE), and 2) the Coefficient of Variation of the Root Mean Square Error (CVRMSE). Limits on these indices have been defined as 5% and 15% respectively for a calibrated building energy model against 12 months of energy use data (e.g. energy bills).

Anil *et al.* [64] identified that as-built BIM data requires significant Quality Assurance (QA) to verify its completeness and accuracy. The authors introduced a Deviation Analysis method for assessing the quality of as-built BIM data, generated from laser scanned Point Cloud data, by analysing and classifying the patterns of deviations between existing building data and derived as-built BIM data. The process assumed the BIM data and Point Cloud agreed to within a defined tolerance. The analysis provided many benefits and the QA procedure was validated using case studies. The author's QA process identified 5.6 times more errors compared with a physical measurement method. Time savings of 40-57% were achieved, via estimations, in using the Deviation Analysis method. The authors were able to identify limitations to the QA method for further research such as 3D pattern recognition for automating classification of 3D deviations.

Coakley *et al.* [65] presented a detailed review of current approaches to energy model development and calibration, highlighting the importance of the calibration process. The authors stated that in some cases, BEMS results can vary by 100% against measured building energy use. This was accompanied by a detailed assessment of the various analytical and mathematical/statistical tools employed by practitioners to-date, as well as a discussion on both the problems and metrics of the presented approaches. The review covered the use of four different BEMS software; 1) DOE-2 [13], 2) EnergyPlus [12], 3) TRNSYS [66] and 4) ESP-r [67]. It was identified that the AEC industry has been under-utilising BEMS for

several reasons that can be broadly categorized as; 1) Modelling and 2) Calibration. The authors highlighted that when a calibration result meets a defined acceptance limit it is not a unique solution and multiple, slightly differing, models of the same building were found to be considered calibrated. Current calibration criteria do not account for uncertainty or inaccuracies of inputs or the simulated environment. Areas of uncertainty that require improvement were identified as building specifications, modelling, numerical parameters and operational scenarios. The authors concluded that calibrated simulations still require more development to provide robust BEMS reliability.

Mustafaraj *et al.* [68] built upon previous research by Bertagnolio *et al.* [69], and Bertagnolio [70], to present a method for calibration of a building model and subsequently used the calibrated model to improve energy efficiencies and reduce energy use in a building. The authors identified that although simulation is considered best practice in the building performance industry there are large discrepancies between simulations and actual measured data which is backed up by the literature reviewed in subsection 2.2.2. This finding highlighted the requirement for a robust model calibration technique. The work was carried out using a three-storey research building, the geometry of which was created in DesignBuilder [11]. Following creation of the geometry, the model definition was completed with the incorporation of a Heating, Ventilation and Cooling (HVAC) system, natural ventilation and lighting systems. The developed calibration process is illustrated in Figure 2.5. The first level of calibration utilises as-built data to validate simulation results however a second calibration level was required to achieve an improved accuracy. The second level of calibration involves collection of site data throughout the building, through metering, which produced a much greater accuracy in the simulation. The authors identified future research was required to allow finer control of HVAC systems within BEMS software and further improve the calibration process.

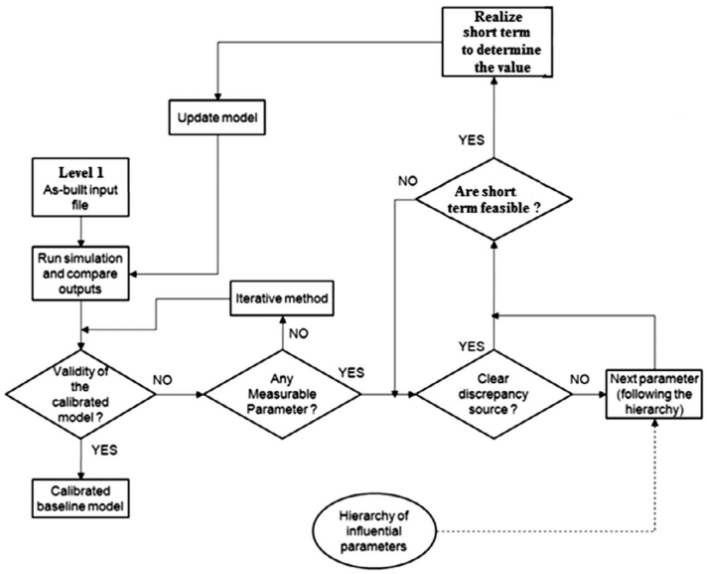


Figure 2.5. Calibration method adapted from Bertagnolio [69,70]

Gerlich *et al.* [71] presented three methods of software validation for building heat transfer simulations that utilised COMSOL Multiphysics; a numerical simulation tool [72]. The authors identified that the demand for software benchmarking tools had increased to aid validation of numerical tools. The three methods of validation included were comparison of results with 1) analytical tools, 2) other building energy software and 3) experimental results. The numerical calculations achieved positive results with the analytical solutions provided that the numerical solver was correctly set-up. When compared against existing software packages the numerical solution was found to produce the lowest mean difference against benchmarks for steady state and transient cases. The final validation technique, experimentation, observed a mean heat flux difference of 1.39% which the authors considered successful. The authors identified future work as further verification against a more complex analytical benchmark and the production of a test case study against ASHRAE Standard 140-2001 [37].

Chaudhary *et al.* [55] evaluated the "*Autotune*" method [73] for its ability to calibrate input variables to the EnergyPlus simulation engine against a manual calibration method. Autotune is a brute force approach whereby supercomputers are used to run millions of iterations of a single BEMS; each with different input variables until the most calibrated inputs are identified using utility bills as a target optimisation. The method is limited to using input variables required by the EnergyPlus simulation engine. Comparable results were found

between Autotune and the manual method however Autotune was a quicker process. The authors identified the shortcomings of existing standards that allow broad ranges of acceptable errors while dismissing uncertainty in input variables.

Yuan *et al.* [74] used EnergyPlus BEMS data to perform a Bayesian calibration on input variables while simultaneously determining the inputs that have the largest impact on the calibration. It was found through their study the most import parameter was the cooling plant Coefficient Of Performance (COP) while the natural exfiltration rate of case study buildings was the least important. While the authors acknowledged that it is impractical for a retrofit BEMS to assess the uncertainty of every input variable they stated that their process should allow for prioritisation of the many input variables.

Kim *et al.* [75] developed a BEMS calibration technique against metered electricity use by adjusting the occupancy and plug load scheduling during simulation. They also proposed a method of schedule determination where metering is not possible. Their method used a high resolution of data by considering hourly changes which provided a much higher level of granularity than many within the existing literature. The method indicated that they were able to take non-calibrated models and calibrate them to meet NMBE and CVRMSE limits of ASHRAE Guideline 14-2002 [51]. The findings of the investigation were limited to office and educational building types.

Enríquez *et al.* [76] proposed a Non-intrusive Thermal Load Monitoring method which was applied to two case study buildings in separate geographical locations. Measurements were taken over 12-months over several variables such as internal/external air temperatures, surface temperatures and weather variables. This enabled building thermal models to be calibrated against measured data. The researchers confirmed $U\text{-}values$ and HVAC air exchange rates through experimentation. While the work presented was of value for long-term studies its implementation may be impractical unless a range of sensors are mandated for inclusion in new building constructions to ensure data is available when energy assessments take place over a building lifecycle.

Harmer and Henze [77] created a calibrated energy model using ASHRAE Guideline 14-2002 [51]. Deviations between new measured and predicted energy performance were subsequently recorded over a range of time periods; daily, weekly,

monthly, quarterly and annually. By feeding in weather and solar radiation forecasts to a building simulation, operators were provided with 24-48h predictions of energy demand and forecasts of certain building variables such as zone internal temperatures. The research enabled the extension of standard capabilities of model-based building commissioning. The authors concluded that reducing the time taken to produce calibrated energy models is crucial in the industry.

Li *et al.* [78] conducted an investigation that detailed a calibration method and new performance metrics for evaluating predictive distributions in addition to comparing regression emulators to Bayesian calibration techniques. The results of the study indicated that regression emulator calibrations were simpler and quicker to conduct than traditional methods while they still provided comparable results. One limitation to the method was that it required utility bills and was therefore was only suited to retrofit assessments and not early design stages.

Roberti *et al.* [79] identified that in historic buildings the uncertainty associated with BEMS inputs such as construction materials is considerable and leads to unreliable results being produced. The authors proposed a calibration method based on measuring hourly data for indoor air and surface temperatures in a historic building case study. The method involved three stages; 1) summer calibration, 2) parameter calibration and 3) winter calibration. The authors recognised that considerable effort was required in their method implementation but that this was important in the specific case of historic buildings that had high levels of input parameter uncertainty.

Sun *et al.* [80] demonstrated a novel calibration process in four steps; 1) a non-calibrated BEMS was performed, 2) a pattern bias by comparing load shape patterns for measured and simulated results was established, 3) parameters that required tuning were automatically selected based on bias pattern, weather and input parameter interaction, and 4) parameters were tuned and then pattern fitting criteria was used to establish a Performance Gap. The method was only tested on a small case study so further evidence is required on its applicability across building stock types as well as conducting calibration with hourly data instead of monthly data.

Having reviewed available literature on BEMS has highlighted the barriers to its wider adoption and use across industry and at different points in the lifecycle of a building. These included its interoperability with BIM, the common Performance Gap observed in simulation results and methods of model calibration are all barriers. In many studies it was stated that a reduction to the time taken to complete the BEMS workflow would be beneficial. A large source of error and time consumption in the workflow included the creation of building geometry. An opportunity has been identified to produce a rapid method of building geometry reconstruction that would utilise digital survey data sets, such as Point Clouds. It was hypothesised that if such a method was successfully developed it could aid the reduction or removal of the identified barriers from the BEMS workflow.

2.3. Point Cloud Reconstruction

Reviews by Wang and Cho [81], Cho *et al.* [82] and Tang *et al.* [83] covered existing research into Point Cloud reconstruction. Key findings within their reviews included that existing documented methods of manually converting Point Cloud to BIM is error prone and time-consuming which then requires labour-intensive processes. In addition, current residential retrofit analysis was found to be primarily performed by detecting thermal defects on 2D thermal images. The reviews also highlighted that degrading heat transfer conditions of building elements, over a building's lifecycle, can lead to inaccurate BEMS results. Following on from these reviews the topic of Point Cloud reconstruction was investigated and categorised as reconstructions at a large urban scale or a smaller local building scale.

2.3.1. Urban Scale Point Cloud Reconstruction

Poullis [84] presented a framework for automatically modelling from Point Cloud data for large urban areas, see Figure 2.6, resulting in a set of non-overlapping, vastly simplified, watertight, polygonal 3D models. The author produced a robust unsupervised clustering algorithm named "P2C", which was based on a hierarchical statistical analysis of the geometric properties of the data. The developed framework was tested with large Point Cloud datasets. It was identified that although a plethora of previous techniques had been proposed for processing Point Cloud data, the gap between the state-of-the-art and the desired goal of automatic modelling from Point Cloud data remained wide.

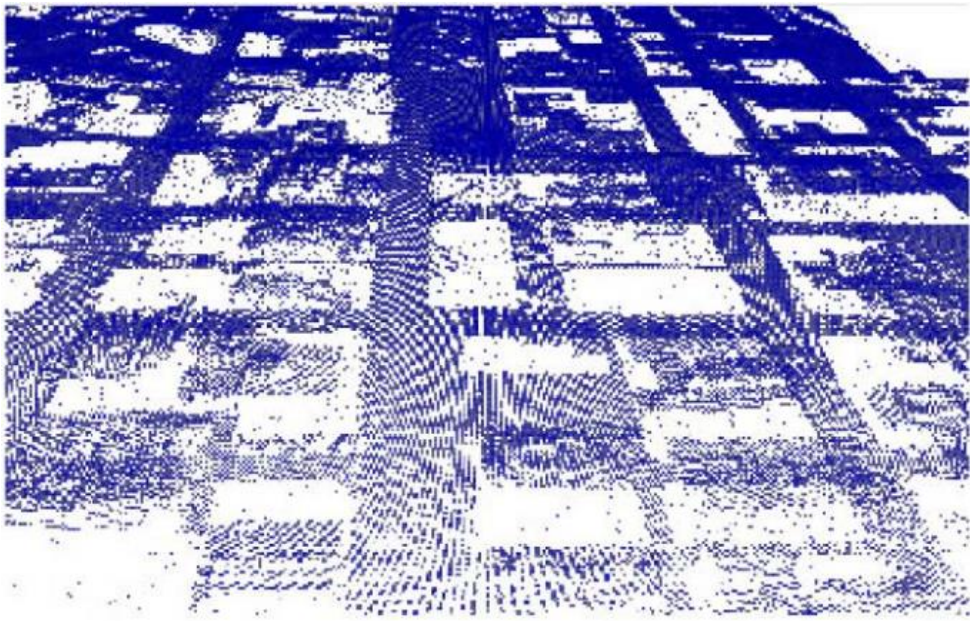


Figure 2.6. Point Cloud data for a 16km² urban area [84]

Tooke *et al.* [85] combined aerial LiDAR data with a range of BEMS input parameters such as envelope resistivity, air leaks and solar radiation gains. Figure 2.7 demonstrates how LiDAR data was used to represent nominal heat demand across an urban area in the form of a heat map. The authors validated the results against BEMS for 10 individual residential properties out of a sample size of 4,289, of which 80% were residential. This small sample size (0.3% of residential buildings in dataset) for validation raised questions over the validity of the results and no further information was provided regarding how the 10 buildings were randomly selected from the dataset. The authors also stated that the method used several key assumptions and the effect of these needed to be researched further in future work.

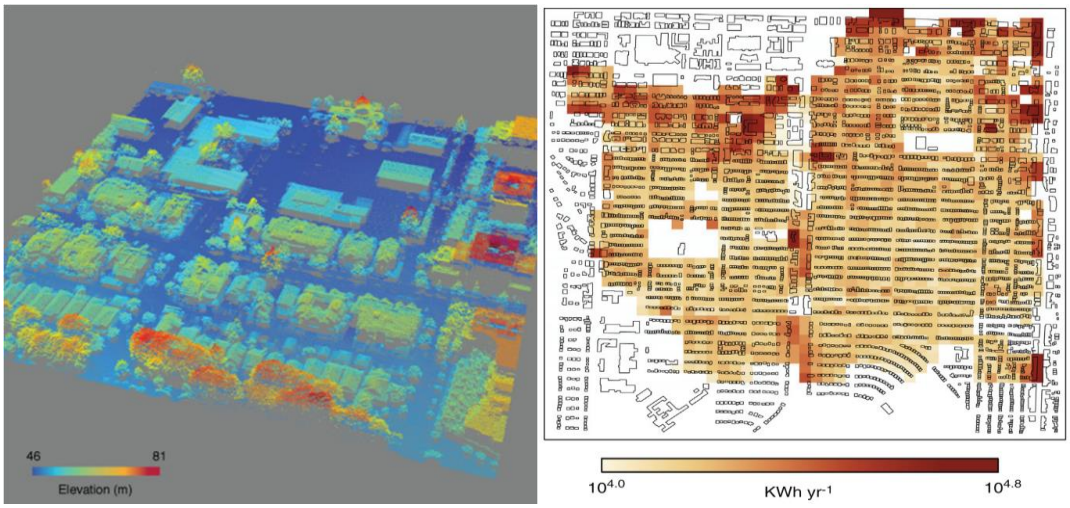


Figure 2.7. Conversion from LiDAR data to heat map of thermal energy demand [85]

Nageler *et al.* [86] used publicly available data of 1,945 buildings over $5km^2$ to develop a method of automated BEMS across urban districts. In dealing with such a large topographical area building geometry generation had been simplified by using available map data, such as that available from OpenStreetMap [87], to create single zone building floor plans and large scale LiDAR was used to detect roof heights. While the authors simulated energy use across the district the accuracy at a single building level had not been established. Thus, the recommendations for retrofit programmes were limited to the macro scale only instead of individual buildings.

Chen, Hong and Piette [88] demonstrated the retrofit analysis feature of City Building Energy Saver (CityBES) to assess energy conservation measures applied across an urban building dataset. The recommended measures included upgraded heating and cooling systems, replacement of windows, the addition air-economizers and installation of Light Emitting Diode (LED) lighting. At the urban scale these recommendations identified savings between 23-38% of site energy use and the most cost-effective measures were also identified. While the study formed a useful tool for city planners and consultants the assessment did not consider the individual building resolution of energy use.

Chen and Hong [89] subsequently assessed the impact of building geometry creation across the urban scale by applying three zoning methods to building types. The three zoning methods included 1) single thermal zone per building storey, 2) split building footprint into perimeter and core thermal zones, and 3) use the DOE reference building prototype shapes for thermal zone creation. The results showed substantial effects on simulation results, depending on which thermal zone creation method had been used. The authors concluded that practitioners must balance the practicalities of accuracy with computational time.

Ma and Cheng [90] conducted a case study on 3,640 multi-family residential buildings in order to estimate the energy use intensity across the urban scale. They used Big Data techniques for simulation parameter estimation and achieved lower estimation errors than previous research. Results were presented as heat maps with individual building geometry not considered. While sophisticated computational infrastructure was required the developed tool could aid city planning but cannot be used by individual building owners.

Buffat *et al.* [91] proposed a method using a generalised large scale geographical information system to estimate building heat demand across large urban regions. Rather than using simplistic building stock models the outlined approach captured the envelope and height of the individual buildings in the dataset. The results were validated against a large dataset of measurements which provided high reliability to their results. The advantage of this method was that it reduced uncertainty in the assessment however individual building owners may not find the data resolution high enough for their own retrofit needs.

A common thread for the urban scale energy assessments is that during reconstruction, building geometry is severely simplified. This limits the utility of the information to individual building owners but can be used by policy makers and city planners to estimate the effect of new technologies across populations. While LiDAR was used in many of these approaches other research was identified to use techniques such as photogrammetry for similar applications (e.g. Buyukdemircioglu *et al.* [92]).

2.3.2. Building Scale Point Cloud Reconstruction

Dorninger and Pfeifer [93] produced seminal work in automating building reconstruction from airborne LiDAR data. From a large LiDAR dataset, they demonstrated that a single building could be isolated and modelled using planar patches to represent the roof shapes. Detection of the roof shape led to determination of a building footprint and subsequent extrusion resulted in a 3D building volume. Whether a roof overhang was included was dependent on user input. Although individual buildings were modelled the authors then combined several reconstructions back to the urban scale, as per research described in subsection 2.3.1. The work was found to be similarly limited for assessing individual building scale energy use as it does not capture features on the vertical walls of a façade. This built upon work conducted by Verma *et al.* [94].

De Angelis *et al.* [95] used a classroom building as a demonstrator of proposed energy strategies to improve the performance of existing buildings. As part of the project the authors aimed to define an interoperability practice, in order to enhance BIM to BEMS procedures. A 3D laser scan produced a Point Cloud of the building which was then transposed into an accurate BIM model in Autodesk Revit [34]. The model was translated into a Sketch-up Model that could be utilised for BEMS by manually improving the geometry

for compatibility with the BEMS software engine EnergyPlus [12] using the OpenStudio [96]/Sketchup [97]] interface. Through BEMS the authors recorded a maximum energy reduction prediction of 37.3% however the authors provided no evidence of confirming results against real-world data by metering the building.

Wang and Cho [81,98] introduced a method of automatic as-built BIM model creation through a case study and subsequently introduced automated thermal zone creation to create a building zone and room zones. An external 3D geometry laser scan and simultaneous thermography of a residential building were taken which were mapped onto each other, see Figure 2.8 (left). 2D floor plans were used to determine the size and location of each thermal zone, see Figure 2.8 (right) The authors demonstrated automatic building model generation using the gbXML [99] schema format from thermal Point Clouds. The Point Clouds were collected from a custom hybrid data collection system that was developed. The generated gbXML model was successfully imported into a BEMS tool illustrating the feasibility of the proposed method and the potential of automating the thermal model preparation process but no energy simulations were conducted.

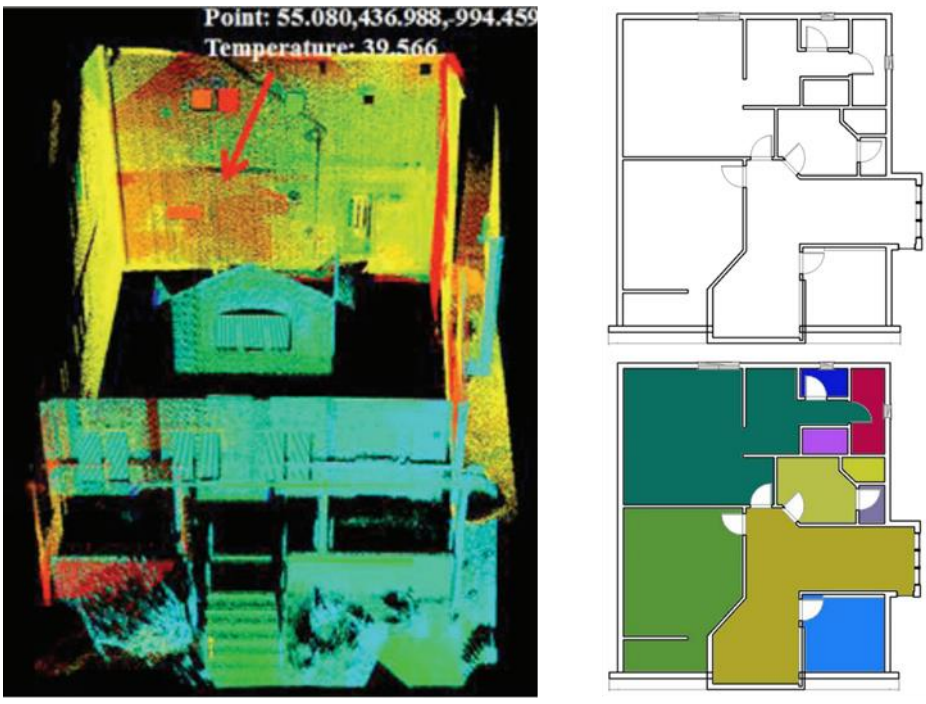


Figure 2.8. (left)3D thermal Point Cloud [81]*, (right) Original floor plan vs room zone segmented floor plan* [98]

Wang *et al.* [100] subsequently developed a technique to extract BIM features from the reconstruction and classify them accordingly (e.g. exterior/interior walls, windows, doors,

roof, slab etc.). The technique was applied to three case studies which had success at semantic BIM component identification. The captured data required downsizing to speed up processing times, but this was found to introduce errors. Incomplete captured data (e.g. roof geometry) also created gaps in the model boundary. The authors did not conduct any thermal energy simulations as part of their investigation.

Previtali *et al.* [20] presented an automated method to derive highly detailed 3D vector models of existing building façades from terrestrial laser scanning data, see Figure 2.9. The final product of the research was a semantically enriched 3D model of the building façade that could be integrated in BIM for planned maintenance. The integration between derived façade models and infrared thermography is presented for energy efficiency evaluation of buildings and detection of thermal anomalies. It is noted that the integration does not extend into a more holistic full lifecycle BEMS.

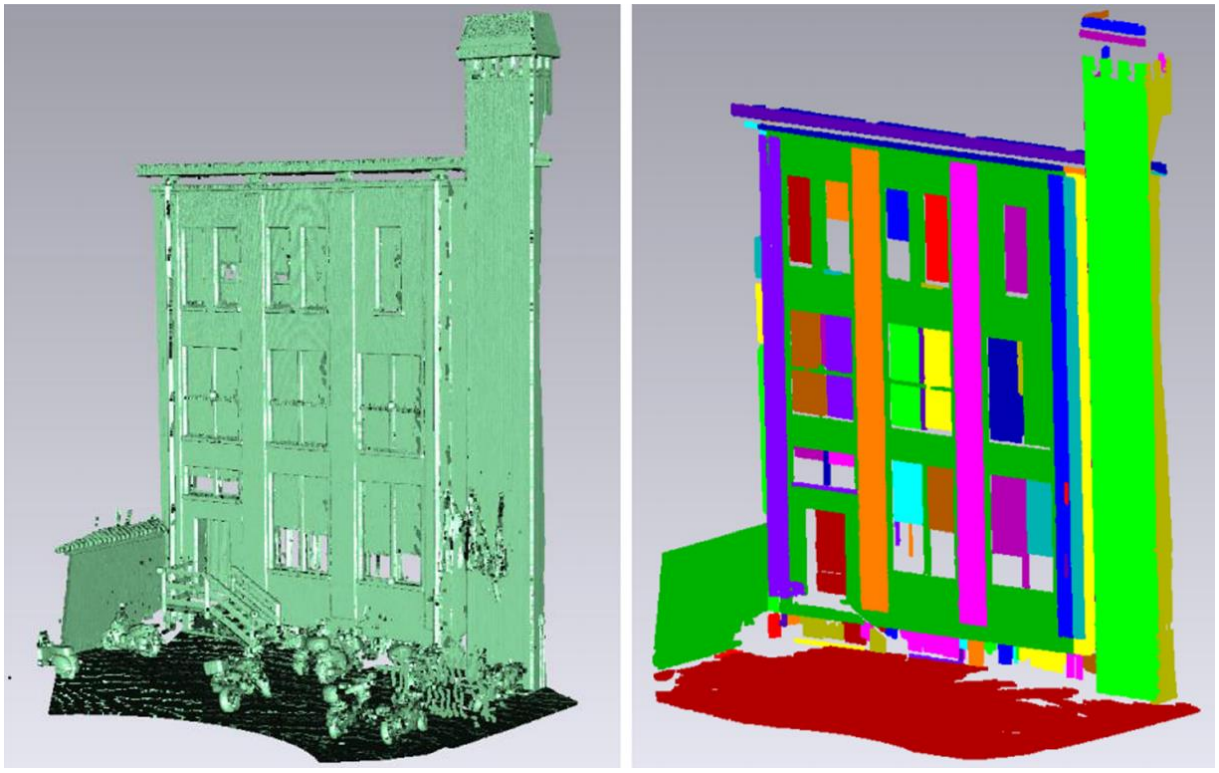


Figure 2.9. (left) Raw Point Cloud, (right) segmented Point Cloud. [20]

Thomson and Boehm [101] aimed to automate generation of 3D geometry from Point Cloud data rather than a labour-intensive manual operation. The authors investigated current technology and proposed a novel Point Cloud processing method. The proposed method only concentrated on major room boundaries; doors, windows and similar objects were

ignored. The authors concluded that there had been partial success towards the aim of fully automatic reconstruction, especially where the survey environment was simple and not cluttered. It was identified that clutter in the environment obscured the building features that needed to be reconstructed. Further work was identified to test the routine on larger datasets and produce further development of the process so that an IFC file could be used as an input and a Point Cloud could be classified from it. The authors also identified that a reduction in clutter would be beneficial to their developed process.

Armesto *et al.* [102] presented a multi-sensor acquisition system capable of automatically and simultaneously capturing the following 1) 3D geometric information, 2) thermographic, 3) optical and panoramic images, 4) ambient temperature map, 5) relative humidity map, and 6) light level map. The Indoor Multi-sensor Acquisition System (IMAS) presented consisted of a wheeled platform equipped with two 2D laser heads, a 360° Red, Green and Blue (RGB) camera, a thermographic camera, a thermohygrometer, and a luxmeter. Energy evaluation was performed via a virtual navigation allowing thermal leakages to be observed. If a thermal leak was classified as a thermal bridge (i.e. not a door or window) the position could be identified within the 3D Point Cloud. Temperature and humidity maps facilitated the detection of insulation problems in outer walls or windows, whilst the illumination map allowed evaluation of working conditions. The authors stated that this data could be transferred to an energy evaluation software. Unfortunately, the presented technique only provides a snapshot of the building construction and use in time making it difficult to predict long term future energy performance.

Rottensteiner [103] compared several methods of urban object reconstruction against benchmark data. The data gathered for testing against benchmarks comprised of both aerial LiDAR and overlapping images. The tests included tree detection and building detection. Buildings greater than $50m^2$ were satisfactorily reconstructed from roof-plane extrusion but similar to most other airborne techniques no wall information was captured. While the reconstructions were suitable for visualisation purposes, they did not have sufficient detail to be used in further AEC applications.

Bosché *et al.* [104] applied Point Cloud reconstruction techniques to cylindrical building components such as pipes and ductwork. This was found to be particularly important in

capturing as-built routing of building services that may change during an installation process. The reconstructions were compared against existing BIM models to detect any discrepancies and identify how much of an installation was built "as-planned". The results were identified as promising but limited in nature with plans presented for future work to expand this technique to more complex installations.

Lagüela *et al.* [105,106] developed techniques to combine thermography data with Point Cloud scans to reconstruct BIM data. A clear limitation to the work demonstrated is the simplistic nature of the case study building, see Figure 2.10. The mapped thermography data allowed for visualisation of the thermal bridges and weak spots in the building envelope but did not facilitate dynamic thermal modelling.

Figure 2.10. As-built building used by Lagüela et al. [106]

Elberink and Vosselman [107] performed a quality analysis on building reconstruction from airborne LiDAR datasets. This reconstruction method was also based on detecting roof shapes and extruding a building footprint from boundary edges. The datasets used contain between 60-324 buildings. The same limitations, observed in other research, of not capturing any elevation details such as walls, doors and windows still existed.

Sinha *et al.* [108] demonstrated a technique of detecting a staircase from a Point Cloud dataset. While the other research reviewed in this Chapter focuses on macro building feature detection this technique was identified as important for when macro detection techniques become suitably mature. The authors technique was found to demonstrate high potential in adding enhanced detail to reconstructed models but was isolated within the literature.

Xiong *et al.* [109] applied Point Cloud reconstruction techniques on internal building scan data and subsequently used artificial intelligence techniques for more detailed modelling.

The reconstruction attempts worked well to ignore occlusions, such as bookcases, by extending walls where these occlusions occur. This technique also worked well for the detection of openings within walls such as doors and windows.

Zhou and Gong [110] used deep learning via neural networks to construct building geometry from airborne LiDAR data. This technique had an advantage over computer vision-based approaches as surface normal orientations were not required which allowed it to be less computationally expensive. The technique was capable of large-scale urban modelling but did not include details of the building's side elevations.

Ochmann *et al.* [111] presented an automatic approach for the reconstruction of parametric 3D building models from indoor Point Clouds. Results of the reconstruction were able to be exported, as an IFC format, into BIM software. The developed algorithm was able to identify walls between adjacent rooms and reconstruct room separating wall elements, see Figure 2.11. The method included Point Cloud segmentation, generation of wall element candidates, wall segment determination, model generation and opening detection. The process was validated on 14 separate floors from 5 different buildings with good levels of success. The authors identified areas of future research to include 1) a more thorough comparison of reconstruction results with existing, manually generated models which could provide quantitative results, 2) a generalisation to multiple stories within a building would allow an entire building to be processed as a single model rather than use a separate model for each floor and 3) the usage of different capturing devices and real-time handling of streamed data. The work was conducted as part of the EC funded Durable Architectural Knowledge (DURAARK) project [112] with the developed prototypes and documentation freely available for future use, commercialisation and further development.

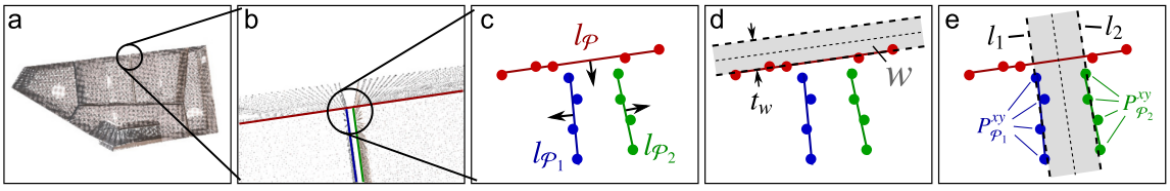


Figure 2.11. The five steps of wall candidate generation [111]

The approaches of automated building geometry reconstruction covered in this review have used a mixture of internal and external survey techniques as well as a multitude of different

data capture sensors. The workflows for internal laser scanning were found to be mature but where external scanning took place it was usually from a terrestrial scanner that could not capture the geometry of a building roof. It was identified that scanning from a UAV would solve this limitation, but substantial capital cost barriers limited the use of air-borne laser scanners for single buildings presenting an opportunity for hardware development.

2.4. Building Survey Hardware Development

Reviews by Sony *et al.* [113] and Jordan *et al.* [114] covered the increased use of smart sensors in structural health monitoring and UAV Survey technologies. To date development was found to be limited to small infrastructure and the reliability of sensors for monitoring of large infrastructure needed a more thorough investigation. Many different sensors such as visual cameras, thermal cameras, X-BOX Kinect sensors, LiDAR, Inertial Measurement Units (IMUs), Sonar and Global Positioning Systems (GPSs) were assessed but many systems were still in the early stages of development. It was identified that there was still substantial potential for unexplored research and development in these areas.

Roca *et al.* [115] mounted an X-BOX Kinect sensor on to a UAV to capture 3D geometric data of a building façade during flight. The investigation used the captured images with overlap ratios between 90-95% to apply photogrammetry techniques in creating a façade model. A Point Cloud was subsequently estimated from this data rather than captured directly by the Kinect sensor. Results were compared against a terrestrial laser scanner with the Kinect producing many more outliers in the Point Cloud. The authors highlighted the accessibility, through low cost, of the sensor also presented drawbacks in its wider applications such as the distance detection being limited to $5m$, noisy depth data and low field of view angles, see Figure 2.12. This meant the flight must be kept in close proximity to the façade and at slow speeds during data capture. The method was limited when a measurement surface received direct solar radiation (e.g. sunlight) as the Kinect signal was washed out. No effort was made to combine UAV on-board data streams such as acceleration and gyroscopic motion with the Point Cloud.

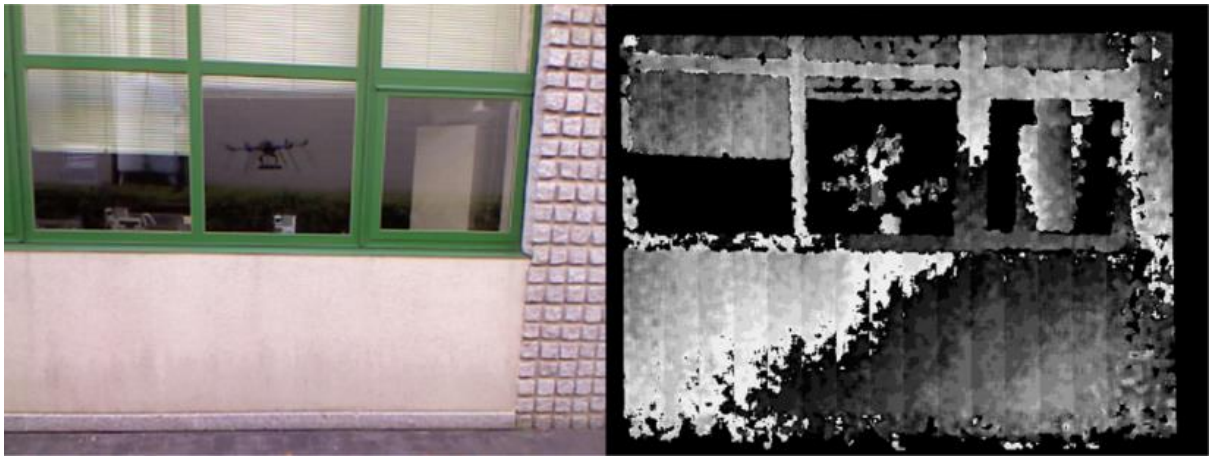


Figure 2.12. Example of photo with corresponding depth image taken from UAV [115]

Roca *et al.* [116] subsequently developed a UAV scanning system that fused data streams from an IMU, GPS sensor, central processor and a 2D LiDAR scanner. The scanner, Hokuyo UTM-30LX, is more sophisticated than that used in previous research with a range of $30m$ and precision of $30mm$. The IMU included data from a 3-axis accelerometer, a 3-axis gyroscope and a magnetometer. The resulting Point Cloud, from direct measurements, was compared with scans of a terrestrial scanner with a maximum difference in linear measurements of 5.9% in the transverse direction which was calculated through UAV positional calculations. Positional deviation between Point Cloud data and seven control points on the ground differed by a mean of 0.239m, 0.337m and 0.201m in the x, y and z directions respectively. Positive results were demonstrated but the cost of hardware used remained high as shown in Table 1.1.

Kaldén and Sternå [117] conducted the design and construction of a LiDAR unit for €125. The sensor was static with a range between $0.5m$ and $5m$ but a higher measurement frequency rate was specified to facilitate aspirational rotational scanning in future work. While the research was able to develop and verify a control system, time measurement transmitter and optics it was unable to verify the operation of an amplifier and detector of the returned signal. The authors identified that the high precision and complexity of a scanning system is a likely reason for commercial systems being so expensive, but they had taken substantial steps in breaking down these barriers through their research.

Madgwick [118,119] developed a computationally inexpensive IMU data filter that allowed the establishment of a sensors orientation, over time, in 3D space. The filter eloquently fused

the data streams from a 3-axis gyroscope, 3-axis accelerometer and a 3-axis magnetometer. At each time step of the filter a compensation is made for any anomalous gyroscope bias drift. The authors identified that IMUs error sources include sensor noise, signal aliasing, quantisation errors, calibration errors, sensor misalignment, sensor axis non-orthogonality and frequency response characteristics. The filter was found to perform more efficiently than the previously proposed Kalman filter [120] and is similar to a filter developed by Mahony *et al.*[121].

As discussed in subsection 2.3.2 Armesto *et al.* [102] built their IMAS that facilitated mobile tracking and surveying of the inside of a building. They achieved their location tracking through a technique known as Simultaneous Localisation and Mapping (SLAM) which is widely used in robotics for navigation. The technique used LiDAR data to build up a 2D horizontal map of an environment in real time as a robot navigated the space. It was stated that the technique also has been used for object and boundary detection.

2.5. Summary

The findings of the literature review are as follows;

1. The significance of BIM interoperability limitations in BEMS applications has been highlighted which creates many disincentives for retrofit BEMS analysis. These disincentives include the need for manual remodelling, time consuming workflows and poor value for money (Bahar *et al.* [10], Jeong *et al.* [33], O'Donnell *et al.* [30] and Reeves *et al.* [32]).
2. The Performance Gap between simulated and post-occupancy measurements has been found to reduce the baseline acceptance of BEMS in industry as a reliable tool. The Performance Gap is regularly quoted as 30-70% and in some cases even higher (Khoury *et al.* [53], Burman *et al.* [54], Menezes *et al.* [57], Coakley *et al* [65] and De Wilde [27]).
3. The research area for BEMS calibration was been found to be substantial in outlining a multitude of differing methods but there was no consensus on the best approach. Many of the studies take industry guidance input parameter values, such as those provide by CIBSE Guide A [122], and compare simulation results against measured

data using ASHRAE Guideline 14-2002 [51] (Harmer and Henze [77], Coakley *et al.* [65] and Mustafaraj *et al.* [68]).

4. Many Point Cloud reconstruction techniques used to create geometry from aerial surveys have vastly simplified the building façades ignoring doors and windows or reverted to simplistic heat maps instead of considering individual building geometry (Poullis [84], Tooke *et al.* [85], Nageler *et al.* [86], and Chen and Hong [89])
5. The internal Point Cloud reconstruction techniques tended to focus on simplified geometry with little to no environmental clutter (De Angelis *et al.* [95], Thomson and Boehm [101], Armesto *et al.* [102] and Ochmann *et al.* [111]).
6. The external Point Cloud reconstruction techniques either used simplified building shapes or floor plans to define internal thermal volumes. gbXML has been identified as a suitable format for BEMS data capture compared with others such as IFC (Wang and Cho [81,98], Wang *et al.* [100], Previtali *et al.* [20] and Lagüela *et al.* [105,106]).
7. Where positive results have been obtained for development of a mobile LiDAR scanner there was still a high capital cost associated with it (Roca *et al.* [116])
8. Research has shown that a static and mobile LiDAR system can be constructed at a low cost but high performance has not been achieved (Kaldén and Sternå [117] and Roca *et al.* [115]).
9. Techniques for positional tracking of a device rely on a range of different sensors and technology such as GPS and IMUs. Most methods cannot be used in isolation for high resolution tracking and it is common to see them used in combination to provide multiple reference frames (Madgwick [118], Mahony *et al.*[121] and Kalman [120]).

2.6. Research Gap

The following gaps in knowledge were identified;

1. Limited results from a BEMS reconstruction have been compared against a validated benchmark BEMS created using a traditional workflow.
2. Limited automatic reconstruction has been achieved of a complex real-world facility, using a large digital internal survey dataset, for BEMS applications.
3. Limited reconstruction methods have been assessed in comparison to the Performance Gap for BEMS applications.

4. Limited successful development has been achieved of a high performance, low-cost and mobile LiDAR unit that can be used to facilitate urban scale surveys.

The aim of this thesis, "*Develop a scientific method to remove barriers to urban scale Building Energy Modelling and Simulation (BEMS) using pattern recognition software to extract built forms from large data sets*", is intended to fill the identified research gap upon achieving the research objectives stated in Chapter 1. Completion of this literature review has fulfilled objective 1 of this research.

Chapter 3.

Building Energy Modelling and Simulation (BEMS)

3.1. Introduction

The theory and application of BEMS on a case study facility is documented in this Chapter including defining boundary conditions for a building energy model for simulation and the steps taken to calibrate and validate the results of a building energy simulation. The underpinning theory on energy transfer within the built environment is described in subsection 3.2. An overview on BEMS uncertainty and simulation validation is provided in subsection 3.3. The BEMS software IES VE [16] was selected for this research; justification and an overview of its features have been stated in subsection 3.4. A case study building, Factory 2050 (F2050), is described in subsection 3.5. The method for producing a benchmark energy model of F2050 is documented in subsection 3.6. BEMS boundary conditions are defined and established in subsections 3.7, 3.8 and 3.9. The conducted calibration procedure and method for determination of the HVAC set point temperature was detailed in subsection 3.10. The findings of the research conducted in this Chapter associated with BEMS and the validation of a benchmark energy model were summarised in subsection 3.11.

3.2. Underpinning BEMS Theory

Several variables have been identified that have a substantial effect on the energy balance of a thermal volume. When the interaction of these variables are modelled and simulated, the thermal behaviour of buildings has been predicted over a set time period. This has enabled the identification of opportunities to save energy and inform occupant comfort levels through building design and system control options. These thermal processes have been summarised in Figure 3.1.

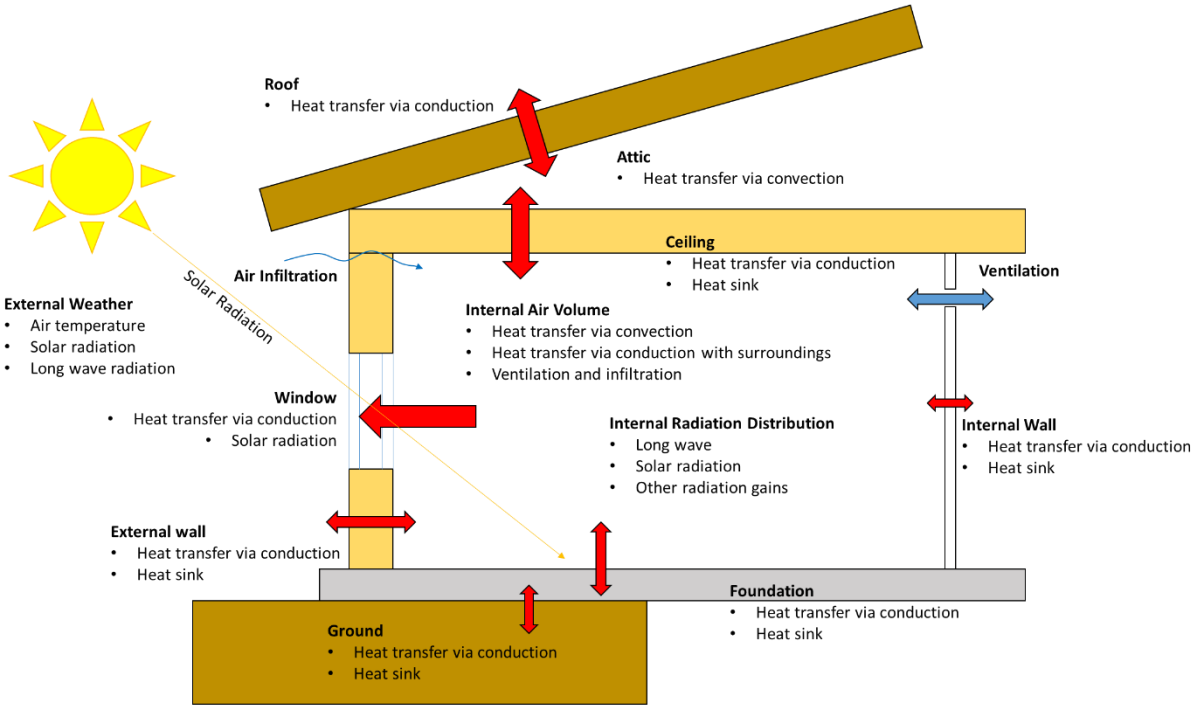


Figure 3.1. Building envelope thermal processes, redrawn from IBPT [15]

Within building control volumes, energy retained by the existing mass of air in a volume is transported through bulk mass flow. Çengal and Boles [123] defined the mass flow rate $\dot{m}$ $(kg\ s^{-1})$ as the product of the air density ρ_{air} $(kg\ m^{-3})$ and the volumetric flow rate $\dot{V}$ $(m^3\ s^{-1})$ as per by Equation 3.1. The volumetric air flow was also defined as the product of the cross-sectional area A_c (m^2) the air is moved through and the average velocity of the air as it passed through the cross-section v_{avg} $(m\ s^{-1})$.

$$\dot{m} = \rho_{air}\,\dot{V} = \rho_{air}\,A_c\,v_{avg}$$

Equation 3.1. Mass flow rate of fluid

The energy flow rate $Q_{\dot{m}}$ (W) associated with the mass flow rate $\dot{m}$ of air in a building has been defined using Equation 3.2 by considering the flow energy, kinetic energy and potential energy of the fluid. The flow energy has been calculated using the ratio of fluid pressure P (Pa) to fluid density ρ $(kg\ m^{-3})$, the kinetic energy considers the fluid velocity v $(m\ s^{-1})$ and the potential energy considers the vertical distance $z\,(m)$ of a fluid under the acceleration of gravity g $(m\ s^{-2})$.

$$Q = \dot{m}\left(\frac{P}{\rho} + \frac{v^2}{2} + gz\right)$$

Equation 3.2. Energy flow rate of fluid

Heat is transferred through different construction mediums in a building using three primary mechanisms that have been defined by Çengal and Boles [123] as follows;

1. **Conduction** has been defined as the transfer of energy from the more energetic particles of a substance to the adjacent less energetic particles as a result of particle interaction. In the built environment this is observed as energy transfer that takes place through solid boundaries such as walls, floors and ceilings. The rate of conduction $\dot{Q}_{x,cond}\ (W)$ through a layer of constant thickness $\Delta x\ (m)$ has been observed to be proportional to the temperature difference $\Delta T\ (K)$ across the layer and the area $A\ (m^2)$ normal to the direction of heat transfer. $\dot{Q}_{x,cond}$ has also been observed to be inversely proportional to the thickness of the layer Δx. The rate of conduction has been defined using Equation 3.3 where $k\ (W\,m^{-1}\,K^{-1})$ was the Thermal Conductivity of the solid substance that energy has been observed to be propagating through. As $\Delta x \to 0$, Equation 3.3 has been redefined to its differential form. This redefinition is called "*Fourier's Law of heat conduction*" where the rate of heat conduction has been observed to be proportional to the temperature gradient in the direction of heat transfer which always occurred in the direction of lower temperatures creating a negative gradient as defined in Equation 3.4.

$$\dot{Q}_{x,cond} = k\,A\,\frac{\Delta T}{\Delta x}$$

Equation 3.3. Rate of heat conduction

$$\dot{Q}_{x,cond} = -k\,A\,\frac{dT}{dx}$$

Equation 3.4. Fourier's Law of heat conduction

2. **Convection** has been defined as the transfer of energy between a solid surface and an adjacent fluid that is in motion. The heat transfer mechanism has been observed to comprise combined effects of conduction and fluid motion. In the built environment this is experienced as energy transfer that takes place from solid boundaries such as walls, floors and ceilings and into a thermal fluid volume such as

a room full of air. The rate of heat convection $\dot{Q}_{x,conv}$ (W) has been defined using "*Newton's law of cooling*" where heat transfer has been observed to be proportional to 1) the surface area A (m^2) through which heat transfer takes place and 2) the difference between the surface temperature T_s (K) and bulk fluid temperature T_f (K) away from the surface. $\dot{Q}_{x,conv}$ has been defined using Equation 3.5 where $\hat{h}$ $(W\ m^{-2}\ K^{-1})$ is the convection heat transfer coefficient.

$$\dot{Q}_{conv} = \hat{h}\ A\ (T_s - T_f)$$

Equation 3.5. Rate of heat convection

3. **Radiation** has been defined as the transfer of energy due to the emission of electromagnetic waves (or photons). When thermal heat transfer is considered, only thermal radiation is applicable and other radiation forms such as X-rays, gamma rays, microwaves etc. are not applicable. In the built environment radiation is measured as the energy transfer from radiation sources onto adjacent surfaces such as external solar radiation onto a building façade or radiation emitted by occupants inside the building envelope onto surrounding surfaces. All bodies of matter have been observed to emit radiation when their temperature is above absolute zero (i.e. $0\ K$). Although radiation is a volumetric phenomenon, in heat transfer calculations it is treated as a surface phenomenon when dealing with materials opaque to thermal radiation such as metals and wood. The radiation emitted by the interior of these materials cannot reach the surface and incident external radiation is only absorbed by a very thin layer of the materials surface. When the Stefan-Boltzmann law is considered, as defined in Equation 3.6, the net heat transfer by radiation $\dot{Q}_{rad}$ (W) between two surfaces is calculated using Equation 3.7 where $\dot{Q}_{emit,max}$ (W) was the maximum rate of radiation that can be emitted from a blackbody, A (m^2) was the surface area at temperature T_s (K), σ $(5.67 \times 10^{-8} W\ m^{-2}\ K^{-4})$ was the Stefan-Boltzmann constant, ε was the emissivity of the surface and T_{surr} (K) was the temperature of the surrounding surface that was separated by a non-radiation intervening gas (e.g. air).

$$\dot{Q}_{emit,max} = \sigma\ A\ {T_s}^4$$

Equation 3.6. Stefan-Boltzmann law

$$\dot{Q}_{rad} = \varepsilon\, \sigma\, A \left({T_s}^4 - {T_{surr}}^4\right)$$

Equation 3.7. Net heat transfer by radiation

When heat transfer through the boundaries of a defined control volume have been considered, as the geometry of surfaces surrounding the volume were known, it has been common to calculate in terms of heat flux Q $(W\ m^{-2})$ through a boundary. The heat flux for conduction, convection and radiation are thus calculated using Equation 3.8, Equation 3.9 and Equation 3.10 respectively.

$$Q_{x,cond} = -k\, \frac{dT}{dx}$$

Equation 3.8. Conduction heat flux

$$Q_{conv} = \hat{h} \left(T_s - T_f\right)$$

Equation 3.9. Convection heat flux

$$Q_{rad} = \varepsilon\, \sigma \left({T_s}^4 - {T_{surr}}^4\right)$$

Equation 3.10. Radiation heat flux

Mathematically the calculations used in BEMS have been described as analogous to an electrical circuit, complete with resistors and capacitors, such that the thermal conditions in each thermal volume are represented by a single central node, see Figure 3.2. The resistors in the analogy are used to represent the ability of air to transport thermal energy via convection and radiation to and from surfaces while the capacitors are used to represent the thermal transfer via conduction across a boundary such as a wall, floor or roof. A capacitor is used in this analogy as the solid boundaries are more capable of storing thermal energy than the working fluid of each thermal volume which, in the built environment, is replaced at regular intervals.

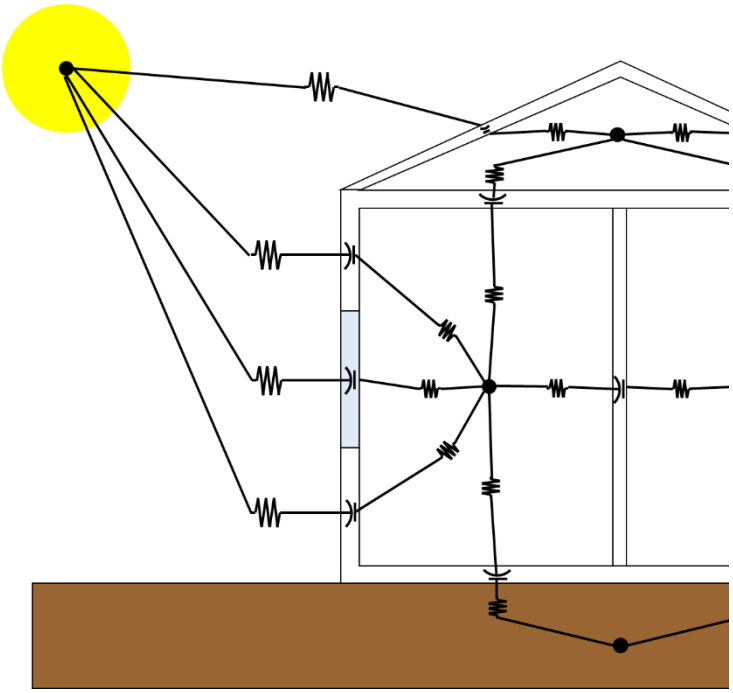


Figure 3.2. BEMS electrical circuit analogy

When the basic principles of heat transfer theory have been expanded, and the many different transfer mechanisms present within the built environment have also been considered, the heat balance $\dot{Q}_{node}$ (W) at a single interior air mass node has been calculated using Equation 3.11. These considerations have included the effects of ventilation $\dot{Q}_{vent}$ (W), gains from space heating & cooling $\dot{Q}_{htg,clg}$ (W), internal gains $\dot{Q}_{conv,source}$ (W) and convective heat transfer from internal walls $\dot{Q}_{conv,int\_surf\_zone}$ (W) where ρ_{air} $(kg\ m^{-3})$ was the air density, c_p $(J\ kg^{-1}\ K^{-1})$ was the specific heat capacity of air, V (m^3) was the thermal zone volume, T (K) was the room temperature and t (s) was time [124], see Appendix A for further information.

$$\dot{Q}_{node} = \rho_{air}\ c_p\ V\ \frac{dT}{dt} = \dot{Q}_{vent} + \dot{Q}_{htg,clg} + \dot{Q}_{conv,source} + \dot{Q}_{conv,int\_surf\_zone}$$

Equation 3.11. Heat balance of interior air mass node and contributing gains

BEMS has been developed as a time-series driven simulation, instead of event-driven, and thus the conditions at each node within a thermal model are calculated and updated at each time step in a BEMS calculation. The calculations are conducted in this manner to achieve a stable thermal balance throughout the network of thermal nodes [9].

The thermal transmittance across the individual thermal zone boundaries is always considered and therefore has been identified as a key BEMS parameter. The thermal transmittance is commonly referred to as the $U\text{-}value$ $(W\ m^{-2}\ K^{-1})$ which was defined as the rate of transmission of energy through a unit area of a material for a one-degree difference through the thickness of the material. The $U\text{-}value$ of a boundary, such as a wall, is directly related to the thermal resistance, R $(W^{-1}\ m^2\ K)$, of the individual construction element materials that are used in the boundary as defined by Equation 3.12.

$$U\text{-}value = \frac{1}{\sum R}$$

Equation 3.12. U-value definition

The thermal resistance, R, of a construction element is calculated for each element within a construction boundary in relation to the thickness of the element, x (m), and the Thermal Conductivity of its material, k $(W\ m^{-1}\ K^{-1})$, as defined in Equation 3.13.

$$R = \frac{x}{k}$$

Equation 3.13. Thermal resistance definition

Having knowledge of the different construction elements that constitute a building and the respective $U\text{-}value$ of each construction boundary has allowed practitioners to determine the heat loss from a building. This determination has been based upon the assumption that heat loss is directly proportional to the temperature difference between the external and internal air temperatures, as defined in Equation 3.14 [125]. The proportionality constant is commonly described as the Total Loss Coefficient (TLC) $(W\ K^{-1})$ which is comprised of two components; 1) thermal losses through the building fabric and 2) thermal losses due to ventilation effects.

$$Heat\ loss = TLC\ \times\ T$$

Equation 3.14. Building heat loss

The fabric component of TLC is calculated by considering heat loss through a boundary and through thermal bridges within a boundary (e.g. around a window). The heat loss through a boundary is calculated by the summation of the product between the $U\text{-}value$ $(W\ m^{-2}\ K^{-1})$ and Cross-Sectional Area, A (m^2) for each thermal boundary. The heat loss through the

thermal bridges of a boundary is calculated by the summation of products of the length l (m) of each thermal bridge with the linear thermal transmittance of each thermal bridge Ψ $(W\ m^{-1}\ K^{-1})$, as defined in Equation 3.15.

$$TLC_{fab} = \sum(U\text{-}value\ A) + \sum(l\ \Psi)$$

Equation 3.15. Calculation of fabric component of TLC

The ventilation component of TLC is calculated using Equation 3.16 where ρ is the density of air (assumed as 1.2 $kg\ m^{-3}$), c_p is the specific heat capacity of air (assumed as $1000kJ\ kg^{-1}\ K^{-1}$) and $\dot{V}$ $(m^3\ s^{-1})$ is the volumetric flow rate of air in the building.

$$TLC_{vent} = \rho\ c_p\ \dot{V}$$

Equation 3.16. Calculation of ventilation component of TLC

Using the stated assumptions on the density and specific heat capacity of air the ventilation component of TLC is commonly simplified as defined in Equation 3.17 where ach is the number of air changes per hour in the thermal volume of the building (h^{-1}) and V (m^3) is the total thermal volume of the building.

$$TLC_{vent} = 0.33\ ach\ V$$

Equation 3.17. Simplified ventilation component of TLC

The TLC for an entire building is calculated by combining the fabric, TLC_{fab}, and ventilation, TLC_{vent}, components as defined in Equation 3.18.

$$TLC = TLC_{fab} + TLC_{vent}$$

Equation 3.18. Calculation of TLC

With knowledge of the TLC of a building the expected energy consumption to heat and cool a space is determined using the Degree Day method. Heat transfer has been observed to be driven by temperature gradients. When the external temperature of a building has differed from the internal temperature heat transfer has been determined using the TLC. Degree Days have been defined as the difference between a baseline temperature and the external temperature of a building multiplied by the number of days that the difference persists. In the UK the standard baseline temperature for heating is selected as 15.5°C [122,125] as at this temperature it has been determined that most buildings are capable of relying on gains from

solar radiation, occupants and equipment to provide the necessary heating required for comfortable building use. However, the baseline temperature is on occasion calculated for each specific site so that for each month a base temperature has been defined to enable more detailed calculations.

The Degree Day method was primarily developed for determining heating demand. When the external air temperature is observed to fall below the defined baseline temperature a building develops a heating requirement and the Degree Days that require heating are known as Heating Degree Days (HDD). The expected heating demand (kWh) is determined using Equation 3.19 to enable specification of equipment, such as boilers, to be installed in a building to meet the necessary demand.

$$Heating\ Demand\ (kWh) = TLC\ (W\ K^{-1})\ \times HDD(K\ day) \times 24(h\ day^{-1})$$

Equation 3.19. Heating demand for a building

In a similar manner the cooling demand (kWh) of buildings is determined through calculation of the Cooling Degree Days (CDD) that enables specification of cooling equipment such as air conditioning, as defined in Equation 3.20.

$$Cooling\ Demand = TLC\ \times CDD\ \times 24$$

Equation 3.20. Cooling demand for a building

CDD have been defined as the Degree Days when the external air temperature is above the baseline temperature however additional consideration is given when defining the base temperature for CDD which has been observed to be dependent on the cooling system used. Several other variables such as fluctuating solar gains and wet-bulb temperatures are also considered in CDD calculations which are not directly related to the external dry-bulb temperature [125].

The simplified calculation method that has been outlined has enabled manual calculations with relative ease for simple geometric shaped buildings over long time periods where a suitable resolution of weather data has been available. The limitations of these calculations are well documented because of their simplistic nature. The method outlined did not consider the thermal mass of a building which has been shown to slow down the heating and

cooling of a building by retaining energy in a manner similar to the electrical circuit analogy provided in Figure 3.2 [125].

When the geometry of buildings being assessed has increased in complexity, manual calculations have become more time consuming and difficult. BEMS software expands on these fundamental building blocks within building physics and they are applied in a differential manner, along with input data, to simulate the expected heating and cooling demands as well as the total energy use of a building over a simulation period. This is usually 12-months consisting of 8,760h of individual time steps to be solved for the model under simulation. The use of modern computational power has enabled building simulations to be conducted at a much higher speeds and with a large number of thermal zones than manual calculations regardless of building complexity. BEMS has therefore transformed the ability to perform energy assessments on buildings as it has been designed to consider the complex interaction between a building and its environment, occupants and use.

In this regard, it is common for the calculation process to be hidden from BEMS practitioners. In this "black box" style arrangement input parameters are provided to BEMS software which produces an output for analysis by a competent practitioner. For this reason, it is important for all BEMS practitioners to have gained an understanding of the uncertainties associated with input parameters and the accepted and appropriate validation methods to aid in the calibration of BEMS.

3.3. BEMS Uncertainty and Validation

It has been identified within scientific literature that all BEMS input variables have an associated natural variation and uncertainty in how the chosen value, provided to BEMS software, was representative of the as-built condition [27]. Many different BEMS software, that are used throughout industry, allow differing user assumptions. This has enabled BEMS to be conducted as a deterministic simulation. A paradox in this method has been identified in that many of these input variables are either stochastic in nature or change over time.

Despite this paradox, an understanding of the sources of uncertainty in BEMS has helped minimise input uncertainty and calibrate simulation results. The calibration of simulation results has allowed the results to accurately reflect measured building energy use data. Four

key uncertainties associated in BEMS have been categorised as 1) weather data, 2) building envelope, 3) HVAC System and 4) Occupant behaviour [49,65,126–129].

Weather data that has been collected at meteorological stations does not represent the exact microclimate variations of individual buildings that have undergone BEMS assessment. The recording stations are generally found to be remote from an assessed building and thus while the weather data has been found to be broadly representative of the climate the building is exposed to, it does not represent the exact weather conditions experienced on site by a building. It is common practice for the weather file, that is selected for BEMS, to be chosen from the closest proximity recordings that have a similar topography. This ensures that when a weather station has recorded any extreme data points (e.g. dry-bulb temperature) there was a high likelihood of similar conditions on the site of the building being assessed. Wang *et al.* [129] quantified the effect of using different weather files from a 10-15year period on BEMS energy use predictions as ranging from -4.0% to +6.1% when compared to a standard single weather file that was used to represent typical weather over the same time period, see Figure 3.3. From these results it was observed that when validating BEMS uncertainty in weather data inputs can be minimised by using weather from a specific year that matches the year of metered energy data.

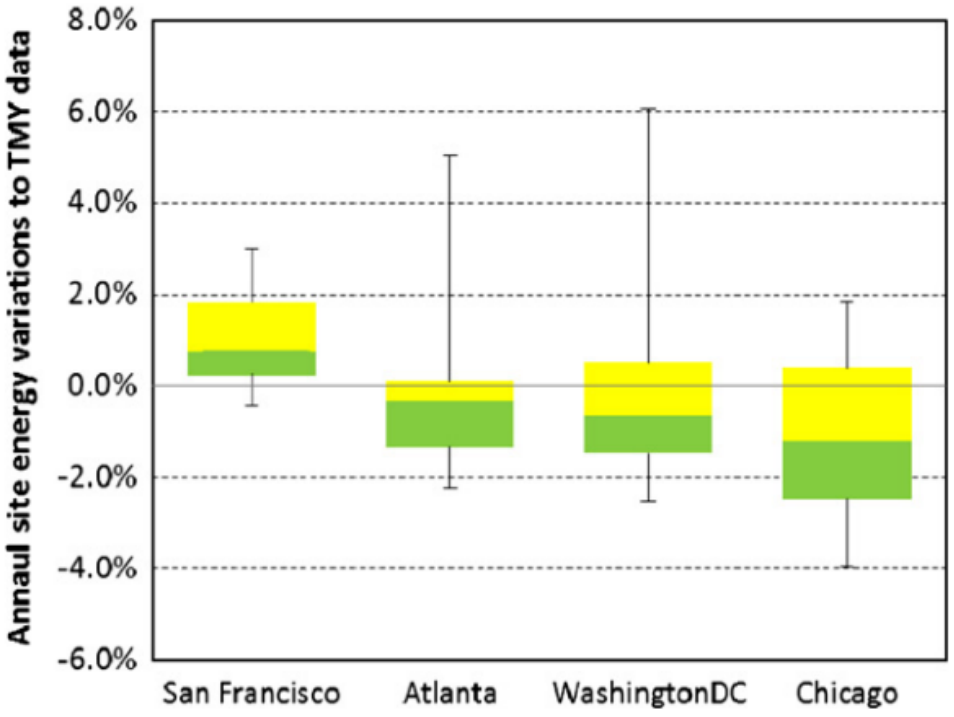


Figure 3.3. BEMS uncertainties observed by Wang et al due to actual weather [129]

To fully define a building envelope for BEMS several input variables are used. These variables have been identified as the thermal properties of the building envelope (e.g. $U\text{-}value$, thermal bridges, convective heat transfer coefficient, thickness of construction elements etc.), surface properties for the emission and absorption of radiation, and air infiltration rates [49]. As with the majority of BEMS input parameters, for black-box style software, a single value is usually specified for each parameter based on industry published data. In reality, the as-built values of these variables are found to have a distribution away from the single selected value over an entire building construction. The maintenance regime of a building is also observed to affect the value of these parameters over the lifecycle of a building due to natural deterioration of construction materials. Several studies have considered the uncertainty present when defining these variables such that adjustments can be made during BEMS calibration [130–133]. Kim, Shen *et al.* [41] found differences in BEMS results ranging 6.9% by varying the construction materials definitions. However, a notable variable, compared with others in this category, was identified as the air infiltration rate. This was found to have greater effect on energy use results and had a high level of uncertainty associated with the defined BEMS parameter. More details on methods used to define the air infiltration rate has been described in subsection 3.9.

In BEMS there are broadly two methods used to define a HVAC system; 1) a generic and idealised system that was defined through the use of mechanical efficiencies and coefficient of performance values, and 2) a fully defined HVAC system that has included consideration of the components incorporated into the system and their overall efficiencies. The choice of which method is used in BEMS is highly dependent on the information that was available to a BEMS practitioner or the stage of building design at the time of BEMS. Where BEMS uncertainty is considered for HVAC systems it is usually implemented through variations in equipment efficiency values. The investigation by Wang *et al.* [129] identified variations in building energy use ranging from -15.8% to +70.3% due to individual HVAC parameter variation; this was identified as the most influential category in their investigation.

When internal gains of a building have been defined assumptions were required into how the population of occupants behave. These assumptions included the number of occupants, the occupancy schedule and equipment available for use by the occupants. Unfortunately, for BEMS practitioners, it has been observed that building occupants to not behave as a

homogenous group. This has led to defined BEMS variables differing from the individuals in a population. Eguaras-Martínez *et al.* [48] have demonstrated that by including accurately monitored occupant behaviour in BEMS, the simulation was able to predict a difference of approximately 30% in energy use of a building. Wang *et al.* [129] have shown that the variation of occupant behaviour and input assumptions have led to a difference in building energy use that ranged between -11.3% and +7.0% for plug loads. Similarly, their investigation identified that variations in the use of lighting also affected the energy use of a building in the range of -5.8% to +9.0%.

In consideration of all the uncertainties identified by Wang *et al.* [129] the total energy use of a building has been demonstrated to vary substantially (i.e.-28.7% to +79.2%). This discussion on BEMS uncertainty has demonstrated the importance of reducing uncertainty in BEMS input parameters where possible and practical. BEMS validation techniques are used to compare how well BEMS results match metered data. A validated BEMS is used to provide a reliable benchmark to inform any decisions around retrofit options with the aim of a reduction in building energy use.

As such a benchmark of the existing BEMS workflows and method was established through consideration of a case study building F2050, see subsection 3.5. This enabled the measurement of subsequent research outcomes of automated reconstruction from digital survey datasets. Based on the literature review in Chapter 2, it was clear that there was no agreed method on how a BEMS model, and all its various input parameters, should be calibrated. However, many of the research papers relied on assessing produced results, once calibrated, against the criteria of ASHRAE Guideline 14-2002 [51] to declare the model as validated.

A produced BEMS model which has been validated against ASHRAE Guideline 14-2002 [51] is utilised to provide confidence in the boundary conditions applied during thermal simulation of a building. While a validated model has been observed to not necessarily be a unique solution, using industry guidance and the experience of BEMS practitioners can increase the probability of correctly selecting boundary conditions when a validated model is achieved.

When the measurement and validation of data is considered, several different statistical indices have been used to measure how well simulation results match metered data. It was

agreed within industry that focussing on two primary metrics was the most useful for characterising model performance and validation. Many candidate statistical indices exhibited large overlaps and as the number of indices used increased it was accepted that it became more difficult to aggregate results and draw any meaningful conclusions [134].

Two validation indices, specified by ASHRAE Guideline 14-2002, were determined to be the most suitable options; NMBE and CVRMSE [51,134]. These indices have been defined by Equation 3.21 and Equation 3.22 respectively. ASHRAE Guideline 14-2002 specified that $p = 1$ in these calculations that accounted for the number of adjustable model parameters. Reddy and Maor [135] suggested that for calibration purposes $p = 0$ should be used when calculating NMBE. However the ASHRAE method (i.e. $p = 1$) has been shown to produce a more conservative NMBE calculation by approximately 9% (i.e.$\frac{11}{12}$) when using monthly metered energy data.

$$NMBE = \frac{\sum^{n} (\dot{y}_i - \hat{y}_i)}{(n-p) \times \bar{y}} \times 100$$

Equation 3.21. Definition of NMBE

$$CVRMSE = \frac{\sqrt{\frac{\sum(\dot{y}_i - \bar{y}_i)^2}{(n-p)}}}{\bar{y}} \times 100$$

Equation 3.22. Definition of CVRMSE

NMBE was defined as a normalisation of the mean of the errors across a dataset. The mean of errors in a dataset, which forms the basis of the NMBE index, is calculated by the sum of differences between corresponding measured and simulated data points within a dataset divided by the number of measured data points. The mean of the errors is normalised by dividing by the mean of actual measured values. It has been asserted that as NMBE does not consider the absolute errors the index, when calculated, was liable to cancelation effects between negative and positive values at each data point. It has been recommended that the NMBE index was not relied upon in isolation [128].

CVRMSE was defined as the measure of variability of errors between simulated and measured values. CVRMSE has been calculated to establish whether predictive results obtained through BEMS have accurately matched the load shape observed in the measured

data. CVRMSE was found to not exhibit the cancellation effects observed with the NMBE index so has been used in conjunction with NMBE to provide a more complete understanding of model validation [128].

For validation, ASHRAE Guideline 14-2002 [51] specified limits on NMBE and CVRMSE. If the measured data only consisted of monthly data, then the limits were set at 5% for NMBE and 15% for CVRMSE. For hourly measured data these limits were increased to 10% and 30% respectively.

Table 5-2 of ASHRAE Guideline 14-2002 [51] outlined the compliance requirements for whole building calibrated simulation which are briefly summarised as follows;

1. Measured energy data shall be available (e.g. utility bills)
2. Energy use measurements must be over a continuous time period
3. Measured data must span at least 12 months
4. A minimum of 12 data points must be provided from measured data
5. The BEMS results shall have uncertainties calculated that fall below the maximum thresholds for NMBE and CVRMSE.
6. The person conducting the simulation shall be adequately experienced
7. The simulation tool shall be a computer-based program that performs calculations for each hour of the simulation window (i.e. 8,760h)

Based on these compliance requirements it was important that an appropriate BEMS software was selected for use in further research. The validation methods that have been outlined relied on the use of metered energy use data. While this method is favourable it is not used on BEMS for new construction in the absence of energy bills or measured data. In this scenario BEMS practitioners have been known to take precautions to minimise uncertainty in the simulation and another method of validation has been required. Acceptable complementary validation techniques, in the absence of measured data, have included comparison of BEMS results 1) against hand calculations, 2) across multiple software packages, and 3) against CIBSE benchmark data [136].

3.4. BEMS Software

There were many different individual BEMS software packages that could have been used in this research and they were found throughout academic literature. IES VE [16] was selected

for use in this research as it was found to outperform other BEMS software for usability, range of simulation inputs and interoperability [32]. Regardless of the individual software package used, the workflow of a BEMS practitioner was found to contain common elements for consideration, as illustrated in Figure 3.4.

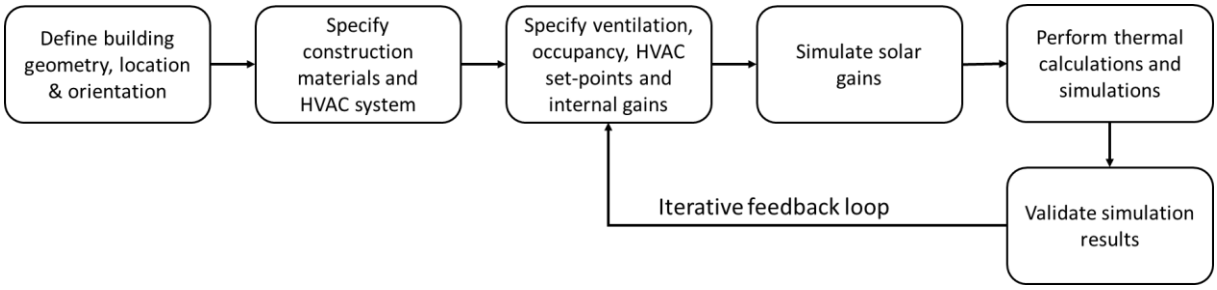


Figure 3.4. Existing BEMS workflow

When IES VE [16] was loaded, by default its ModelIT module was presented; one of several modules used IES VE. The key modules used in this research included ModelIT, SunCast, Apache, MacroFlo and VistaPro. ModelIT was used to define the geometry of a building including the doors and windows. SunCast was used to simulate the solar gains incident on the building based on its geometry, location and orientation. Apache conducted the dynamic thermal modelling and simulated how the many BEMS parameters interacted over the course of a simulation year. MacroFlo was used to specify how airflow occurred around building features such as doors and windows. The VistaPro module was used to access the results of the simulation conducted in Apache.

IES VE also contained a building template manager that was accessible from any module. This allowed the methods of building use such as HVAC set points, occupancy schedules, internal gains and construction thermal properties to be defined. These parameters been explained further through the presentation of a case study in subsection 3.5.

3.5. Case Study Building

A case study building is required to demonstrate how the research that has been developed for this thesis performs in real-world scenarios which provides tangible research value. Its importance is highlighted through inclusion of objective 2 in Chapter 1.

3.5.1. Description

As part of the University of Sheffield's Advanced Manufacturing Campus F2050 was created as a demonstrator plant for new manufacturing technologies and research. F2050 comprised a large circular workshop that was surrounded by external glazing with offices located in the centre, and a large rectangular portion extending away from the circular workshop that had no external glazing. This lack of glazing enabled the use of F2050 for conducting work of a more sensitive commercial nature, see Figure 3.5.

Figure 3.5. (left) Artistic render of F2050 [137], *(right) Aerial View of F2050 under construction* [138]

The building is of an industrial construction type, but its research and development activities meant that compared to other industrial buildings it had a low industrial energy footprint (e.g. no assembly lines). Building occupants had reported thermal discomfort and the metered energy data for the building was identified by the operators as being higher than other comparable building types on the Advanced Manufacturing campus. The building was selected as a case study for this research because of the permitted site access and the availability of metered energy data.

3.5.2. Metered Energy Use

The metered energy data for the facility was provided via utility bills for the full 12-months of 2016 as illustrated in Figure 3.6. Redacted copies of the utility bills have been provided in Appendix B.

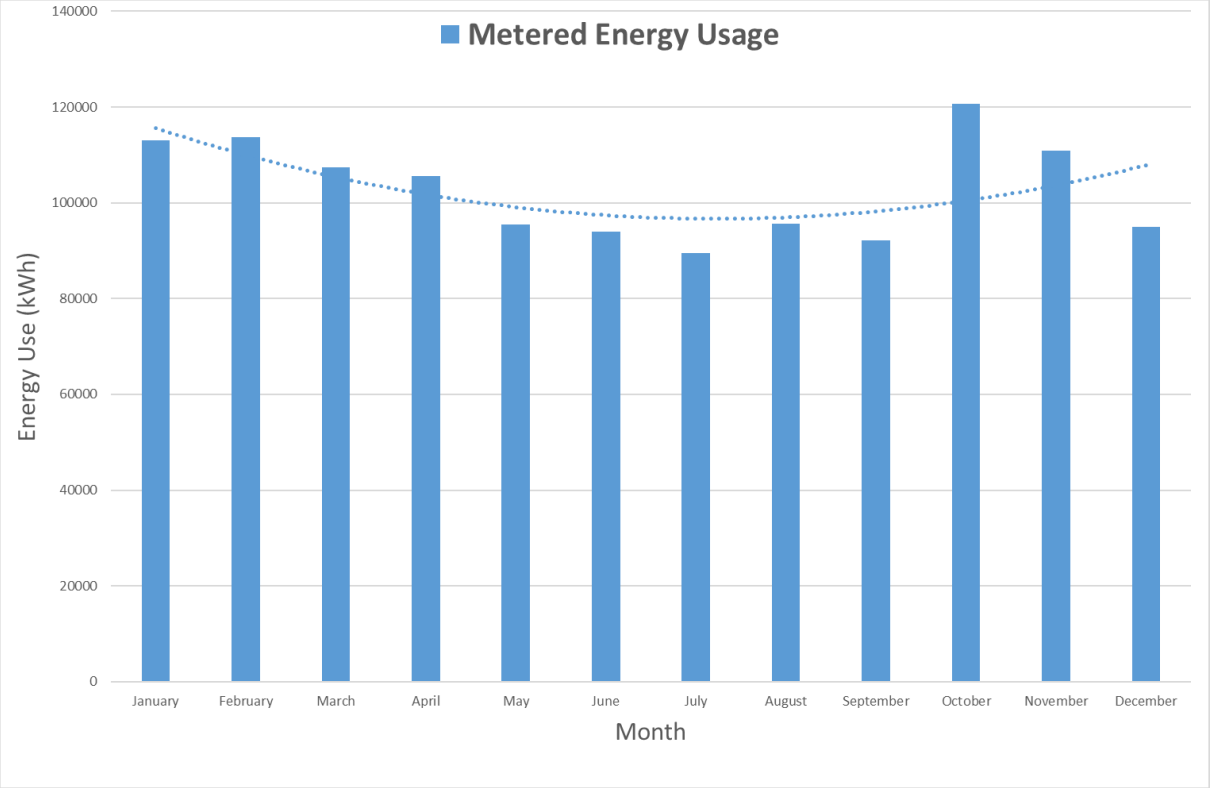


Figure 3.6. F2050 metered energy use in 2016

The metered energy data for F2050, that was used in this research, was only available at monthly resolutions. As such all simulated BEMS results were compared against the 5% and 15% limits for NMBE and CVRMSE respectively as stated in subsection 3.3. Figure 3.6 illustrates a spike in energy use in October 2016 that does not appear to match the trend of the other 12-months but it is unknown as to the cause of this. At the time of measurements, the building was still relatively new so it is possible that some commissioning work was conducted which involved leaving access doors opened for prolonged periods. Coupled with the dropping temperatures in October then a spike in energy use would be observed.

In addition to the metered energy data, BIM data of F2050 was supplied in an IFC format however as energy analysis was not considered important to the BIM author it could not be effectively imported into the BEMS software IES VE [16]. This reinforced the real problem of BIM interoperability that was discussed in Chapter 2.

Even though the IFC file could not be imported into IES VE [16], it was imported into Autodesk Revit [34] which enabled measurements of the building geometry to be taken in developing

a benchmark energy model. The F2050 model, as viewed in Autodesk Revit, is illustrated in Figure 3.7.

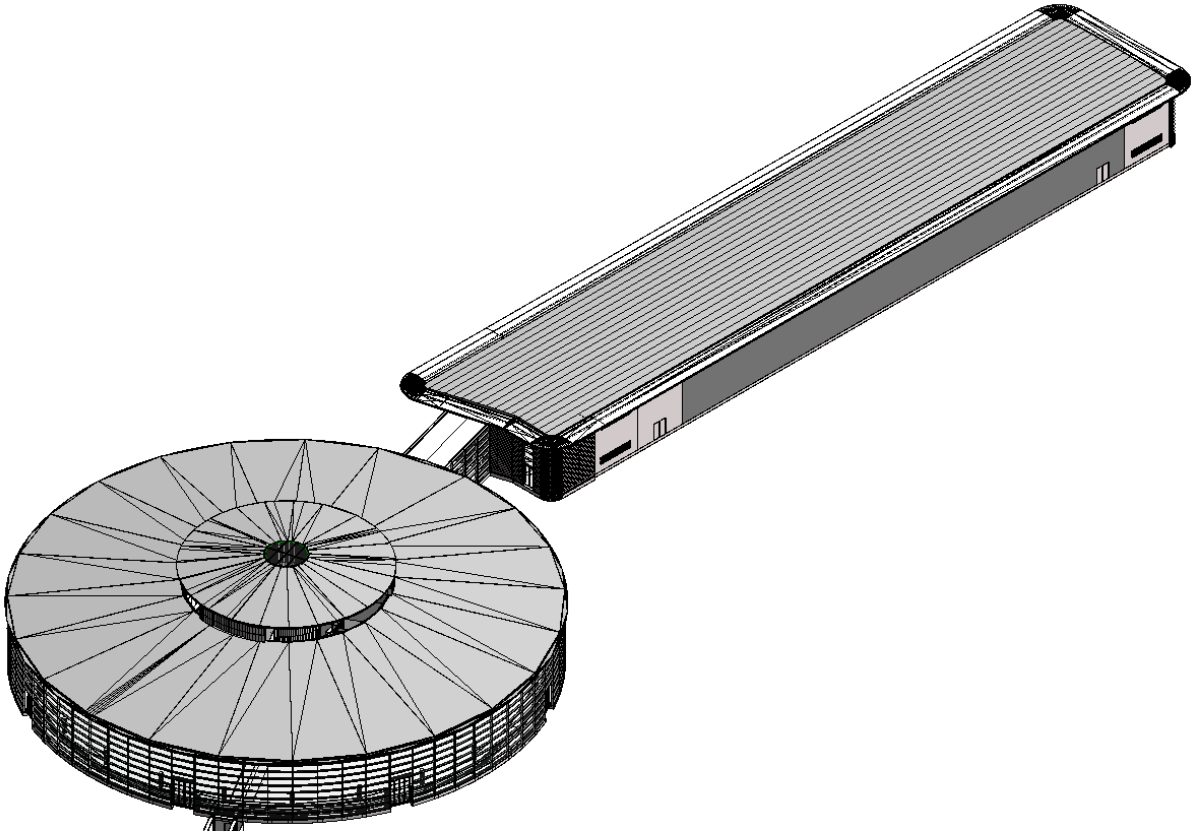


Figure 3.7. F2050 Autodesk Revit model

3.5.3. Location and Climate

F2050 is located between Sheffield and Rotherham in the United Kingdom as illustrated in Figure 3.8.

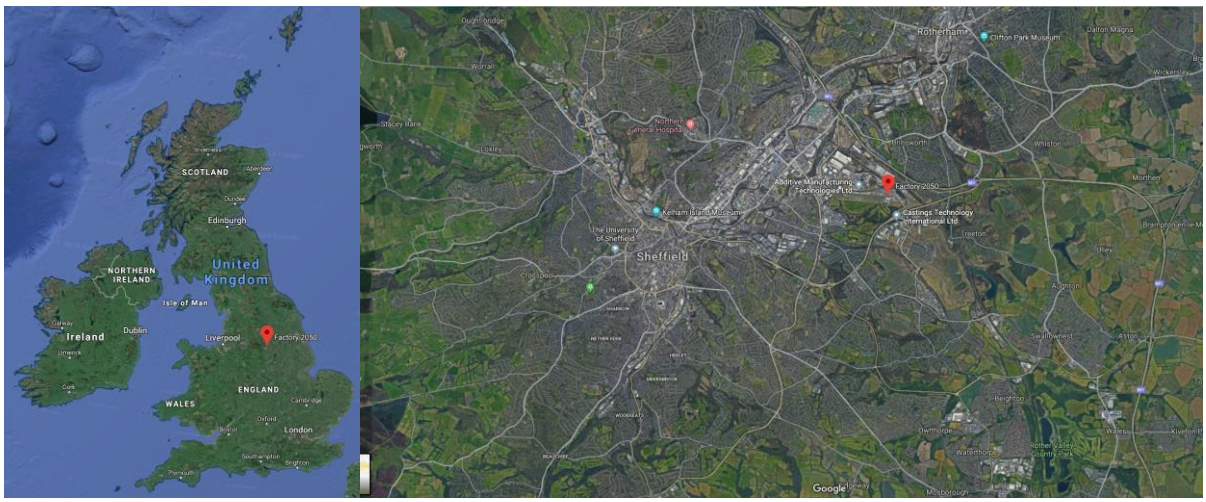


Figure 3.8. Geographical location of F2050

In conducting BEMS on F2050 knowledge of the location was important to define the appropriate environmental conditions that were applied during thermal simulations. BEMS has used two different types of weather data depending on the specific simulation application. The first is known as an Example Weather Year (EWY), or Design Summer Year (DSY), and the second is a Test Reference Year (TSR). Both types have been used to represent the environmental conditions of a building site location for conducting BEMS on a building. Both EWY and TRY datasets have been used to define weather input variables for a building simulation for each hour over a 12-month simulation window (i.e. 8,760h in total). The datasets were produced such that they represent the typical range of variation in weather variables that could reasonably be found over the course of several years of measurement. Such data has been used across the BEMS industry in an attempt to make BEMS a more deterministic tool by simplifying the stochastic nature of the many parameters that effect building simulation.

EWY data has been generated using a single and continuous 12-month timeframe in historical weather data that was considered as typical for the historic window under consideration. Typically, an EWY weather file has been used to conduct overheating assessments of buildings. TRY data has been generated by considering the weather for individual months in a historic time window to produce an amalgamation of typical months that would be combined to represent an example year. TRY datasets are used in long term energy analysis predictions and compliance assessments.

The standard installation of IES VE [16] contained a set of EWY for several locations around the UK. These were produced using the former method, as described above, in identifying a single continuous 12-month period that was considered to be representative over the full historic dataset that was used. Each EWY file was sourced from the MET office [139] and included the following information;

1. Dry-bulb temperature ($°C$)
2. Wet-bulb temperature ($°C$)
3. External dew-point temperature ($°C$)
4. Wind speed ($m\ s^{-1}$)
5. Wind direction (° clockwise from NORTH)

6. Direct radiation ($Wh\,m^{-2}$)
7. Diffuse radiation ($Wh\,m^{-2}$)
8. Global radiation ($Wh\,m^{-2}$)
9. Solar altitude (°)
10. Solar azimuth (°)
11. Cloud cover ($okta$)
12. Atmospheric pressure (Pa)
13. External relative humidity (%)

Each EWY was selected by considering historical data of up to 25years. Within this data set, any year that had a monthly mean value, of the weather parameters listed above, that differed more than 2 standard deviations from the long-term monthly mean were ignored. The next step was to discard, from the remaining years of the 25year window, those years that had the highest deviations from the long-term monthly mean until a final single year remained. This remaining year was then selected as the EWY for a particular site location.

This approach has been standard practice within the BEMS industry for conducting overheating assessments. The approach of conducting BEMS on F2050 deviated from this standard practice. The metered energy data for F2050 was obtained for the year of 2016 as previously discussed. As such, it was deemed more appropriate to obtain the weather data for the specific year of metered data; this had the advantage that any difference in weather data for 2016, when compared to the EWY, would be matched against a change in the heating or cooling demand of F2050.

A MET office [139] sourced weather file for Sheffield in 2016 was used for the F2050 case study as this matched with the time window of metered energy data obtained for the facility as described in subsection 3.5.2. The 2016 weather file differed from the EWY for Sheffield across all of the parameters previously described. Figure 3.9 was provided as an example of how a single weather file parameter; atmospheric pressure, differed between the EWY and the Sheffield 2016 weather file. The EWY file has recorded atmospheric pressure ranging from 96.6kPa to 104.1kPa with a mean value of 101kPa which is compared with the Sheffield 2016 weather file that recorded a pressure range of 95.4kPa to 102.4kPa with a mean pressure of 99.5kPa.

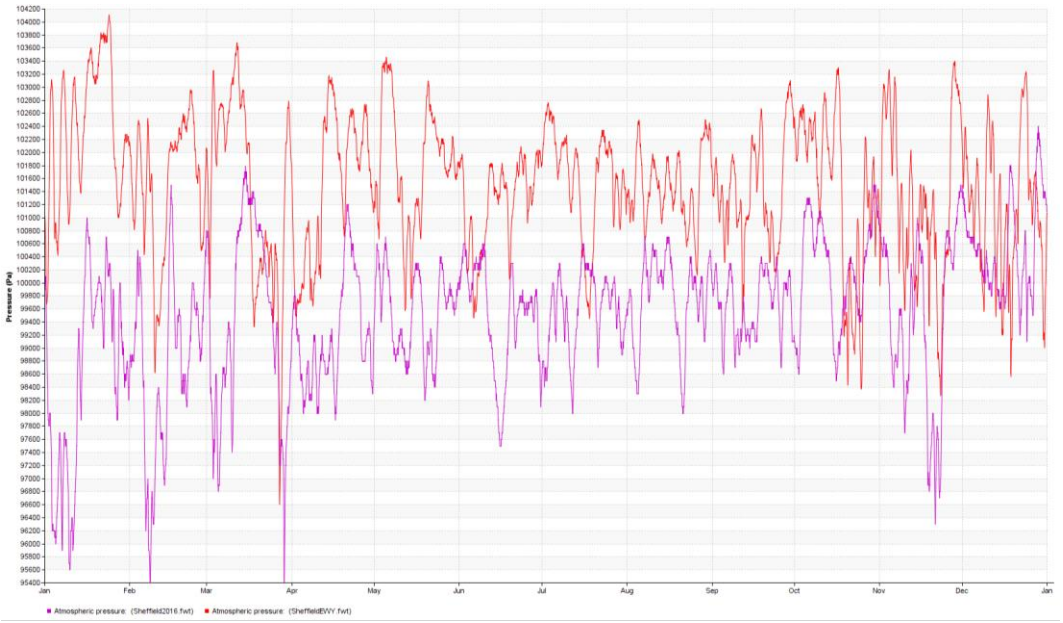


Figure 3.9. Difference of atmospheric pressure between Sheffield EWY and Sheffield 2016 weather files

The Dry-bulb and Wet-bulb temperature variations across 2016, in Sheffield, were illustrated in Figure 3.10 and Figure 3.11 respectively. The Dry-bulb temperature ranged from 27.1°C in July to -3.1°C in April with a mean annual temperature of 8.79°C. Similarly, the Wet-bulb temperature ranged from 22.7°C in July to -3.5°C in March with a mean annual temperature of 7.76°C.

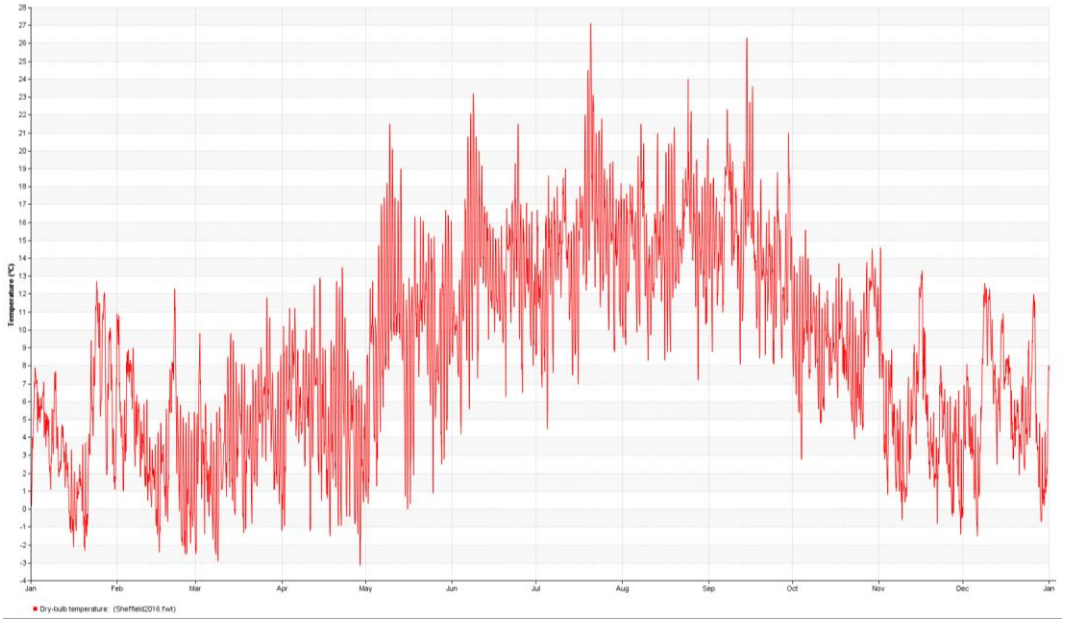


Figure 3.10. Dry-bulb temperature for Sheffield in 2016

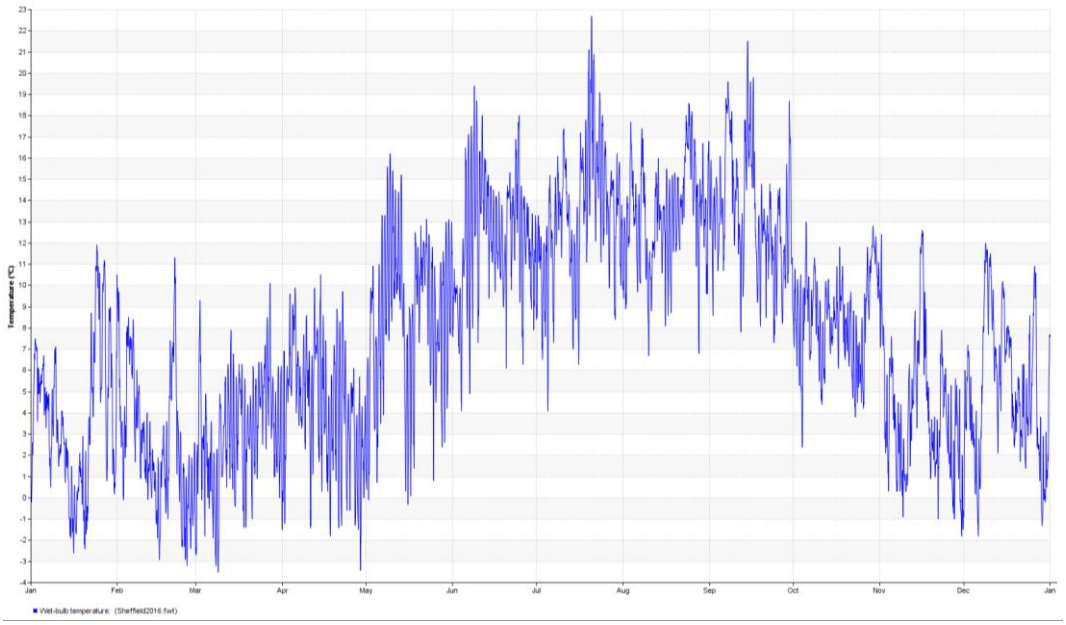


Figure 3.11. Wet-bulb temperature for Sheffield in 2016

A wind rose visualisation of wind speed frequencies, and the associated direction of travelling from, are presented in Figure 3.12. The data demonstrated a prevalent wind direction from South-West with the highest frequency wind speeds occurring between 3-5$m\ s^{-1}$. The mean wind speed over the entire weather file dataset for 2016 was calculated at 4.34$m\ s^{-1}$.

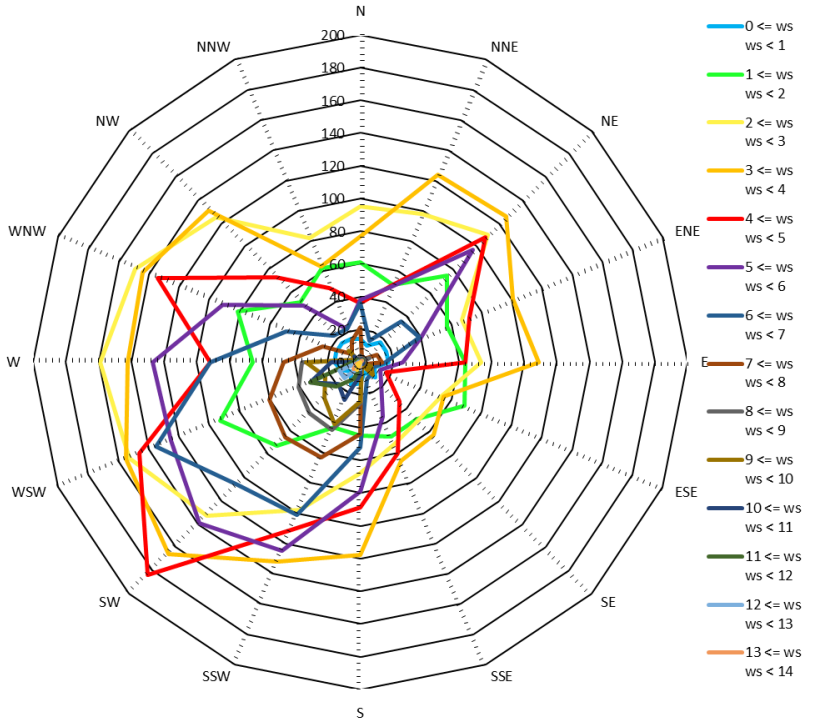


Figure 3.12. Wind Rose for Sheffield in 2016

As previously discussed a source of well documented uncertainty in BEMS is the use of weather data [49]. Only a limited number of EWY data files are provided for the whole of the UK and the MET Office has not obtained sufficient data for every city, town and village. This has meant that the weather files selected for use in BEMS may not actually have corresponded with the exact site location that was assessed. However, it has been common practice for weather data to be selected from as close a proximity to the assessed building to minimise this uncertainty. In the case of "Sheffield" weather data, the actual MET office monitoring station was located at Doncaster Airport, Finningley. This monitoring station was located 26.5km North-East of F2050, see Figure 3.13.

Figure 3.13. Distance between F2050 and Doncaster Airport

Although the recorded weather data did not correspond directly with the exact conditions at F2050 it was representative of the building's external climate. Unfortunately, in BEMS it is common to conduct assessments without prior site measurements of climatic data. This method has avoided projects taking over 12-months to complete by utilising historic data from close proximity recording stations. Weather data used in BEMS was found to represent a number of the BEMS parameters that had an associated uncertainty in their definition. It was important to correctly validate BEMS because of the presence of such uncertainty as discussed in subsection 3.3.

3.6. BEMS Geometry Creation

The Revit model of F2050 included a 3D model and floor plans that allowed measurements of the building geometry that were used to manually remodel the building's geometry for BEMS. The novel geometry of F2050, specifically its circular workshop, was too complex to model directly within IES VE [16] due to limitations with the native geometry modelling tools. To overcome these limitations Sketchup [97] was used to recreate the geometry of the F2050 circular workshop. The IES VE Sketchup plugin [140] then converted the completed Sketchup geometry into an appropriate format for importing into IES VE for BEMS.

The complex geometry of the F2050 circular workshop required the Sketchup [97] model to be split into multiple volumes (>425) that could be easily recognised by the IES VE Sketchup plugin [140] as illustrated in Figure 3.14. In simplistic terms the main room volumes were split up into smaller sub-volumes that represented cuboids and wedges with a maximum of six faces. This simplified the volume detection routine in Sketchup [97] to clearly identify each space within the model.

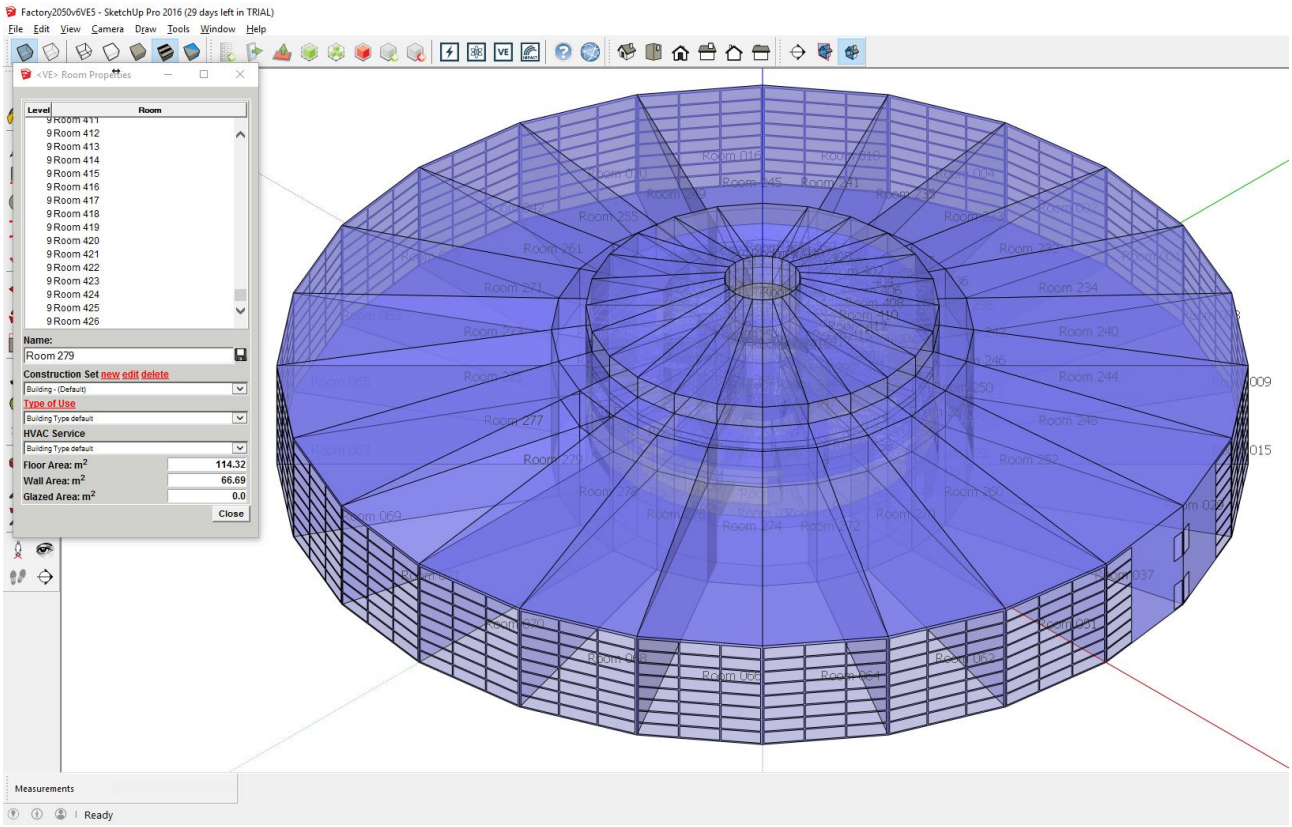


Figure 3.14. Sketchup geometry of F2050 for BEMS

Once all the individual volumes had been successfully identified by the Sketchup plugin [140] the geometry of the F2050 circular workshop was imported into IES VE [16] and combined with the simpler F2050 geometry. The large number of volumes created in the Sketchup model were then combined as appropriate to make a smaller number of large volumes that accurately represented the internal volume of F2050 and had the correct amount of glazing and doors. The final IES VE geometry has been presented in Figure 3.15.

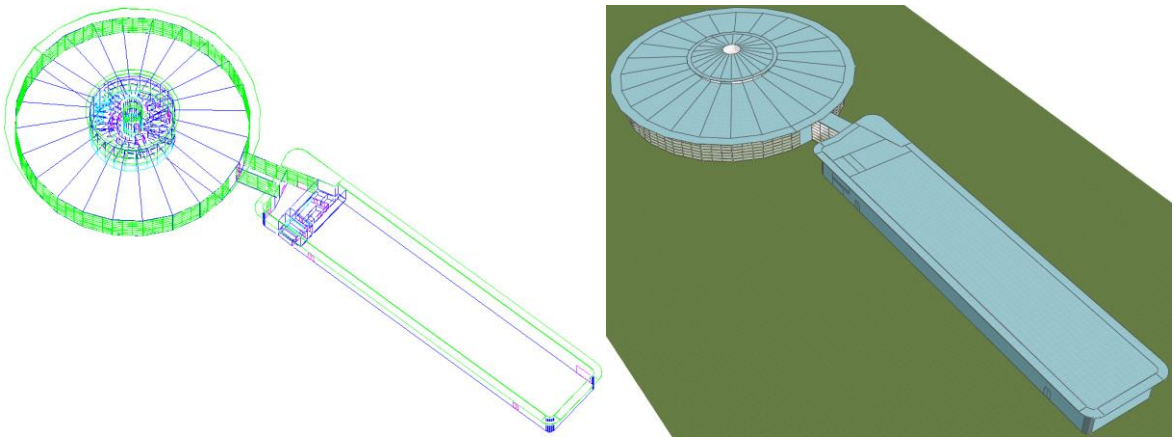


Figure 3.15. Cleaned benchmark F2050 geometry in IES VE

Having BIM data of F2050 available to take measurements from prevented the need of a site visit to establish the building geometry. However, the time spent generating building geometry was still experienced to be time consuming and inefficient. This F2050 geometry was recreated in 3 weeks (i.e. 120 hours) using this manual and traditional BEMS workflow.

3.7. BEMS Construction Materials

The next step in the BEMS workflow was to establish the construction materials used in F2050 so that $U\text{-}values$ could be assigned to each thermal boundary for use in thermal simulations. There was an inherent uncertainty associated with construction material $U\text{-}values$ as they have been found to change over the lifecycle of a building; usually through deterioration. As very little empirical information was available on $U\text{-}value$ variation and degradation, typical deterministic values were used as is standard practice across the BEMS industry.

Through a site survey, interrogation of the F2050 Revit model, conversations with the building operators and industrial guidance the construction elements, and their thermal

transmittance, used in each thermal boundary were assumed. These $U\text{-}values$ were tabulated in Table 3.1.

Table 3.1. F2050 BEMS construction materials

Parameters	Specification	U value ($W\ m^{-2}\ K^{-1}$)
External Walls	$5mm$ Lightweight Metallic Cladding – $70mm$ Expanded Polystyrene (EPS) Slab – $1mm$ Hardboard	0.453
Roof	$12.7mm$ Stone – $9.5mm$ Felt & Membrane – $325mm$ Insulation Board – $1.5mm$ Steel Siding – $12.7mm$ Cavity – $19.1mm$ Acoustic Tile	0.121
Ground Floor	$750mm$ London Clay – $250mm$ Brickwork – $100mm$ Cast Concrete – $25mm$ Dense EPS Slab – $25mm$ Chipboard – $10mm$ Synthetic Carpet	0.415
Window (External)	$6mm$ Glazing – $12mm$ Cavity – $6mm$ Glazing	2.86
Window (Internal)	$12mm$ Glazing	4.080
Window (Roof Light)	$8mm$ Polycarbonate – $12mm$ Cavity – $8mm$ Polycarbonate	3.5
Doors (External)	$6mm$ Glazing – $12mm$ Cavity – $6mm$ Glazing	2.86
Doors (Internal)	$6mm$ Plywood (Heavyweight) – $30mm$ Cavity – $6mm$ Plywood (Heavyweight)	2.288
Internal Partition	$12mm$ Plasterboard – $50mm$ Cavity – $12mm$ Plasterboard	1.892
Internal Ceiling/Floor	$20mm$ Chipboard – $50mm$ Cavity – $50mm$ SCREED – $100mm$ Reinforced Concrete – $50mm$ Cavity – $12.5mm$ Plasterboard	1.048

3.8. BEMS Boundary Conditions

In addition to specifying the construction material used it was important that appropriate boundary conditions were set for the building's HVAC system, occupation schedule and internal gains from equipment.

The HVAC system specification used in IES VE [16] was tabulated in Table 3.2. Based on discussions with the F2050 occupants the HVAC system was assumed operational 24 hours a day, 7 days a week, all year round with identical set points for heating and cooling. However, the actual temperature set-point used by the HVAC system was unknown and could not be determined on site due to limited plant room access. As such, a range of typically expected

set-points (19-25°C) were simulated at 1°C intervals to calibrate of the benchmark F2050 model.

Table 3.2. Assumed F2050 HVAC specification

Parameter	Set Values
Heating	Central Heating Convectors Heat pump (electric): ground or water source Electricity
Cooling	Air-conditioning Electricity
HVAC Settings	19-25°C
Hours of Operation	$24h$ (Weekdays) $24h$ (Weekends)

The energy gains due to occupancy and other fixed internal gains were incorporated into the thermal model on a schedule that was agreed with the building occupants. The schedule assumed full occupancy between 0800-1700h Monday to Friday. CIBSE Guide A [122] was used to inform typical internal gains for a building of similar type to F2050. The associated internal gains were tabulated in Table 3.3.

Table 3.3. Assumed F2050 internal gains

Internal Gain	Selected Values
Fluorescent Lighting	$8Wm^{-2}$
People	50 occupants @ $16m^2person^{-1}$ $74Wm^{-2}$ Sensible Gain $56Wm^{-2}$ Latent Gain
Misc. (Equipment)	$12Wm^{-2}$

The final boundary conditions that were required to define F2050 included the location and orientation of the building relative to North which was determined to be approximately 5°, see Figure 3.16.

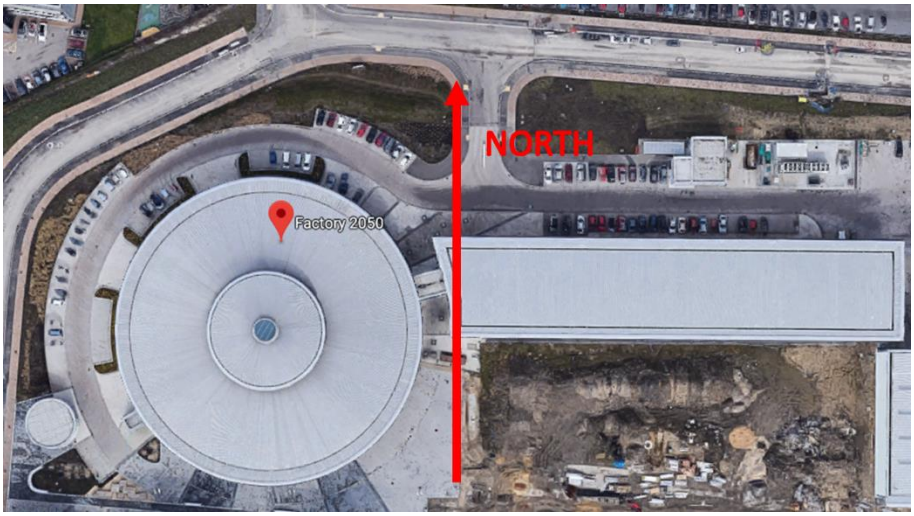


Figure 3.16. Orientation of F2050

Combined with a MET office [139] sourced weather file for Sheffield in 2016, an accurate simulation of solar gains and local environmental conditions was conducted. A view of the extreme sun paths, relative to F2050 in 2016 were illustrated in Figure 3.17.

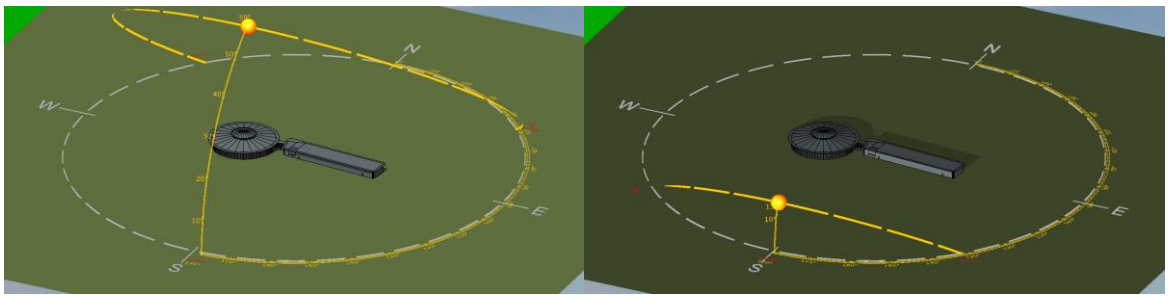


Figure 3.17. (left) 21st June 2016 @ 1200h, (right) 21st December 2016 @1200h

3.9. Air Infiltration

As described in subsection 3.3 the rate of air infiltration used in BEMS is a parameter that has been identified to dramatically alter the results of predicted energy use in a building. When a higher infiltration rate is used more energy is put into the thermal air volume of a building to maintain the defined HVAC set-point temperatures. Tests have been, on occasion, conducted following the construction of a new building to determine if air infiltration rates match those specified in the design specification.

These tests are known as "*Blower Door Tests*" and are documented to involve blocking a single doorway with a panel that includes an integrated fan. The fan is used to draw air out of a building to create a negative pressure inside the building. A differential pressure across the building fabric with the external atmospheric pressure is then generated and air is

forced into the building through any air leakage flow paths. When the pressures on either side of the blower door and air flow rates through the door are measured the effective air tightness of a building is established.

For small residential building types tests have been conducted in approximately $1h$ [141]. However, due to the size, geometrical complexity and occupant use of F2050 it was deemed impractical to employ a blower door test in this research. One disadvantage to these tests was highlighted in that measurements are only obtained that are valid for the exact time of the test. Air infiltration has been identified as highly dependent on pressure and temperature differences across the fabric of a building. As such, a blower test couldn't be used measure long term infiltration rates. An alternative to the blower door test is to perform hand calculations to approximate the value of air infiltration of a building.

3.9.1. Simplified Air Infiltration Calculation

As F2050 was a recent construction it was expected to comply with the 2010 Part L Building Regulations [142] on air permeability, q_{50} $(m^3\ h^{-1}\ m^{-2})$, such that air infiltration can be assumed as a maximum of $10m^3\ h^{-1}\ m^{-2}$ at $50Pa$ as stated in CIBSE Guide A [122].

By taking the thermal volume and surface areas of the F2050 IES VE model of approximately $53{,}000m^3$ and $17{,}700m^2$ respectively the air changes per hour at $50Pa$, ach_{50} (h^{-1}), for F2050 were calculated using Equation 3.23 where A_{therm} (m^2) and V_{therm} (m^3) are the boundary surface area and volume of the thermal envelope respectively:

$$ach_{50} = q_{50} \times \frac{A_{therm}}{V_{therm}}$$

Equation 3.23. ach_{50} calculation

Accordingly, ach_{50} has been determined as $3.34\,ach$. It has been accepted within the literature, through empirical studies, that ach_{50}can be converted to ach at normal working pressure using Equation 3.24 [143–145]:

$$ach = \frac{ach_{50}}{20}$$

Equation 3.24. Conversion of ach_{50} to ach at normal working pressure

For F2050 this was calculated as $0.167ach$. It is worth noting that the ach defined in IES VE represented the maximum infiltration rate over the full simulation window but that the ach

at any given time will vary based on temperature and pressure gradients across the building fabric.

3.9.2. Detailed Air Infiltration Calculation

A more detailed calculation method of the air infiltration rate of F2050 was conducted for comparison with the simplified calculation method as outlined by Liddament [145]. This detailed method developed a zonal model of F2050 in which the building was considered as a single zone which was considered an acceptable assumption for open plan buildings such as F2050. Liddament also noted that "*if resources are limited or if insufficient data are available, then a single zone approximation may prove the only viable route* for air infiltration calculations".

Zonal methods are used to calculate air flow rates through each potential opening in the envelope of building. Thus, a zonal calculation method offers an approximation of ventilation and air infiltration that is close to reality and has enabled an approximate calculation of ach for F2050. The geometry of the F2050 building was considered such that every potential air leak flow path was identified; An example is provided in Figure 3.18; In total 228 infiltration flow paths were identified. Liddament [145] emphasised the importance of how calculated air infiltration results risk quickly deviating from reality if any potential leakage flow paths were omitted from the calculation.

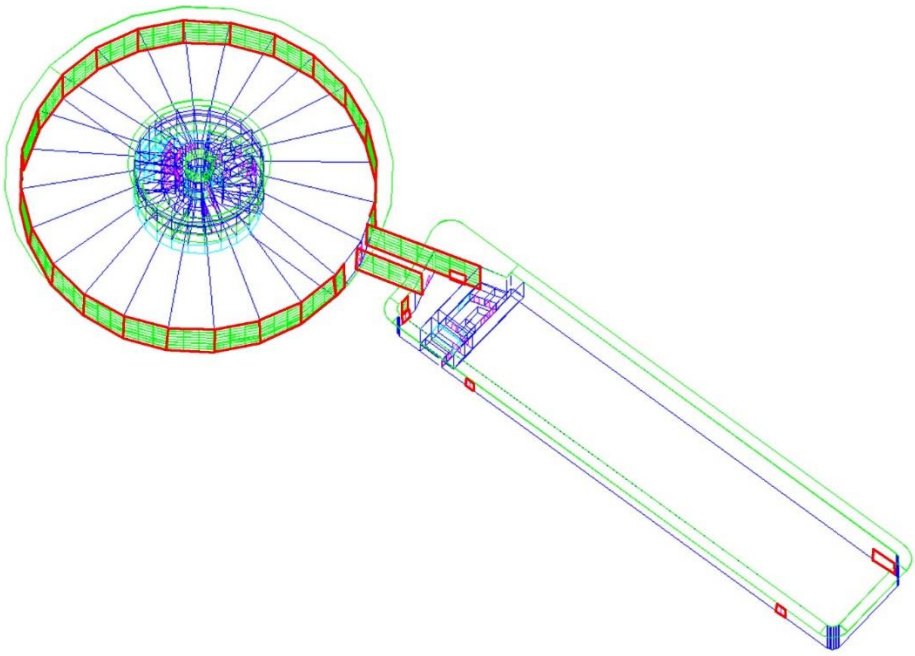


Figure 3.18. Identification of air infiltration flow path through the F2050 window frames and doors

Each flow path was defined using three parameters; 1) a flow coefficient C_f, 2) a flow exponent n_f, and 3) either the length or surface area of a potential infiltration flow path. Published data was available for the coefficients C_f and n_f for a range of different leak path types based on empirical tests [145]. A typical example of the published data has been tabulated in Table 3.4. One observation of this data was that the sample sizes provided for empirical tests were small which raised the question of reliability in the published quartile range and median values. However, as values were empirically derived they were used cautiously in the detailed air infiltration calculation to establish a range of potential air infiltration rates.

Table 3.4. Leakage characteristics - wall/window and wall/door frame [145]

Data expressed for each metre length of joint	**Lower Quartile**		**Median**		**Upper Quartile**		**Sample Size**
	C_f	n_f	C_f	n_f	C_f	n_f	
Caulked joint – Laboratory and field tests	3.3×10^{-7}	0.6	2.5×10^{-6}	0.6	1.2×10^{-5}	0.6	7
Uncaulked joint – Laboratory and field tests	5.3×10^{-5}	0.6	6.1×10^{-5}	0.6	6.7×10^{-5}	0.6	5

Once defined with respect to the flow coefficient and flow exponent values, each flow path was further defined by its height above ground level and a wind pressure coefficient c_{wp}. For vertical flow paths the height above ground was taken as its midpoint for simplification. Wind pressure coefficients were defined using published data that took into account the shape of the building segments, direction of prevailing wind (assumed from South-West as per Figure 3.12) and the exposure conditions for F2050.

Liddament [145] published an Air Infiltration Development Algorithm that was written in the computer language BASIC. For this research the algorithm was rewritten in C++ and

modified to accept a large amount of data instead of manually providing the input of each individual flow path to minimise input error, see Appendix C.

The algorithm performed an iterative calculation on the volumetric flow balance of a building under specified environmental conditions as per Equation 3.25 where $\dot{V}$ $(m^3\ s^{-1})$ was the volume flow rate through infiltration path i and j was the total number of infiltration flow paths. The volumetric flow was calculated by consideration of the pressure differences across each infiltration flow path. The volume flow rate $\dot{V}$ through each infiltration path i was defined by Equation 3.26 where P_{ext} (Pa) was the external pressure of the flow path i and P_{int} (Pa) was the internal pressure of the building.

$$\sum_{i=1}^{j} \dot{V} = 0$$

Equation 3.25. Volumetric flow balance of building zone

$$\dot{V} = C_{f,i}\left(P_{ext,i} - P_{int}\right)^{n_i} \quad \text{if} \quad P_{ext} > P_{int}$$

$$\dot{V} = -C_{f,i}\left(P_{int} - P_{ext,i}\right)^{n_i} \quad \text{if} \quad P_{int} > P_{ext,i}$$

Equation 3.26. Volumetric flow through an infiltration flow path

The driving pressure at each infiltration flow path was calculated based on the assumption that it was caused by a combination of stack pressures P_s and wind pressures P_w. Stack pressures were calculated by considering the temperature differentials between the internal building volume and the external air. This differential was assumed to alter the air density and created pressure gradients across each infiltration path which was also dependent on the height of the flow path above ground level. This was calculated using Equation 3.27 where ρ_0 is the air density at $273K$ (i.e. $1.29kg\ m^{-3}$), g was the acceleration due to gravity (i.e. $9.81m\ s^{-1}$), T_0 (K) was the temperature at $0°C$, z_i was the height of infiltration flow path i above ground level, $T_{ext,i}$ (K) was the external air temperature of infiltration flow path i and $T_{int,i}$ (K) was the internal air temperature of infiltration flow path i.

$$P_{s,i} = \rho_0\ g\ T_0\ z_i \left(\frac{1}{T_{ext,i}} - \frac{1}{T_{int,i}}\right)$$

Equation 3.27. Stack pressure across an infiltration flow path

When a surface has been exposed to wind in a perpendicular direction a positive pressure has been observed on the wind side. When wind is observed to flow around a building flow separation has been found to occur that has created a variation in pressure on all other surfaces that were either a positive or negative pressure. The wind pressure was calculated for each infiltration flow path using Equation 3.28 where $\rho\ (kg\ m^{-3})$ was the density of air, c_{wp} was the wind pressure coefficient of the infiltration flow path i and $v_{F2050}\ (m\ s^{-1})$ was the wind velocity at building height on site.

$$P_{w,i} = \frac{\rho\ c_{wp,i}\ {v_{F2050}}^2}{2}$$

Equation 3.28. Stack pressure across an infiltration flow path

As previously stated, the mean wind speed from the 2016 Sheffield weather dataset was calculated as $4.34 m\ s^{-1}$. It is standard practice for these wind speeds to be measured at a remote weather station and as such adjustments were required to account for the height of F2050 and local site topography. BS EN 5925 [146] provided a method for this correction which is reproduced in Equation 3.29 where $v_{F2050}\ (m\ s^{-1})$ was the corrected site wind velocity at building height, $v\ (m\ s^{-1})$ was the wind speed measured in open country at a reference height of $10m$, $z_{F2050}(m)$ was the height of F2050 (i.e. $15.5m$) and, k_t and a_t were terrain coefficients which were selected as 0.52 and 0.20 respectively. These values represented coefficients for topography described as "*country with scattered wind breaks*" which matched that of F2050 in 2016. A corrected site wind velocity v_{F2050} at the F2050 building height was calculated as $3.904 m\ s^{-1}$ which was provided as an input into the detailed infiltration calculations.

$$v_{F2050} = v\ k_t\ {z_{F2050}}^{a_t}$$

Equation 3.29. Site wind velocity correction

With the stack and wind pressure defined the total pressure $P\ (Pa)$ across each infiltration flow path i was then calculated using Equation 3.30.

$$P_i = P_{s,i} + P_{w,i}$$

Equation 3.30. Total pressure across an infiltration flow path

The final inputs into the detailed infiltration calculation were internal and external zone temperatures. The external zone temperature used the mean dry-bulb temperature from

the 2016 weather file for Sheffield which was previously stated as 8.79°C. The calculation was performed against the range of HVAC settings tabulated in Table 3.2 to define the internal zone temperature.

With all inputs defined the calculation was performed iteratively until a flow balance in the F2050 internal zone was achieved. As per guidance from Liddament [145], the calculation initially assumed a negative internal pressure of -100 Pa as a starting condition to aid convergence of a solution. Each iteration adjusted the internal pressure value until a flow balance was achieved for the F2050 internal zone within a tolerance of $1 \times 10^{-4} m^3\ s^{-1}$. During the calculation the infiltration rate $\dot{V}\ (m^3\ s^{-1})$ was converted to air changes per hour $ach\ (h^{-1})$ as per Equation 3.31 where $V\ (m^3)$ was the building volume used in calculations. This calculation method produced the results as tabulated in Table 3.5.

$$ach = \frac{\dot{V}\ \times 3600}{V}$$

Equation 3.31. Conversion of air infiltration rate to ach

Table 3.5. Results of detailed air infiltration calculation

Temperature (°C)	Variable	Lower Quartile	Median	Upper Quartile
19	Air Infiltration Rate ($m^3\ s^{-1}$)	1.608	2.125	3.326
	Air Changes per Hour (h^{-1})	0.109	0.144	0.226
20	Air Infiltration Rate ($m^3\ s^{-1}$)	1.753	2.315	3.613
	Air Changes per Hour (h^{-1})	0.119	0.157	0.245
21	Air Infiltration Rate ($m^3\ s^{-1}$)	1.888	2.491	3.877
	Air Changes per Hour (h^{-1})	0.128	0.169	0.263
22	Air Infiltration Rate ($m^3\ s^{-1}$)	2.016	2.655	4.123
	Air Changes per Hour (h^{-1})	0.137	0.180	0.280
23	Air Infiltration Rate ($m^3\ s^{-1}$)	2.138	2.813	4.357
	Air Changes per Hour (h^{-1})	0.145	0.191	0.296

Temperature ($°C$)	Variable	Lower Quartile	Median	Upper Quartile
24	Air Infiltration Rate ($m^3\ s^{-1}$)	2.254	2.963	4.579
	Air Changes per Hour (h^{-1})	0.153	0.201	0.311
25	Air Infiltration Rate ($m^3\ s^{-1}$)	2.367	3.107	4.792
	Air Changes per Hour (h^{-1})	0.161	0.211	0.325

3.9.3. Air Infiltration Approach for Case Study

As a reminder the maximum ach calculated for F2050 using the simplified method was $0.167ach$. This demonstrated a suitable comparison with the ranges calculated at varying temperatures using a more detailed approach as $0.167ach$ fell within the upper and lower quartile bounds between 19-25$°C$. At 21$°C$ $0.167ach$ was found to be within approximately 1% of the median value calculated from the detailed method. While this was a positive finding, limitations to the detailed method have been observed.

As previously discussed, the empirical data used to define flow characteristics of infiltration flow paths was based on small sample sizes so while lower quartile, upper quartile and median values have been published the extent to which they can be relied upon to represent flow paths across the entire built environment is unknown. The calculation relied on the use of deterministic inputs for internal and external temperature as well as site wind speeds. While mean values were used to represent the whole data set the actual results that have been obtained could only be considered valid for those specific environmental criteria at a single point in time. This made it difficult to extrapolate the results across a full weather year used in BEMS. Finally, when providing boundary conditions into BEMS, infiltration was defined as a limiting maximum value and the results obtained via a more detailed method cannot be used to determine this. Once again, the small sample sizes have made extrapolation difficult outside of the quartile ranges that were measured.

For these reasons a single maximum value of $0.167ach$, derived from the simplified method of calculating air infiltration, was used as the accepted boundary condition for conducting BEMS on F2050. Although the simplified method may be more generalised fewer questions are raised over the reliability of calculation inputs. This aided the development of a manually

created baseline BEMS against metered energy data that was calibrated to establish the HVAC temperature setting.

3.10. F2050 Benchmark Validation

BEMS was conducted on the manually recreated F2050 geometry with the boundary conditions, that were described earlier in this Chapter, applied. Due to the uncertainty around the HVAC set point used in F2050 a range of HVAC set point temperatures were run as separate simulations for 19-25 $°C$ at one-degree increments. The results of each simulation are tabulated in Table 3.6 in which the calculated NMBE and CVRMSE, as outlined in subsection 3.3, were also provided.

Table 3.6. BEMS results across varying HVAC set point temperatures

Month	F2050 Data (MWh)	Predicted Energy Use at differing HVAC Set Points						
		$19°C$ (MWh)	$20°C$ (MWh)	$21°C$ (MWh)	$22°C$ (MWh)	$23°C$ (MWh)	$24°C$ (MWh)	$25°C$ (MWh)
Jan.	112.9	107.5	110.3	113.25	116.3	119.4	112.6	125.8
Feb.	113.7	96.2	98.8	101.4	104.1	106.8	109.7	112.6
Mar.	107.5	103.4	105.9	108.5	111.2	114.0	117.0	120.1
Apr.	105.6	99.5	101.4	103.5	105.7	108.1	110.7	113.4
May	95.5	94.3	95.5	96.9	98.4	100.1	101.9	103.9
June	93.9	86.3	87.0	88.0	89.1	90.4	91.8	93.3
July	89.5	92.7	93.0	93.6	94.2	95.1	96.0	97.2
Aug.	95.6	85.3	85.9	86.7	87.6	88.7	90.0	91.5
Sep.	92.2	87.5	88.2	89.1	90.1	91.2	92.5	93.9
Oct.	120.6	96.9	98.3	100.0	101.9	104.0	106.3	108.8
Nov.	111.0	96.5	99.0	101.8	104.5	107.4	110.4	113.5
Dec.	95.0	103.8	106.3	109.0	111.9	114.8	117.7	120.8
NMBE (%)		7.35	5.59	**3.66**	**1.59**	**0.62**	**2.97**	5.45
CVRMSE (%)		**11.12**	**10.10**	**9.30**	**8.87**	**8.94**	**9.59**	**10.78**

THE NMBE and CVRMSE maximum criterion, as defined by ASHRAE Guideline 14 [51] were 5% and 15% respectively when monthly measured data was used as it was in the F2050 case study. Every HVAC set-point temperature achieved a satisfactory CVRMSE value below the 15% threshold but a NMBE value below 5% was not always achieved. Based on these limits the F2050 BEMS results indicated a validated model for $21°C$, $22°C$, $23°C$ and $24°C$ HVAC set point temperatures. These results were a good example of how a validated model was not a necessarily a unique solution in BEMS.

The BEMS results data was reviewed to select one of these HVAC set point temperatures that was chosen to represent a benchmark BEMS for use in further research. The monthly discrepancies present in the simulations ranging from 21-24 $°C$ all had slightly larger discrepancies in October and December. When the ASHRAE Guideline 14 [51] criteria was considered the lowest calculated indices occurred at $23°C$ with 0.62% NMBE and at $22°C$ with a 8.87% CVRMSE. A review of the individual months at $23°C$ temperature identified that the results for December had a discrepancy in excess of 20%; This was also observed at a HVAC set point temperature of $24°C$.

When the individual months for both $21°C$ and $22°C$ simulations were investigated all months were found to be within 20% of measured data. When compared against individual monthly discrepancies of 15% the results from $21°C$ only had a single month in excess of this value (i.e. October). In comparison the results at $22°C$ produced two months with a discrepancy against measured data in excess of 15% (i.e. October and December). While ASHRAE Guideline 14 [51] did not provide limits to discrepancies against individual months in a data set it was worthy of consideration in an attempt to minimise the effect of any anomalous data that departs from the trend line.

A review of CIBSE Guide A [122], in light of these results, suggested that suitable temperatures needed to achieve thermal comfort in a factory where sedentary work took place, such as the F2050 environment, were 19-$21°C$ in winter months and 21-$25°C$ in summer months. As the building was a recently new construction in 2016 it is likely the HVAC set point was set to a single value for commissioning and not changed. The fact that the HVAC system was also operational for $24h$, instead of only during specified occupied hours as identified by occupants, also provided credence to this assumption. Based on this information, and the results of BEMS on F2050 over several different HVAC temperatures, the model selected as the benchmark model in this research used $21°C$ as the HVAC set point temperature.

The comparison of the benchmark simulated energy use against the metered energy use was illustrated in Figure 3.19. Over the full simulation window of 12-months the simulated and metered overall energy use differed by 3.4% with the simulation underestimating the metered data.

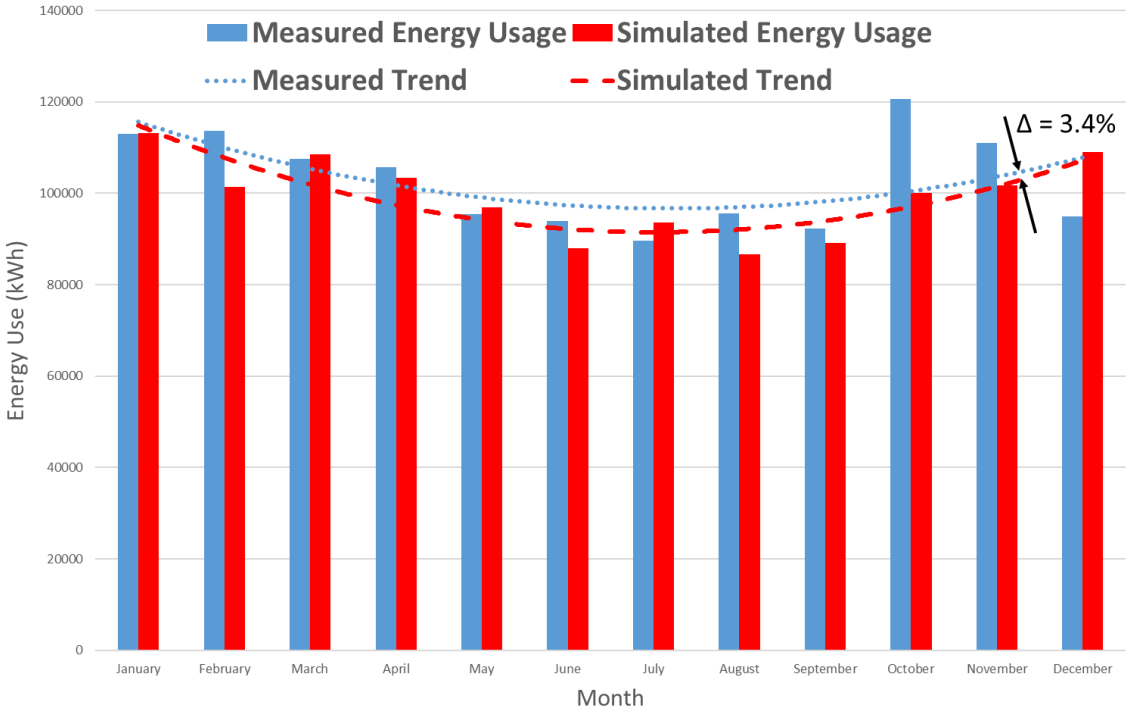


Figure 3.19. Validated benchmark BEMS results for F2050

The trend lines in Figure 3.19 demonstrated good agreement between the metered and benchmark simulated data. The indices specified by ASHRAE Guideline 14-2002 [51]; NMBE and CVRMSE were calculated as 3.66% and 9.30% respectively, as per Table 3.6 and as such the benchmark BEMS for F2050 was considered validated.

Closer inspection of Figure 3.19 highlighted that the simulation results match the metered data closely for most months but that a larger discrepancy was observed in October and December. Comparison with adjacent months in the measured dataset suggested that a larger amount of energy was used in October compared with September and that the difference appeared more pronounced than between other adjacent months. A similar observation was made with the December energy measurements which were below the simulation results. For this reason, it was likely that the measurement was anomalous for these months. Results like these are usually caused if the building was operated in an unexpected manner (e.g. low staffing levels, intensive machinery use etc.) which highlighted how the uncertainty of occupant behaviour on results is difficult to account for in BEMS.

These discrepancies also highlighted the importance of considering building performance over a longer time frame. As per industry guidance at least 12-months was used for BEMS as

single months can exhibit abnormal behaviour that is only obvious on consideration of the cyclical trend of energy use. Higher resolution would have increased confidence in this validation, but unfortunately metered energy data was only available for 2016 and at monthly resolutions.

Achieving a validated benchmark BEMS for F2050 provided confidence in the use of the applied boundary conditions such as HVAC settings, infiltration rate and internal gains. These same boundary conditions were then used in further research on automated geometry reconstruction for comparison with metered data. While unknown levels of uncertainty still existed in the boundary conditions of F2050 the use of boundary conditions that were applied to a validated benchmark ensured a consistency in simulation assumptions and inputs.

3.11. Summary

This Chapter provided an overview of the primary energy transportation mechanisms within the built environment. From these basic concepts the Degree Day method and calculation of the Total Loss Coefficient, were outlined. The sources of uncertainty when conducting BEMS were presented with the main categories defined as 1) weather data, 2) building envelope, 3) HVAC System and 4) occupant behaviour. The industry accepted methods of BEMS validation were described, including the defined limits to validation indices, as well as the requirements for performing compliant BEMS.

The BEMS software used in this research was identified as IES VE [16] which was found to outperform other BEMS software in the literature. Following the introduction of a case study building, F2050, and the available information on its location and climate the method of manual BEMS model geometry creation was documented. A large area of uncertainty in BEMS was identified as the definition of the infiltration rate of a building's envelope. Two methods of calculating this input parameter were described as a simple and a detailed method. The value of $0.167ach$ was used in the production of a benchmark BEMS of F2050.

A range of BEMS were conducted on F2050 for 2016 using a different internal temperature in each (i.e. 19-25°C). The $21°C$ BEMS achieved validated results of 3.66% and 9.30% for NMBE and CVRMSE respectively and was adopted as the benchmark model for future research. The total predicted annual energy use differed by 3.4% compared to metered data.

Having produced a benchmark BEMS for F2050, with validated boundary conditions, the research was progressed into automated reconstruction of the geometry of F2050 to improve the BEMS workflow. As noted the manual benchmark model took 120 hours to produce by replicating BIM data in IES VE [16]. The next stage of research conducted digital site surveys and developed a method of automatically processing the data to recreate the geometry for BEMS applications.

Chapter 4.

Automated Geometry Reconstruction

4.1. Introduction

The research conducted into the development of a tool for automated reconstruction of building geometry in a BEMS format is described in this Chapter. The data collection method and resulting Point Clouds are presented in subsection 4.2. The Point Cloud registration method, existing modelling tools, and useful data storage formats for Point Clouds are documented in subsection 4.3.

An overview of the most prominent tool within academic literature was included in subsection 4.4 which also demonstrated preliminary testing with the site survey data. The method involved in further developing the automated building geometry reconstruction is described in subsection 4.5. The results of automated reconstructions on F2050 data is documented in subsection 4.6 which includes a description of noise reduction methods to improve the reconstruction outcomes.

A summary of the research conducted on Automated Geometry Reconstruction in this Chapter is provided in subsection 4.7. It includes a discussion of key findings from the Chapter as well as limitations associated with the method that was carried out.

4.2. Building Survey

To develop an automated building geometry reconstruction workflow, the first step was to conduct a building survey that generated a digital dataset using LiDAR. In LiDAR a beam of light is emitted from a transmitter and the time taken for the beam to be reflected and return to the LiDAR unit is measured. The constant speed of light, when passing through a homogenous medium, has enabled this process to be highly accurate and allowed for a direct measurement of the distance that the targeted surface is positioned away from the sensor. The first uses of LiDAR were outlined by Goyer and Watson [147] for meteorology applications and probing the atmosphere.

Common configurations of LiDAR units have included a static sensor facing in a single orientation or the use of a rotating mirror, which the laser has been directed onto, in addition to a rotating base that has allowed measurements to be taken at regular intervals within a spherical view.

Some of the most well-known commercial entities that have supplied LiDAR units to survey, measurement and construction industries include Leica Geosystems [148] and Faro [149]. Typically, LiDAR systems that have been used in building surveys have been static terrestrial scanners. They are usually mounted on tripods and tend to be bulky, heavy and expensive to purchase.

There are two primary methods of conducting static terrestrial LiDAR surveys. In each, the LiDAR scanner is positioned throughout a survey site and used to scan from several discrete locations. The first method uses LiDAR scan targets, see Figure 4.1, throughout the facility that can be recognised by post-processing software to stich all the separate scans together in a common coordinate system. When a facility has been surveyed that exhibits complex geometry or it has not been practical to use targets a second method is used in which the individual scans are manually registered together in a common coordinate system – this has been identified as labour intensive especially when a large survey data set has been collected. The survey of F2050 was undertaken using a Leica Scanstation P20 [150] terrestrial scanner as shown in Figure 4.2.

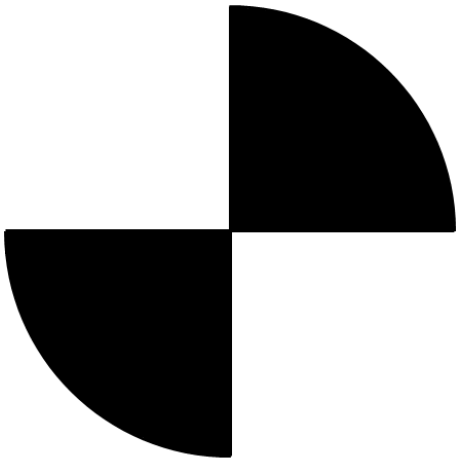


Figure 4.1. Typical LiDAR survey target

Figure 4.2. Leica Scanstation P20 [150]

The scanner was mounted onto a tripod and several scans were performed from discrete locations around the F2050 facility. In total 85 scans were initially conducted around the facility and the location of each scan was recorded which has been provided in Appendix D.

Each laser scan measurement was recorded as a 3D Cartesian coordinate relative to the origin point of the scanner. On completion of a single scan the data was visualised as a Point Cloud, see Figure 4.3.

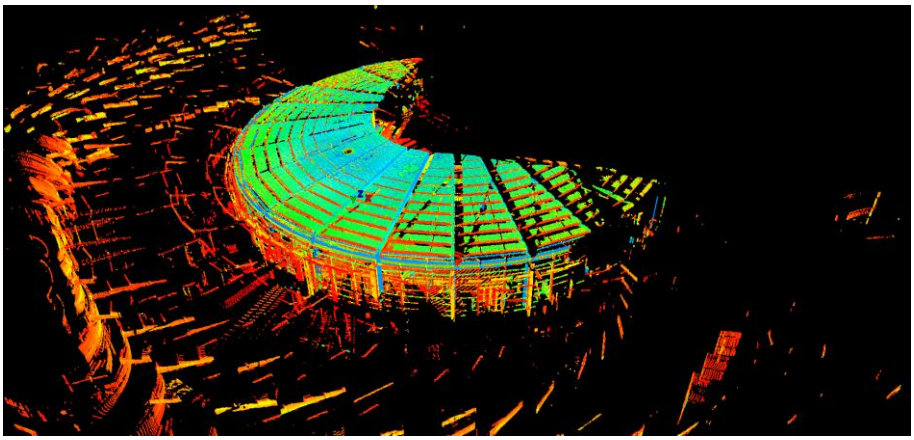


Figure 4.3. Example of single scan from F2050 in Point Cloud format

Evident in Figure 4.3 was the varying colours displayed in the Point Cloud. In addition to measuring the distance to the surrounding environment from the scanner the intensity of the received reflection during measurements is also recorded. Blue and green data points represented high intensity measurements that were considered more accurate than the red low intensity measurements. The red data points tended to be sparsely recorded too as they are based on the scanner having received a weaker signal from reflecting off these objects.

Opaque objects like roofs, walls and floors tended to be measured with high intensity results compared to transparent objects, such as windows, that created reflections and 'ghosted' objects in scan results that didn't actually exist. In buildings with high levels of glazing, such as F2050, ghosted glazing is a substantial cause of survey noise. Close fitting blinds were eventually used to reduce this errant noise, but this increased the risk of obscuring important details from scans, see Figure 4.4.

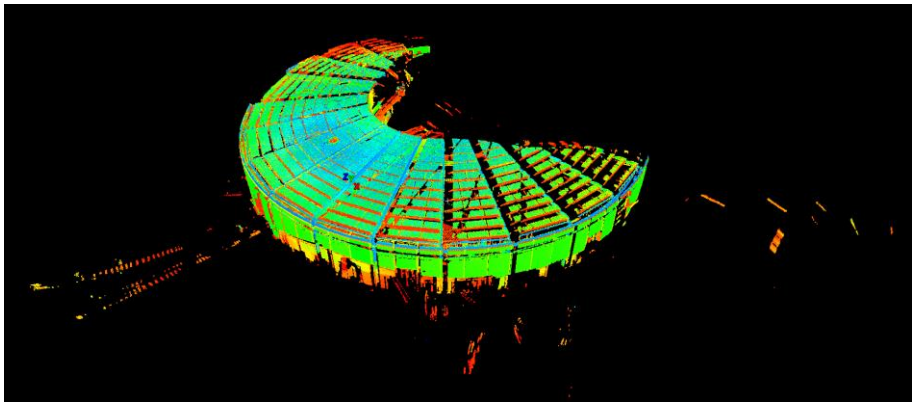


Figure 4.4. Example F2050 scan with blinds down

For each scan a laser was directed onto a central prism of the scanner which was rotated at high speeds to measure the distances of surfaces around the scanner in a vertical plane with a 270° field of view. The base of the scanner also rotated through 360° during this process. This resulted in a spherical measurement of surfaces around the laser scanner except for a blind spot directly above and below the scanner as demonstrated in Figure 4.5 as black circles.

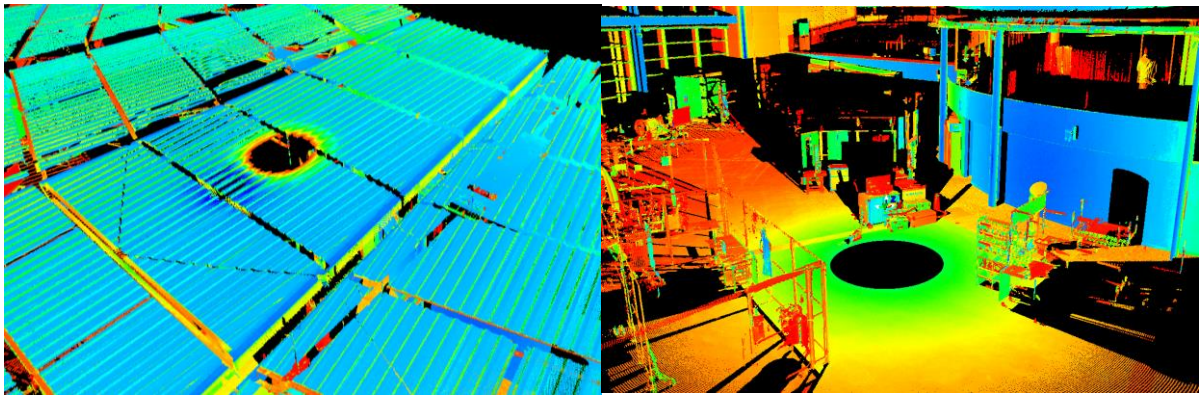


Figure 4.5. (left) Blind spot above scanner, (right) blind spot below scanner

Across all the scans of F2050 the capture resolution between each distance measurement was 12.5mm at a 10m distance from the scanner. The scanner was capable of a much lower resolution of 0.8mm spacing between measurements however this was only possible with an increased time per scan which was decided against. The balance between scan resolution and survey time needed to be carefully considered and was dependent on the object or structure being scanned and the potential disruption to other activities on the scan site during the survey.

On completion of the distance measurements the scanner took a series of overlapping photos from each scan location. An example of one of these photos in which the ceiling of the circular F2050 workshop is visible has been provided in Figure 4.6.

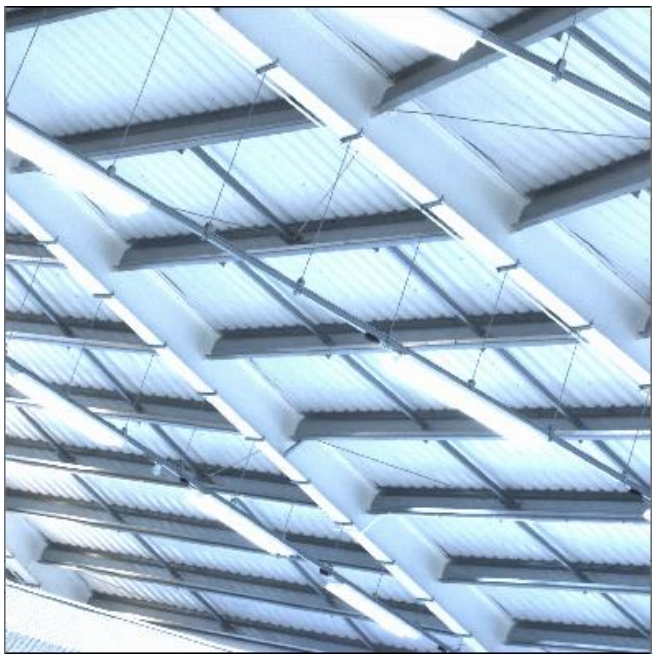


Figure 4.6. Example of photo taken from Leica Scanstation P20

The addition of the photos enabled an enhancement of the Point Cloud visualisation on Leica's proprietary software [151] where the colours captured in photos are mapped directly onto the Point Cloud as illustrated in Figure 4.7.

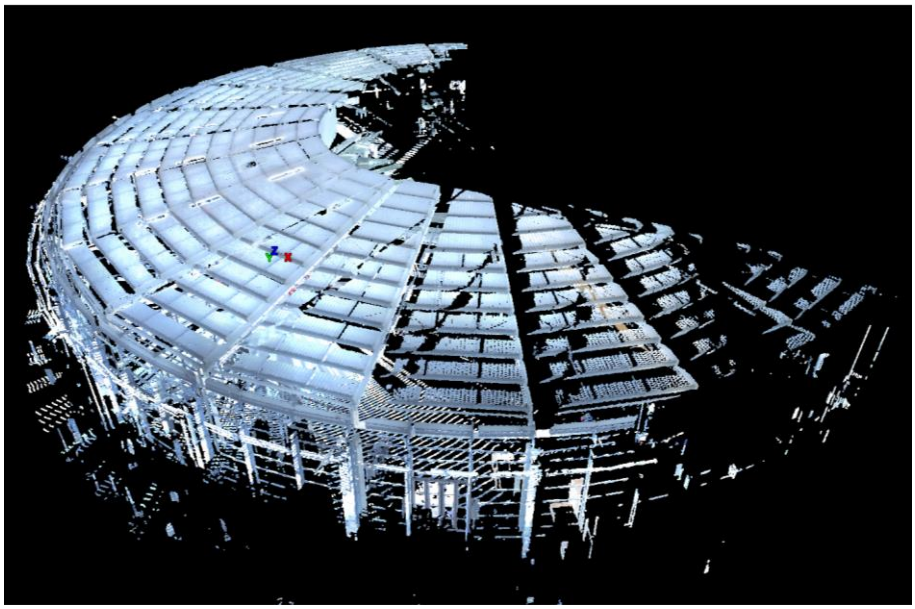


Figure 4.7. Enhanced Point Cloud visualisation with photo colours mapped onto measurements

The result was a set of individual scan data files that consisted of coordinates relative to the scanner position, intensity of each measurement and a real-world colour map for each measurement. Once captured as individual scans across a facility the scans must be registered together so that they share a common coordinate system.

4.3. Point Cloud Registration

4.3.1. Process Description

The process of Point Cloud registration required the alignment of several independent survey scans into a common coordinate system that represented the full survey dataset. Survey targets could have been used to facilitate the automated alignment of scans following the F2050 survey. This automated registration is conducted by proprietary algorithms in post-processing software and it has been observed to have varying degrees of success. Large levels of noise were generated in the F2050 scan from reflections and F2050 had a high degree of structural symmetry both of which negatively affected the potential to use the automated registration process. It should also be noted that in some circumstances, such as scanning F2050, the use of scan targets was impractical. As scan targets were not used for registering F2050 the scan registration process was conducted manually.

To register all of the collected scans together, relationships between two individual scans were specified several times for the full dataset until at least each scan had a relationship to at least one other scan. The relationships were defined by selecting two scans that were taken from adjacent scan locations, or had a line of sight with the same fixed object, in F2050. The first scan was taken as a datum and was fixed while the second scan was moved and rotated until alignment between the scans was achieved. This included from plan and elevation perspectives. An example of this registration process for F2050 is provided in Figure 4.8.

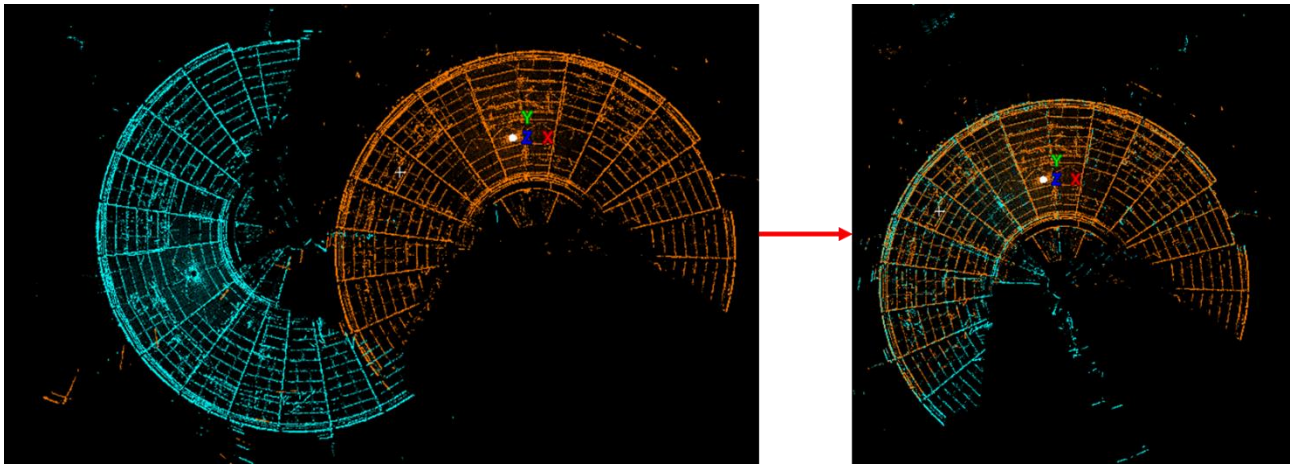


Figure 4.8. Manual registration of two Point Cloud scans

Once the two scans were closely aligned the registration software was used to optimise the specified constraint between the two scans which aimed to minimise a Root Mean Square (RMS) value when the data points of the two scans were compared. If the scans had substantial overlap and little noise it was relatively easy to achieve a low RMS (e.g. $6mm$). It became increasingly difficult to achieve a low RMS value when only a small portion of both scans overlapped, or substantial noise existed. In some cases, when the F2050 scans were registered, this was difficult to avoid, and a higher RMS value was only accepted following a thorough review and multiple attempts to keep the RMS value low.

Once a relationship existed between all scans in the dataset the Point Cloud was considered to be registered. An example of a registered F2050 Point Cloud was provided in Figure 4.9.

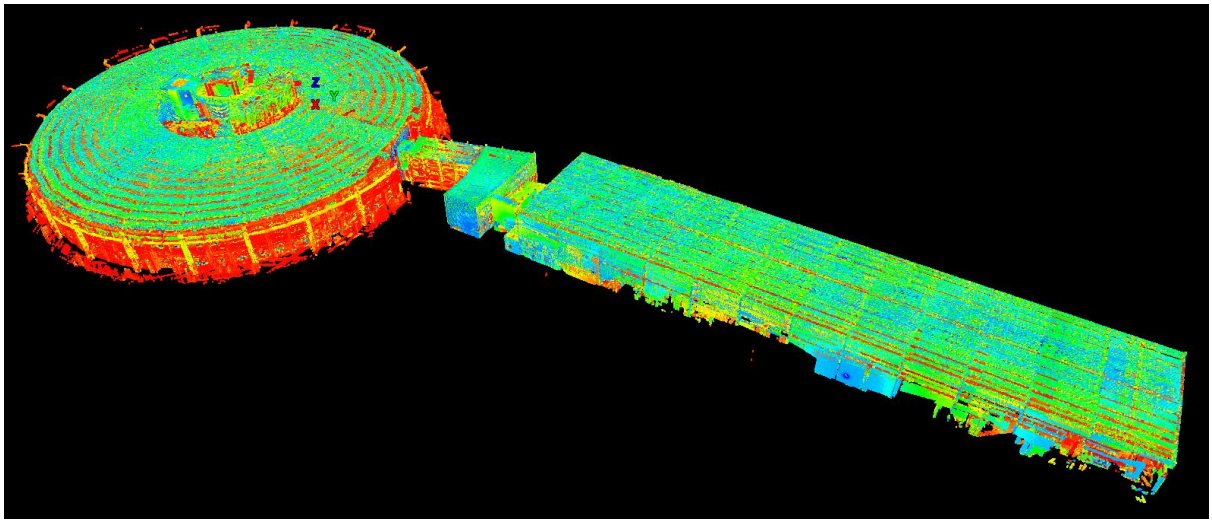


Figure 4.9. Registered Point Cloud of F2050

Some noise was manually removed from outside the external walls of F2050 in Figure 4.9 for clarity. This demonstrated a registered Point Cloud that consisted of several overlapping Point Clouds from individual scans where the total number of points in the scan registration was approximately 941 million. A common coordinate system was then shared between all captured scans that allowed for its use in other AEC applications to represent the as-built status of F2050.

4.3.2. Modelling from Registered Point Cloud

To establish the capabilities of existing Point Cloud software – namely Leica Cyclone [151], the registered F2050 Point Cloud was used to explore the capabilities of its existing tools. Two tools were identified that included a Point Cloud density filter and a Patch Growing tool.

The density filter of the Point Cloud software was tested to see how the built-in filters affected the final Point Cloud output and whether it might have reduced noise levels. This was conducted by implementing the "*Unify*" tool in the Point Cloud software and setting a specific density level. When activated, the "*Unify*" tool combined all the overlapping but separate Point Clouds into a single Point Cloud at a pre-specified and homogenous average point spacing. The exact method of how the Point Clouds were combined, averaged and spaced was not publicly disclosed however it was likely to utilise voxel grid representation which has been described in more detail in subsection 4.5.2. Figure 4.10 has illustrated the output of the Unified F2050 Point Cloud at two different densities. A Unified Point Cloud with no defined point spacing is shown on the left consisting of 676 million points, and when

a $2000mm$ point spacing was defined, as shown on the right, the Unified Point Cloud is reduced to 30,867 individual points.

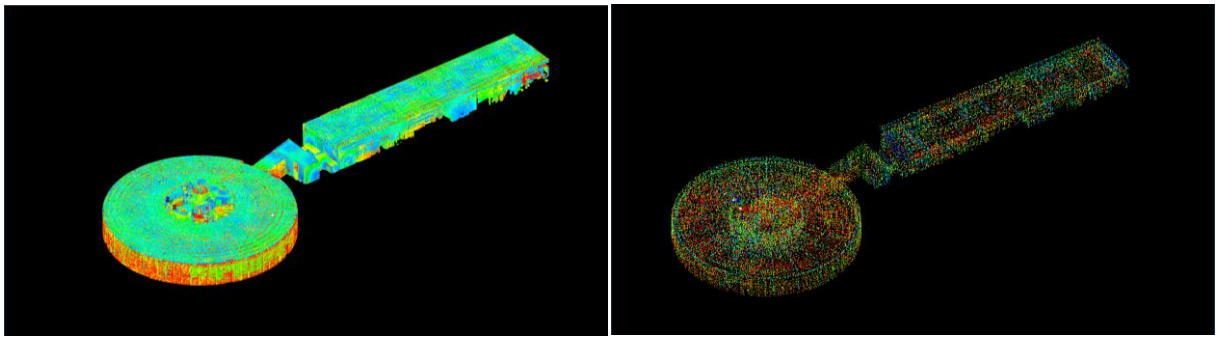


Figure 4.10. Output of unified F2050 Point Clouds, (left) native resolution, (right) 2000mm spacing

What this tool showed was that through Point Cloud reduction the basic structural shapes and forms of a building were retained and were visually recognisable and this was identified as potentially facilitating faster data processing methods. This was required to be balanced with the accuracy of how the Point Cloud represented the as-built form of F2050.

The Patch Growing tool was used to pick seed points in a plane (e.g. walls, floors and ceilings) and the tool used surrounding points, in close proximity to the seed point, to grow a planar patch. This was conducted on the non-reduced Unified Point Cloud to grow patches for the wall, floor and roof planes of F2050. This resulted in a Unified Point Cloud that had associated patches applied to the data points, see Figure 4.11 (top left). The patches were then isolated, see Figure 4.11 (top right) and exported from Leica Cyclone [151] in a ".coe" format for use in other software such as Autodesk Revit [34]. This manual point picking and region growing was found to be a time-consuming process. When the ".coe" file was imported into Autodesk Revit the patches were observed to be unconnected and not recognised as building elements useful for BIM or BEMS, see Figure 4.11 (lower).

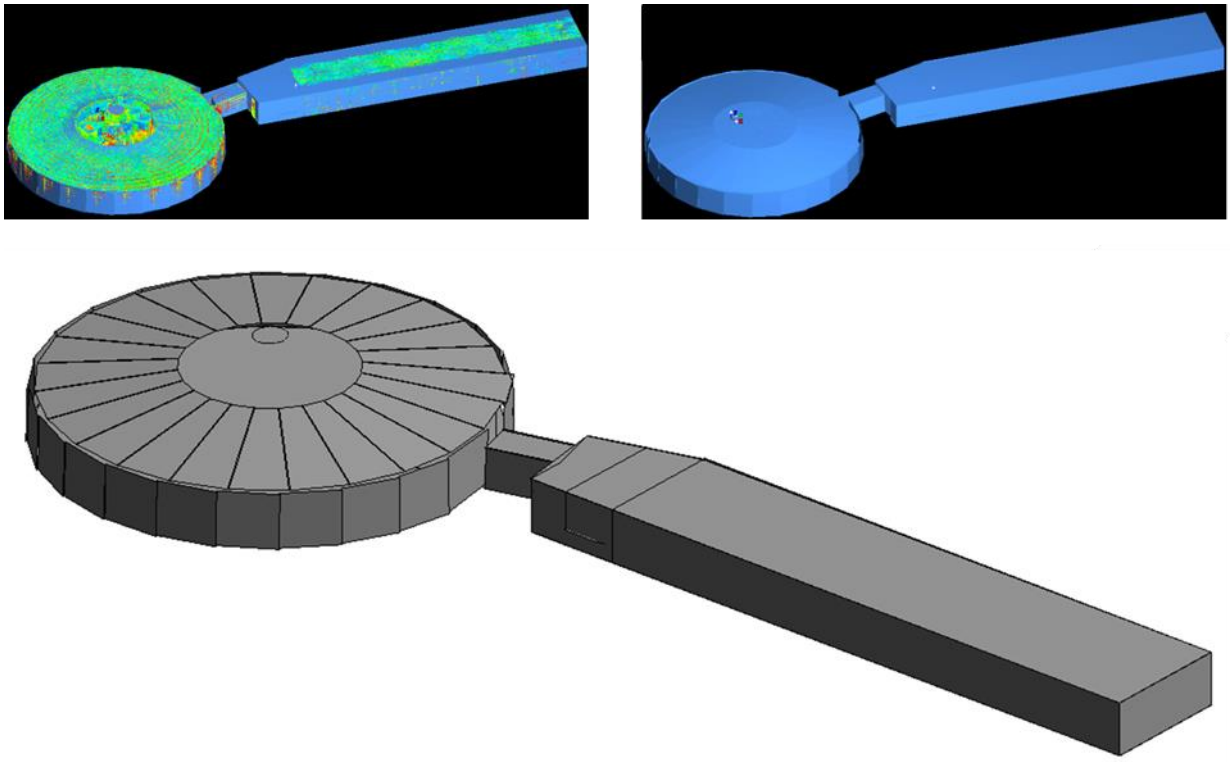


Figure 4.11. (top left) Unified Point Cloud with patches applied, (top right) Generated patches isolated from Point Cloud, (lower) View of F2050 .coe file in Autodesk Revit

Leica Cyclone [151] required development in order to provide the desired functionality of converting a Point Cloud into useful BIM or BEMS formats. Due to licensing restrictions Leica Cyclone, or any other commercial Point Cloud processing software, could not be developed by third parties without access to the necessary Software Development Kit (SDK). As such the use of Leica Cyclone was not promising for further research and development however it did illustrate that building features were identified from a Point Cloud and therefore the potential existed for a stand-alone development of a Point Cloud reconstruction solution.

4.3.3. Exporting Point Clouds

The native file format from the Leica Scanstation P20 was proprietary but once all of the individual Point Cloud scans had been registered they were exported as a single file in a more common point cloud format called E57. The E57 format has been widely used within the laser scanning industry as it has been used to provide a compact and structured method of storing large amounts of data such as that described above. The format was not owned by any single laser scanner manufacturer and was considered open source. The structure and method of data storage in the E57 format was specified by ASTM E2807-2011 [152].

The E57 format was, at a high level, stored in a hierarchical tree that used an XML format as described by Huber [153] and illustrated in Figure 4.12.

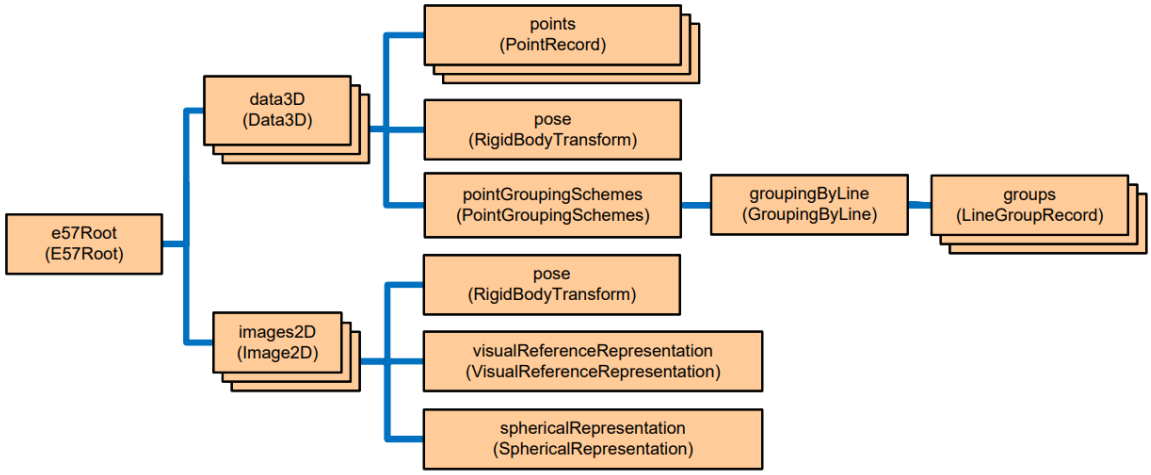


Figure 4.12. Illustrative example of E57 tree structure [153]

Due to the amount of data collected during a laser scan survey, storing all the captured data in this XML format would have been extremely inefficient from a read and write perspective. To improve this, the E57 format referenced a compressed Binary Large OBject (BLOB) of data. A data BLOB is used to store information such as 3D points or photographs, in each node of the XML hierarchical tree. This enabled efficient use of reading and writing large amounts of data while providing the flexibility offered by the XML structure. For reference, the full data captured across all F2050 scans in an E57 file required approximately 20GB of storage. Existing point cloud reconstruction tools within the literature were identified that used the E57 format.

4.4. Existing Point Cloud Reconstruction Tool

Following a further review of the work discussed by Ochmann *et al.* [111], which was previously described in Chapter 2, it was identified that a software prototype was developed as part of the DURAARK project [112] named “PC2BIM”. This prototype was freely available, and this subsection describes the preliminary investigation undertaken with the PC2BIM prototype.

The PC2BIM prototype was supplied with a lightweight Graphical User Interface (GUI) capable of viewing Point Clouds in an E57 format. The E57 format was used as it was able to distinguish between multiple scans within the file and as such unification of all overlapping

Point Clouds into a single Point Cloud was not necessary; this also retained a high resolution of captured geometry detail. The method employed by the PC2BIM prototype depended on the use of multiple scans so that surface normal directions were calculated based on where the scanner was located for each scan.

To test the prototype an E57 file representing the Point Cloud captured from a single position within F2050 was fed into the software, see Figure 4.13. Important features to note in this Point Cloud included the large amount of noise/erroneous points generated by scanning the glazed walls of F2050. During Point Cloud capture transparent and reflective materials, such as glass, refracted and reflected the laser radiation transmitted by the scanner. Time-of-flight instruments, such as the Leica ScanStation P20 [150], interpreted the radiation that returned from the glazing as being further away from the scanner than reality creating erroneous points in the captured Point Cloud.

The PC2BIM prototype performed a 3D model reconstruction of the Point Cloud. The results of this reconstruction of the Point Cloud shown in Figure 4.13 have been provided in Figure 4.14 and Figure 4.15.

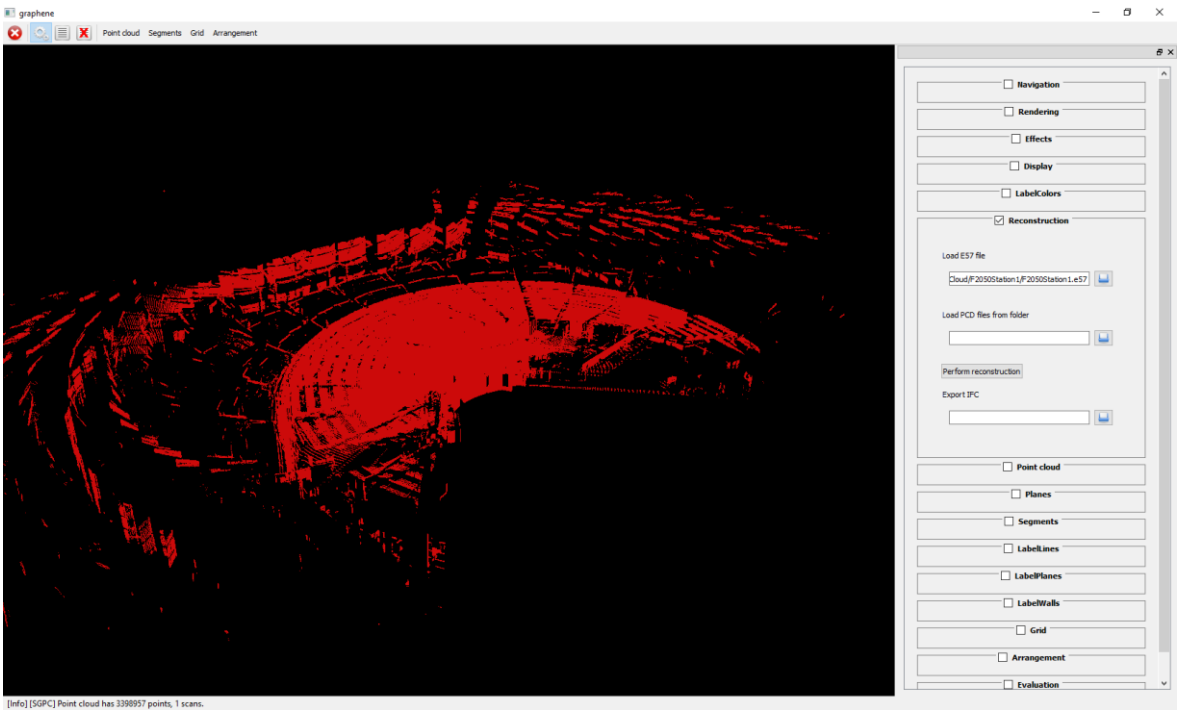


Figure 4.13. E57 file viewed in PC2BIM

Figure 4.14 demonstrated that the detection algorithms within PC2BIM produced a good match to approximate the layout of a small section of F2050 relative to the scanner which

created a single "room". All of the erroneous reflections were discarded by PC2BIM, due to small planar surface areas, which prevented the need for a time-consuming manual cleaning process. In addition, the algorithm generally seemed to ignore any internal clutter such as people and equipment.

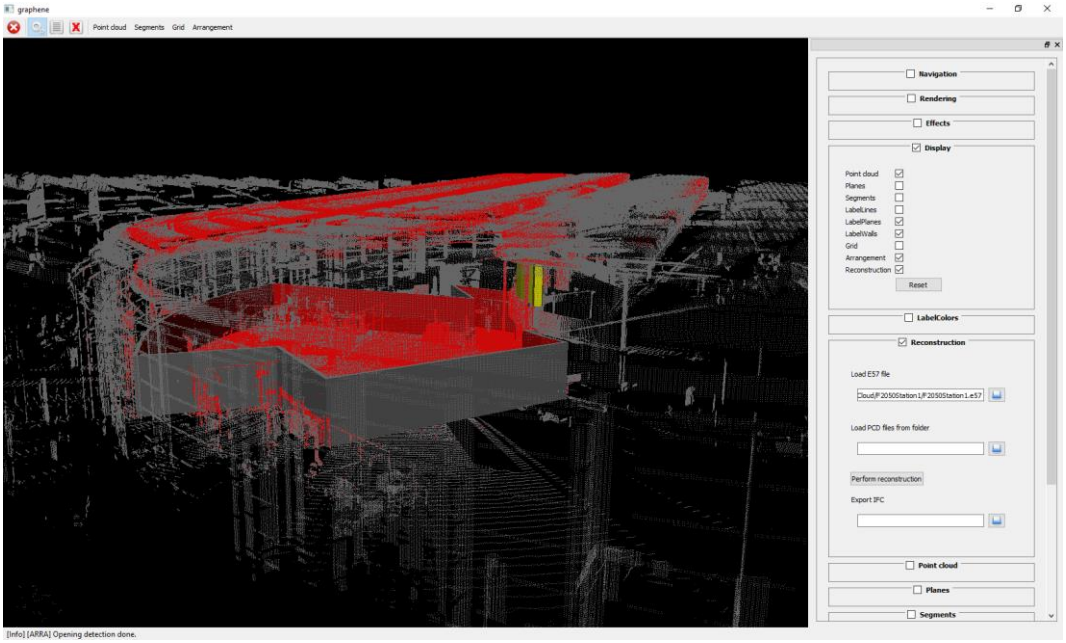


Figure 4.14. Resultant IFC reconstruction of single F2050 scan with point cloud

Further detail of PC2BIM detections has been provided in Figure 4.15, once the Point Cloud had been hidden. For example, a few yellow panels were identified correctly as glazed elements towards the inside of F2050. The results for the outside of the building are not quite as satisfactory. It is assumed these poorer results are due to the large glazed area present on the external walls of F2050. The outer wall reconstruction was actually set inside of the position of the real F2050 wall. Initially the algorithm searched for walls and then identified features on them such as windows and doors. The large glazing made it difficult to identify walls above a certain planar area threshold and so the associated Points Cloud data points may have been inadvertently discarded from consideration as erroneous reflections.

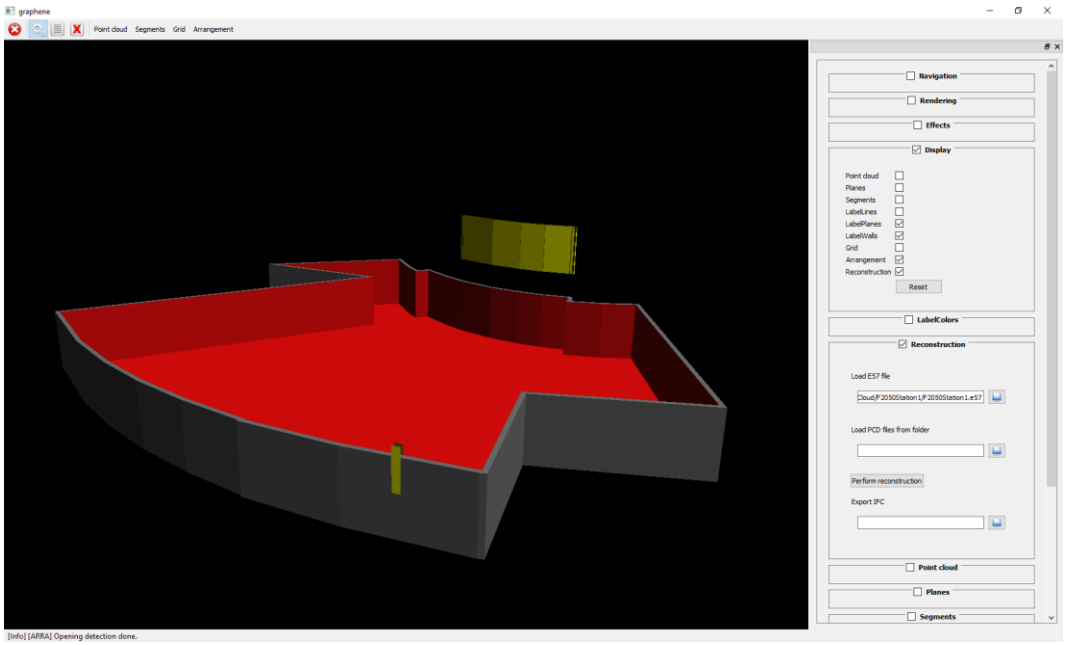


Figure 4.15. E57 file viewed in PC2BIM

Another key observation of these results suggested that the room height had not been correctly calculated and its method should have been refined. It was noted that the building layouts that the PC2BIM software had been tested on by its developers [112] were single storey offices. As such all rooms were of the same height in a single storey.

When combined with multiple scans Ochmann *et al* [111] suggested that room walls that have been generated by the program but that do not actually exist were eventually removed in PC2BIM as illustrated in Figure 4.16.

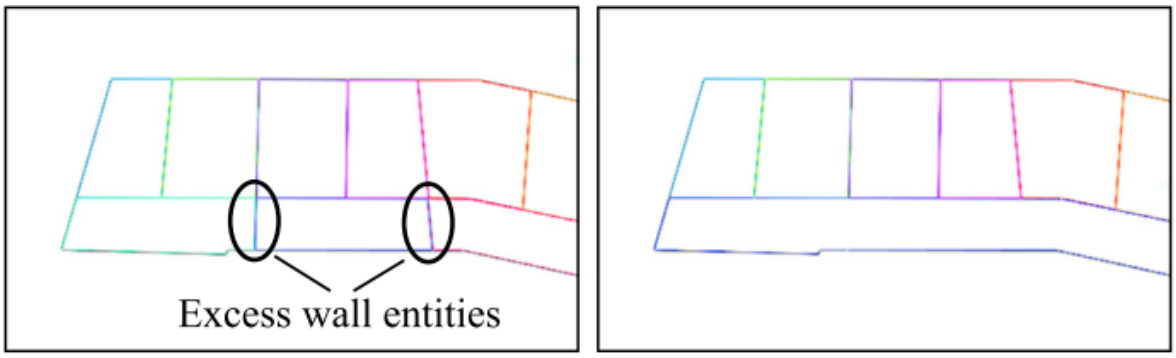


Figure 4.16. Removal of excess walls [111]

These results illustrated the basis of a powerful tool but with significant scope for development and improvement. As an additional test an E57 file was fed into the prototype of all 85 scans of F2050, see Figure 4.17. This test illustrated that the software was capable

of differentiating between each of the individual scans which were identified by being labelled with different colours. During an attempt to run the IFC reconstruction in the PC2BIM software, for this larger and more complex F2050 dataset of E57 scans, no results were achieved because of computational restrictions; Once again this highlighted a development opportunity. The source code used by the PC2BIM prototype was obtained for this research with the aim of further development for BEMS applications.

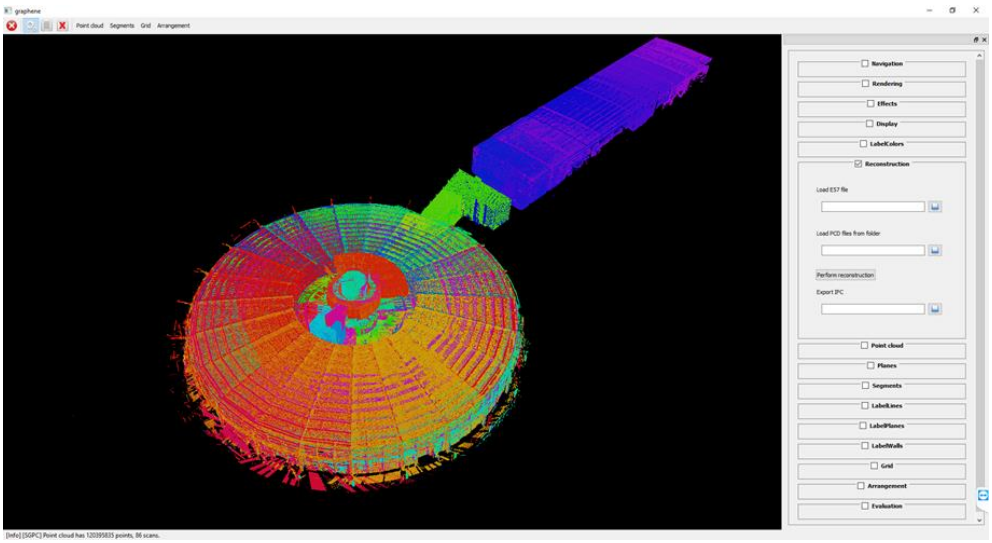


Figure 4.17. E57 file of 85 F2050 scans

4.5. Geometry Reconstruction Process

4.5.1. Normal Direction Estimation

An important feature of the approach carried out by Ochmann *et al.* [111] was the use of surface normal directions which were assigned to the Point Cloud. Knowledge of this direction allowed their method to establish the difference between an internal surface, with its surface normal pointing towards the scanner location, or an external surface. Following registration, the F2050 Point Clouds were absent of surface normal information and therefore surface normal estimation was required.

To facilitate this an open source software [154] was utilised to read the E57 file and compute the surface normal directions. The software relied on several libraries in the C++ programming language; libE57 [155], boost [156], eigen [157] and pcl [158] and their respective dependencies. The software had two mandatory inputs; filenames for the input and output E57 files. In addition, the user could specify two optional variables; 1) voxel size for Point Cloud down sampling, with 0.0 set as the default value, which has been described in more

detail in subsection 4.5.2, and 2) the number of nearest neighbours used for normal direction estimation, with 12 taken as the default value.

The pcl C++ library was directly responsible for normal estimation in the software [158]. Point Cloud data is recognised as an as-built record of the surfaces contained within an environment that surrounded a scanner during a survey. As such the surface normal directions could be approximated, through inference, by consideration of the Point Cloud points. The pcl library applied a least-square plane fitting estimation in order to estimate the normal direction of a plane that was tangent to the surface represented by the Point Cloud. This allowed approximation of the normal direction for a specific point on a plane, see Figure 4.18.

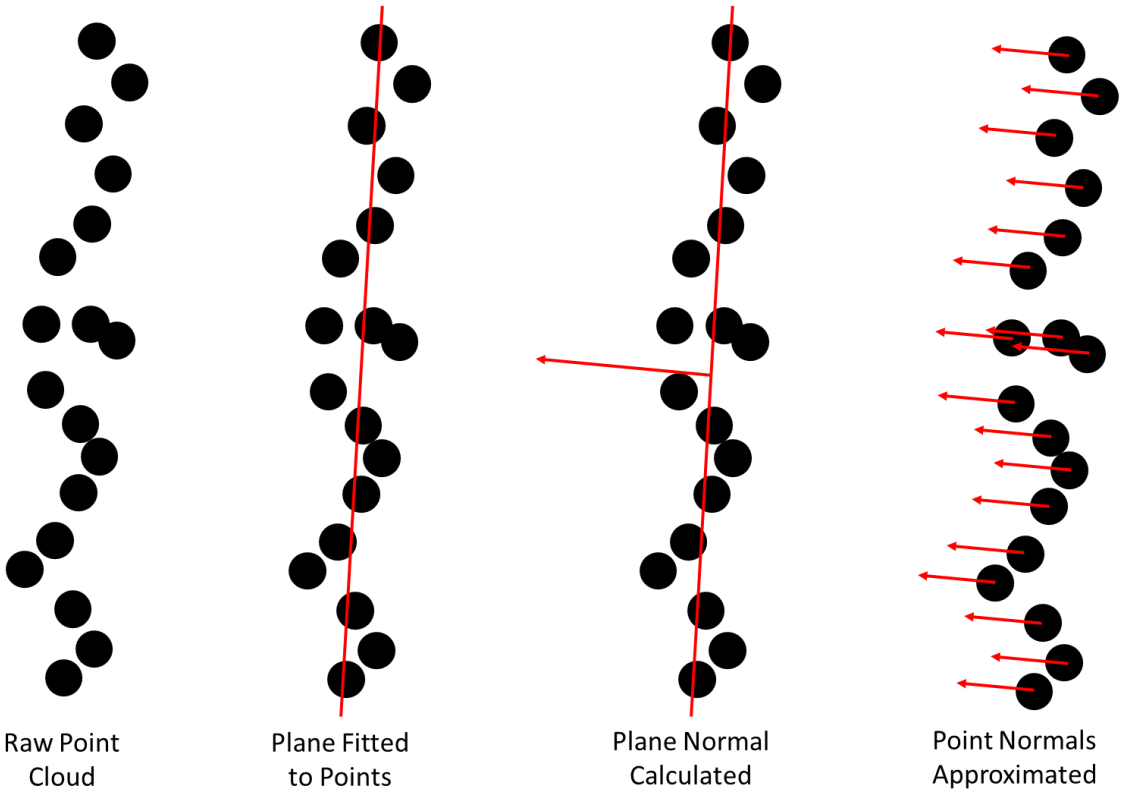


Figure 4.18. Point Cloud normal approximation

Rusu [159] described that the solution to estimating the surface normal direction used eigenvectors and eigenvalues in a covariance matrix of the nearest k_{nn} neighbour points, as defined through user input, to a seed point $p_{pc,i}$ such that a covariate matrix C can be created as per Equation 4.1 where $\overline{p_{pc}}$ was the 3D centroid of the nearest neighbour set, λ_j was the j-th eigenvalue of the covariance matrix, and $\vec{v_j}$ were the j-th eigenvector.

$$C = \frac{1}{k_{nn}} \sum_{i=1}^{k_{nn}} \left(p_{pc,i} - \overline{p_{pc}}\right) \cdot \left(p_{pc,i} - \overline{p_{pc}}\right)^{\hat{T}},\ C \cdot \vec{v_j} = \lambda_j \cdot \vec{v_j},\ j \in \{0, 1, 2\}$$

Equation 4.1. Covariate matrix for nearest neighbour normal estimation

This method computed a normal direction $\vec{n}$ which was assigned to each of the points, but it was unable to apply a consistent sign across the dataset. However, with Point Clouds the scanner location was known and therefore orienting all normal directions, following the estimation process, toward the scanner was a straightforward process, see Equation 4.2.

$$\vec{n}_i \cdot \left(v_p - p_{pc,i}\right) > 0$$

Equation 4.2. Ensuring consistent normal direction towards scanner

The pcl authors warned that the user must carefully consider the value assigned to the k_{nn} nearest neighbours. If too large a value was selected points included in the normal estimation dataset may have included many points from two distinct planar surfaces. This would have had a substantial distortion effect on normal estimation and efforts taken to avoid it.

For estimating normal directions in the F2050 E57 file the default value of 12 nearest neighbours was used. As explained in subsection 4.2 the building survey was conducted at a resolution of 12.5mm at 10m from the scanner. This meant that for a point located 10m away from the scanner all of the points used in the normal estimation were contained in a radius of approximately 25mm which was sufficiently low enough to avoid the distortions described above, see Figure 4.19. The only exception to this would have been where planar surfaces meet (e.g. room corners) but transition between the surfaces was expected in these locations and any distortions would have been limited to surface planar boundaries.

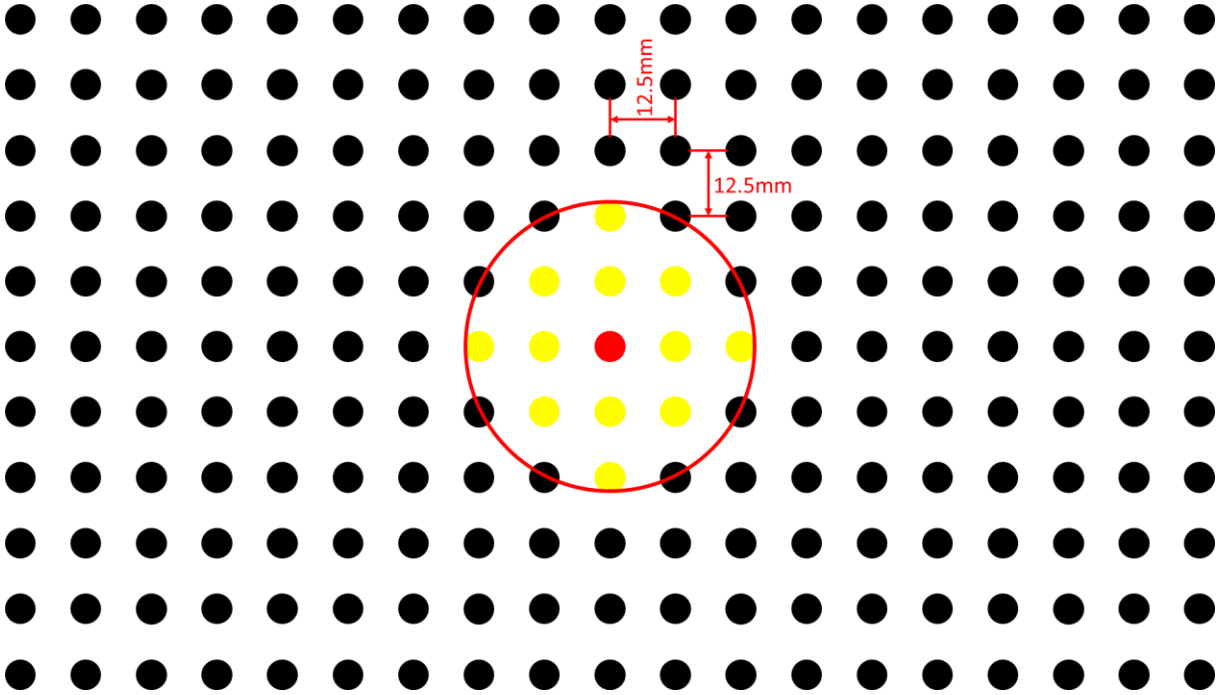


Figure 4.19. The 12 nearest neighbour points (yellow) to the seed point (red)

4.5.2. Point Cloud Resolution

While the pre-processing described in subsection 4.5.1 was primarily for the estimation of surface normal directions it allowed the optional possibility of reducing the size of the Point Cloud. This had benefits of requiring less computational effort to process a dataset however it also degraded the as-built likeness of the dataset. In order to down sample a Point Cloud dataset a technique known as voxel grid filtering was applied and carried out using the pcl C++ library [158].

A voxel was defined as an element within an array of discrete volumes in a 3D space. Using the pcl library the user defined the length of one side of cubed voxels to be used in the voxel filter [158]. Following size definition, a 3D grid of voxel volumes was overlaid on the input E57 point data. Each voxel encompassed a varying number of points from the original dataset and the centroid of the encompassed points was calculated. This created a reduced size Point Cloud data set where the centroid of each voxel replaced all of the data points from the original Point Cloud that were contained by the voxel volume. A 2D representation of this process has been illustrated in Figure 4.20.

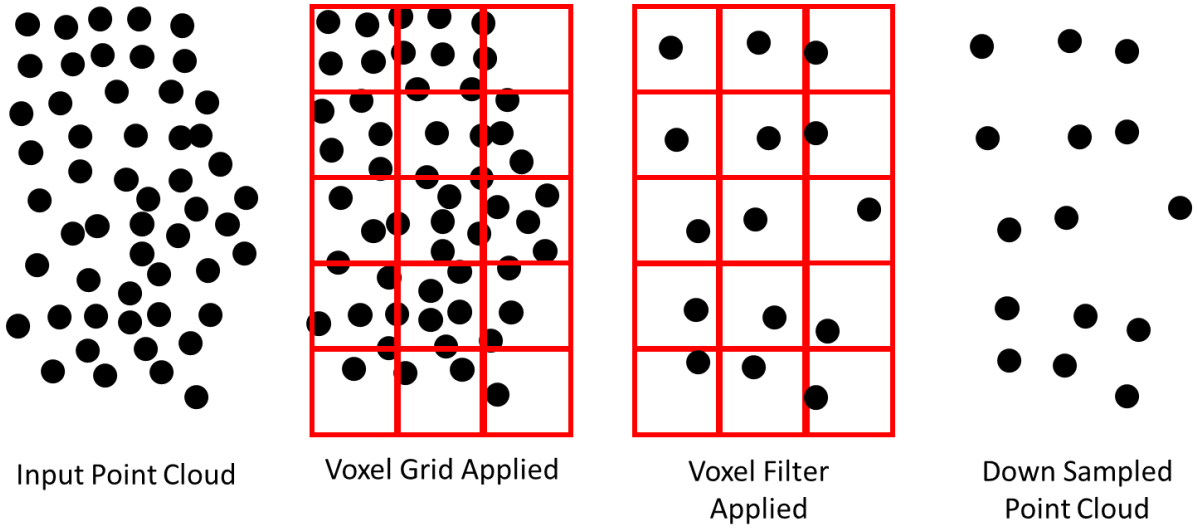


Figure 4.20. 2D demonstration of voxel filter process

This had the result of applying a coarse filter to the Point Cloud data and drastically reducing the number of points to be processed which resulted in smaller file sizes. Another simpler method of Point Cloud down sampling was to only select a subset of the full registered scans. If the selected subset of scans still maintained overlap or line of sight with each other reconstruction could still have been possible. Both of these down sampling methods could have been combined too. Using down sampled data sets may have had the effect of reduced accuracy during reconstruction attempts. As an investigation into the effect of voxel resolution on E57 file size, all of the scans from the ground floor of F2050 were selected as a subset and several different voxel filters at differing resolutions were applied. The selected resolutions ranged from $0mm$ (i.e. no down sampling) to $100mm$ in $5mm$ increments. The resulting E57 file size has been plotted in Figure 4.21 which demonstrated a drastic file size reduction when voxel gird filtering was applied.

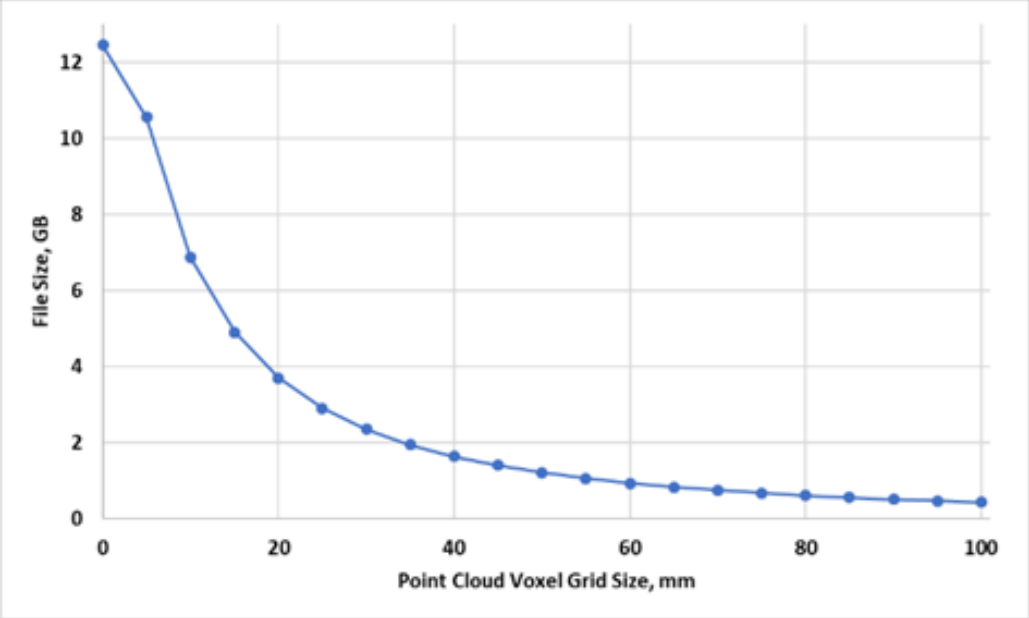


Figure 4.21. E57 file size vs voxel resolution

4.5.3. Automated Reconstruction

The source code used in the PC2BIM prototype, as described in subsection 4.4, was obtained for further development in this research. This subsection has described the additional development of the PC2BIM prototype conducted for this research. The source code that was developed did not interface with a GUI, like the PC2BIM prototype, and commands were passed by the user through the command prompt on Windows 10.

The automated reconstruction tool was written in the C++ programming language and is dependent on several different primary libraries in addition to the standard C++ libraries; boost [156], CGAL [160] eigen [157], pcl [158], e57_pcl [161] as well as their respective dependencies. New libraries that were added as dependencies during development included XSD [162] for parsing data into gbXML.

The automated reconstruction process consisted of twelve separate steps which have been described in full detail in Appendix E. As part of this research further developments were made to “Step 10 Geometry Reconstruction” and the “Step 12 Export BEMS Geometry” steps as described in this section.

10) Geometry Reconstruction

In previous steps of the automated reconstruction, having extracted edge and face data that represented the 2D floor plans of a building and the elevations of individual walls in the

arrangement, the data was processed for reconstruction purposes. Individual wall elevations were previously calculated, and these were now considered at a room basis by finding a common mean height among all of the walls that surrounded a single room. This was then used to define the distance between the opposing floor and ceiling faces to determine a room height.

The next step defined each room and the room definition was developed further than in previous research. Originally a room was defined by a list of node coordinates where the walls of the room intersected, the floor and ceiling heights and the floor area of the room. In addition to these features the room definition was enhanced to include a list of edges, in order, around a room perimeter and the face index used for the floor and ceiling. Doing so provided a more complete picture of the elements that comprised a room and how they related to one another.

Following this enhancement in room definition, the individual walls of the reconstruction were defined, the process of which remained unchanged from previous research. It was possible that a single wall consisted of multiple edges as shown in Figure 4.22. The wall definition included a list of these edges, the room index associated with the wall, the node coordinates at the start and end of the wall, vertical coordinates of the wall base and top, wall thickness, wall direction and wall length. In addition, an empty data store was created to reference any openings (i.e. doors, windows or holes) detected within the wall planar surfaces during a later stage of geometry reconstruction.

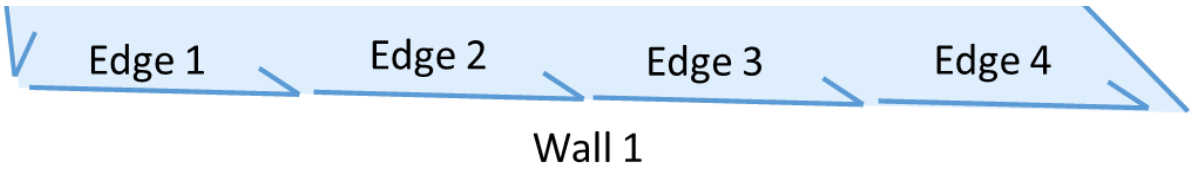


Figure 4.22. Single wall that consisted of several edges

In the original PC2BIM prototype the individual walls, floor and ceiling were defined as disparate surfaces which is suitable for BIM geometry definition. This definition included nodes around the edge of each surface, the assumed thickness of the surface and whether the surface represented a wall, floor or ceiling. This was subsequently enhanced in this research such that references were made between each surface and the edges that made

up the perimeter of each surface. While this was suitable for BIM definition it was still lacking in detail to provide geometry definition for BEMS.

The geometry reconstruction was then developed further to consider a group of identified surfaces in terms of thermal volume rather than individual and discrete surfaces. This involved the definition of thermal surfaces specifically for BEMS application. This meant prior to conducting this research each reconstruction element was not well related to other components by keeping horizontal and vertical building elements completely separate (e.g. the walls were not easily related to the floor and ceiling).The thermal surfaces were defined such that they represented the centreline of any wall, floor and ceiling that were previously defined but, crucially, they were used to create watertight thermal volumes by relating all surfaces surrounding a volume to each other, see Figure 4.23.

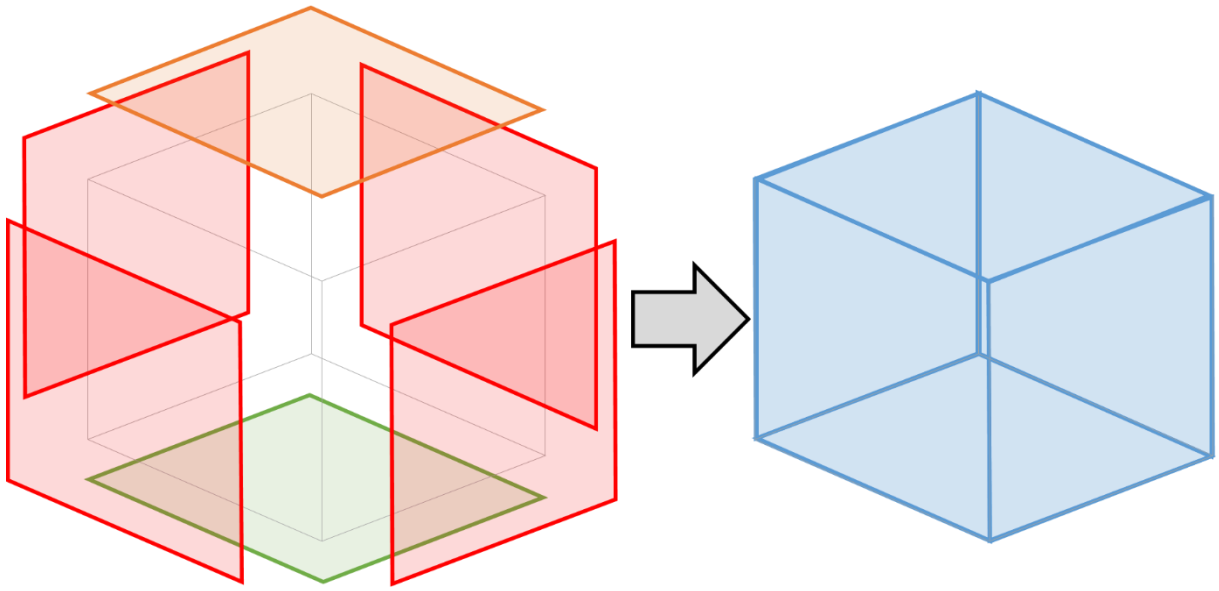


Figure 4.23. Reconstructing thermal surfaces (right) instead of individual construction elements (left)

This thermal surface arrangement provided a fuller definition of the geometry of each thermal volume that allowed consideration of finite volumes for energy assessment in BEMS.

12) Export BEMS geometry

Additional developments were made so that the reconstructed geometry could be written into a gbXML format and exported for BEMS applications. It considered the full schema definition and took steps to identify the minimum elements in the gbXML hierarchical tree that needed defining in order to produce a validated gbXML file following the geometry

reconstruction process [163,164]. Describing the data output as a validated gbXML file provided recognition that the file had been written to comply with the gbXML schema definition and was not related to the quality of information contained within the gbXML file (e.g. room dimensions). While some researchers have used gbXML in reconstruction work they had not defined the full minimum schema required for reconstruction applications [81].

Figure 4.24 defined the minimum gbXML elements that were required to ensure a validated gbXML file. Interrogation of the elements showed how the new definition of thermal surfaces during the reconstruction process aided the gbXML file creation especially in the "Surface" and "ShellGeometry" tiers of the XML tree.

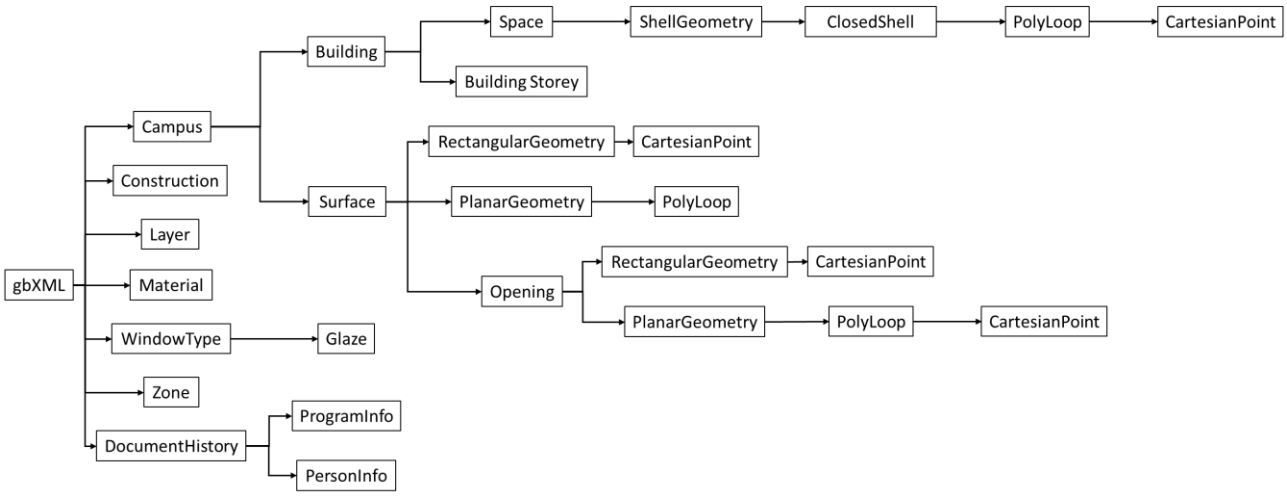


Figure 4.24. Minimum element definition required in gbXML for Point Cloud reconstruction

Figure 4.24 did not define terminating elements in the gbXML structure for clarity. The remaining gbXML terminating child elements and their parents are tabulated in Table 4.1.

Table 4.1. Terminating gbXML child elements

Parent gbXML Element	Terminating Child gbXML Element
Campus	Name
Space	Name
	Area
	Volume
	TypeCode
CartesianPoint	Coordinate
BuildingStorey	Name
	Level
Surface	AdjacentSpaceId
Surface->RectangularGeometry	Azimuth
	Tilt

Parent gbXML Element	Terminating Child gbXML Element
	Height
	Width
Opening->RectangularGeometry	Height
	Width

Having defined the required gbXML elements the results of the reconstruction process were parsed from memory into the correct place in the gbXML structure thus producing a validated gbXML file that could be used for BEMS applications. When an E57 file containing a single laser scan of F2050 was processed, a gbXML file was produced as demonstrated in Figure 4.25 using DDS-CAD Viewer [165]. The reconstruction result showed that room boundaries had been defined as well as floor and ceiling surfaces. Visible within the wall surfaces were rectangular planar patches where openings have been identified. These opening classifications included windows, doors and an invalid opening categorisation.

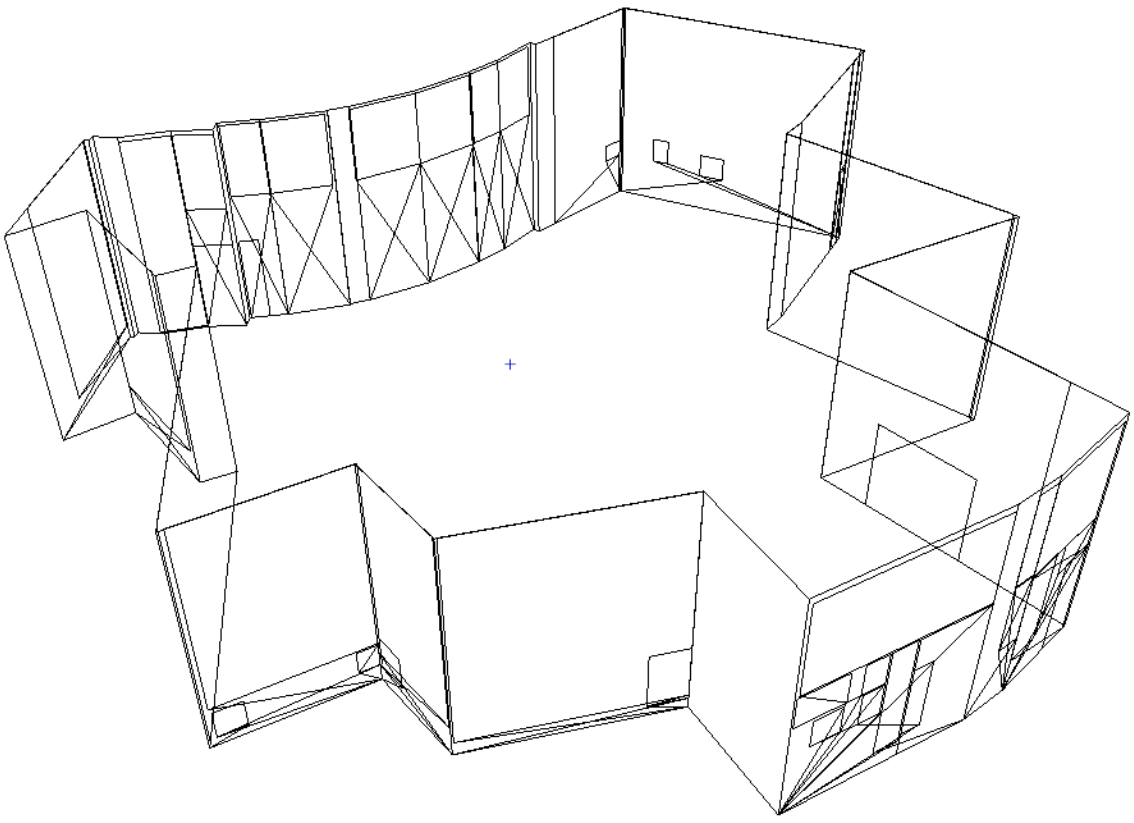


Figure 4.25. Example of gbXML geometry from a single F2050 scan

The description of developments to the automatic reconstruction process from previous research has been concluded to apply the technique for BEMS in this research. An E57 file,

complete with point normal directions, was provided as an input to the software and the software produced a valid gbXML file describing the reconstructed geometry of the environment captured in the E57 file. This process was then subsequently applied to the full survey data captured as described in subsection 4.2 on F2050 to conduct reconstruction of the building geometry for BEMS applications.

4.6. F2050 Reconstruction

4.6.1. Preliminary Results

Following further development of the automatic reconstruction software, as part of this research, it was tested on scans of F2050 that were obtained as described in subsection 4.2. Compared to research previously undertaken, F2050 represented a real-world operational environment complete with large amounts of environmental clutter and non-typical building shapes (i.e. the circular workshop). The building also had large amounts of floor to ceiling glazing that generated a considerable amount of noise during the laser scan survey. Previous research has used simplified environments, but substantial research value can be garnered through testing the developed software in a real-world scenario. The F2050 environment has provided a benchmark on the maturity of using such a software within and outside of a purely academic sphere of study in a top-down approach.

As indicated in Appendix D the laser scans for F2050 took place over several stories and these were all included within the Point Cloud registration. One limitation of the reconstruction software was that it was only capable of processing a single floor instead of a complete building. The complex arrangement of floors in F2050 meant that full height of the circular workshop also encompassed the volumes of storeys enclosed in the centre of the building, see Figure 4.26.

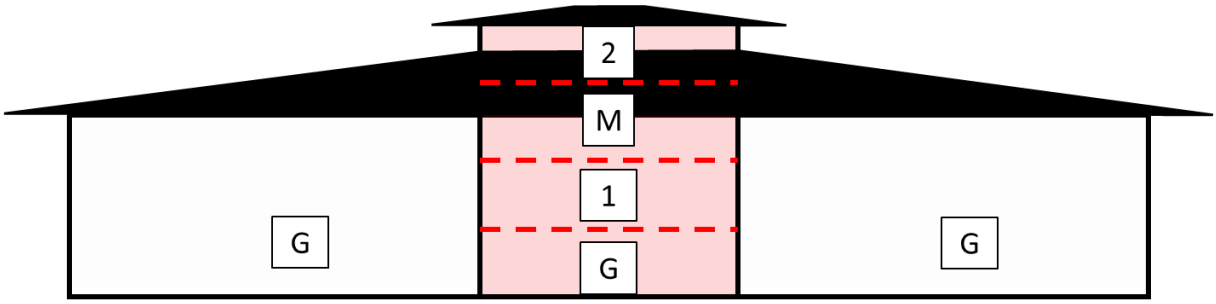


Figure 4.26. Multiple storeys of F2050's circular workshop

Although the circular workshop only encompassed half the height of the top storey, the reconstruction of the workshop's thermal volume would have included the internal volumes of storeys G, 1 and M as well as a large proportion of storey 2. In consideration of BEMS the top storey was only used for large and infrequent meetings and as such did not exhibit large levels of regular internal gains therefore its partial exclusion from BEMS geometry reconstruction was expected to only have a minor effect of the BEMS results of the whole of F2050.

When the full set of F2050 scans, without any down sampling, were passed through the reconstruction software, complete with multiple storeys, the software was unable to produce any results and software crashes were a common occurrence before reaching the end of Point Cloud processing. During the programme execution the computer RAM capacity (128GB) reached its maximum and remained full until the crash occurred.

The Point Cloud input data was manually down sampled by selecting only those scans that were taken on the Ground floor around F2050. This immediately resulted in a reduction from 85 to 51 scans and the E57 file size was reduced to 9.6GB once it had been processed to include normal estimations for the individual points. The input E57 file that was fed into the reconstruction process consisted of 378 million data points and, unfortunately, the reconstruction process was unable to complete successfully. This was most likely due to the large presence of noise generated in the laser scan survey.

Subsequently, the set of F2050 ground floor scans was then batch processed to estimate normal directions and apply a voxel grid filter in an attempt to reduce the effect of large noise levels on the reconstruction process. The voxel grid filter was applied in resolutions starting from 10 mm to 100 mm in 5 mm increments. When these Point Clouds were processed by the reconstruction software results were only achieved at voxel grid resolutions of 15mm and 30mm. It is likely that other resolutions had too much noise in them or not a sufficient amount of structured geometry data for any reconstruction data to be produced. The resulting geometry written into a gbXML file has been provided in Figure 4.27 and Figure 4.28 respectively.

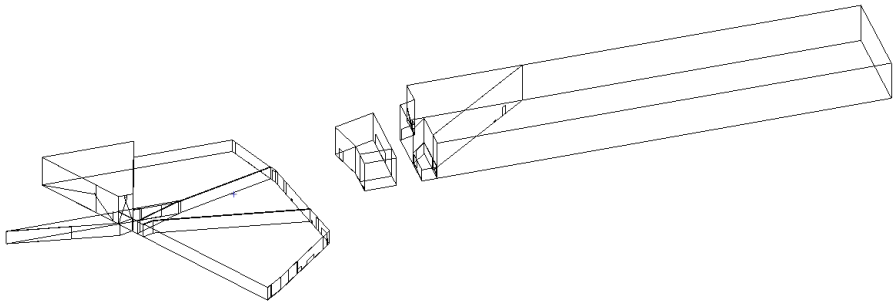


Figure 4.27. F2050 reconstruction from 15mm voxel filtered ground floor scans

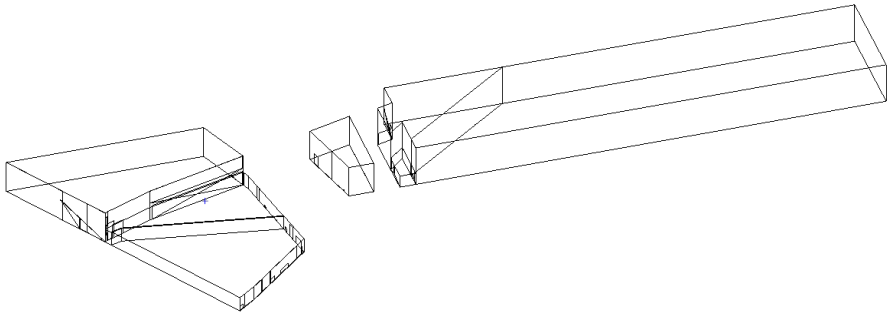


Figure 4.28. F2050 Reconstruction from 30mm voxel filtered ground floor scans

Both reconstructions captured the rear portions of F2050 well but struggled with the circular workshop of F2050 and the connecting walkways, both of which were highly glazed and therefore produced large amounts of noise in the survey scan through reflections.

In comparison of the two voxel gird resolutions the reconstruction of the $15mm$ voxel grid filter had detected more features than the coarser $30mm$ filter. This highlighted that while incorporating a voxel filter may reduce noise sufficiently to achieve reconstruction results, a coarser filter may eventually become detrimental to the results.

A key observation that was made of the reconstructions were that the software struggled when assessing the large levels of glazing. When considering how the reconstruction process used ray casting to identify wall surfaces first followed by detecting openings within the walls the difficulty in detecting fully glazed walls was evident. Potential solutions to this would be to cover the glazing with blinds or use twinned photograph data to perform image recognition of glazed walls to assist the reconstruction.

4.6.2. Survey Noise Reduction

In order to combat the difficulty posed by the glazing and high levels of noise in the reconstruction process, a second smaller site survey was conducted with the laser scanner

of the circular workshop in F2050. The difference in the second survey was that the large glazing of F2050 was covered using the tight-fitting blinds present in the building. The scan map for this second survey has also been included in Appendix D. Subsection 4.2 had already shown how this drastically reduced the level of noise captured in the survey. These new scans were registered with some of the existing ground floor scans of F2050 which created a ground floor survey that consisted of 67 scans.

While the use of blinds prevented the capture of structural details, such as the steelwork around the windows, the potential to capture the full thermal volume instead of opening details was more beneficial when reconstructing for BEMS applications. It was identified, based on knowledge of creating the benchmark F2050 BEMS, that opening details can be added in manually within BEMS software with ease when required.

Several small manufacturing cells were present within the circular workshop of F2050 and these were surrounded by clear Perspex safety barriers. These also created internal noise through reflections like the external glazing surfaces. To combat this noise generation that was created in the survey, the scans passed into the reconstruction software were manually down sampled from the full set of 67 scans. The purpose of scan selection was to select the minimum number of scans that could adequately represent a full survey of F2050 for reconstruction.

The full set of 67 scans had large levels of overlap, and therefore noise, in the environment captured. As such, manually reducing the number of scans passed into the reconstruction software reduced the computing power that was required to process the dataset. This also reduced the amount of errant noise that needed to be processed increasing the chance of successful reconstruction. The minimum subset of scans that resulted in successful reconstruction without any voxel filtering was identified as 26 as presented in Figure 4.29.

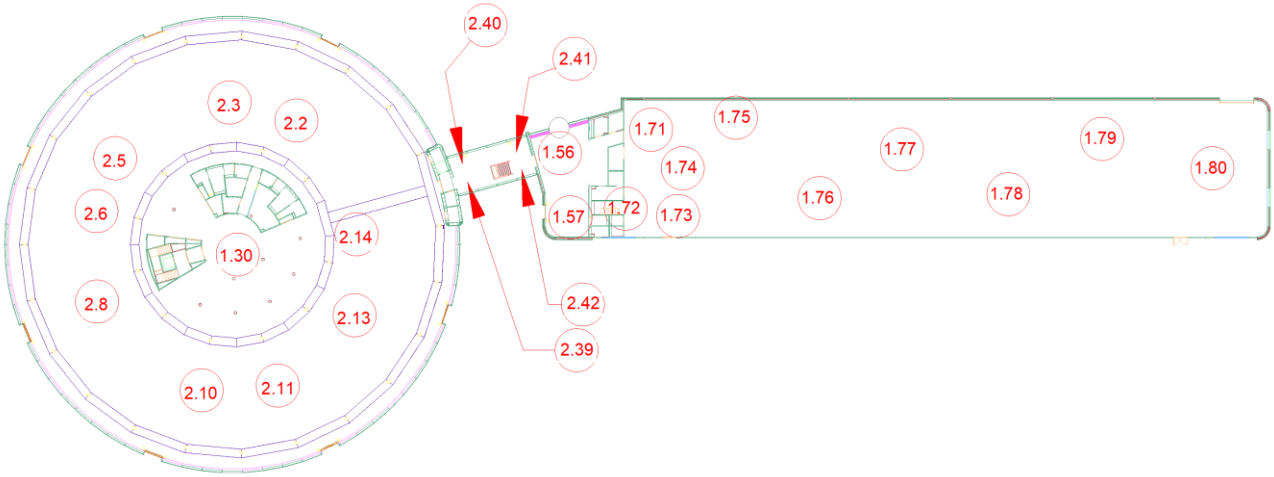


Figure 4.29. Reduced scan selection for F2050 reconstruction

This subset of scans was provided as input to the reconstruction software and the resulting geometry, written into a gbXML format, has been displayed in Figure 4.30 and Figure 4.31.

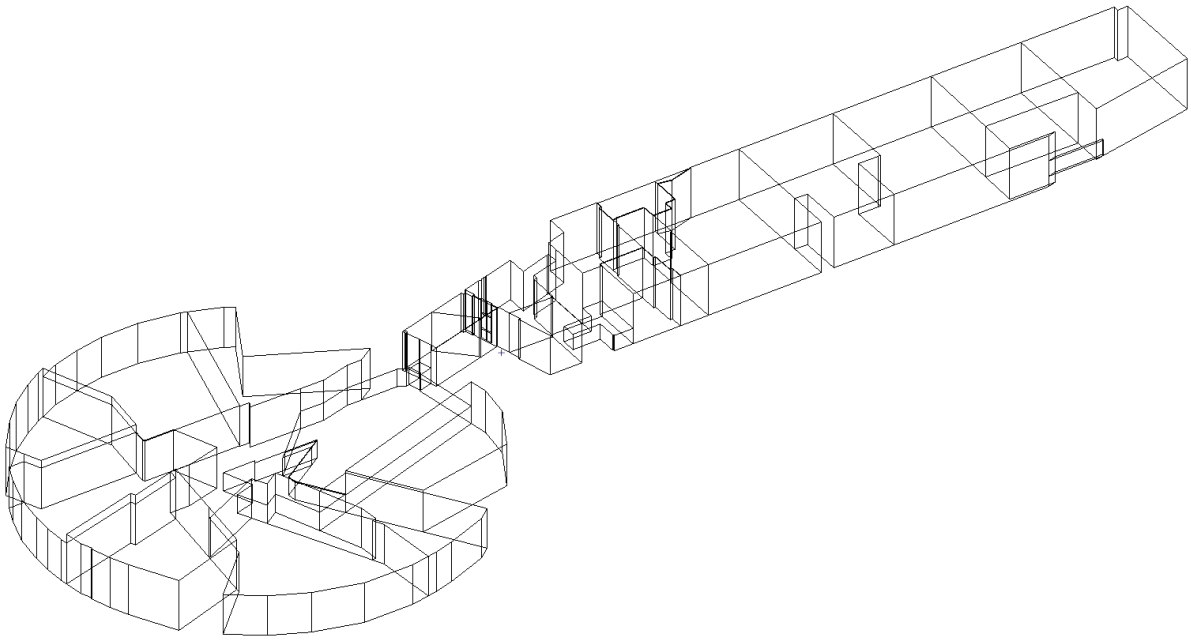


Figure 4.30. F2050 reconstruction of 26 ground floor scans w/ blinds down

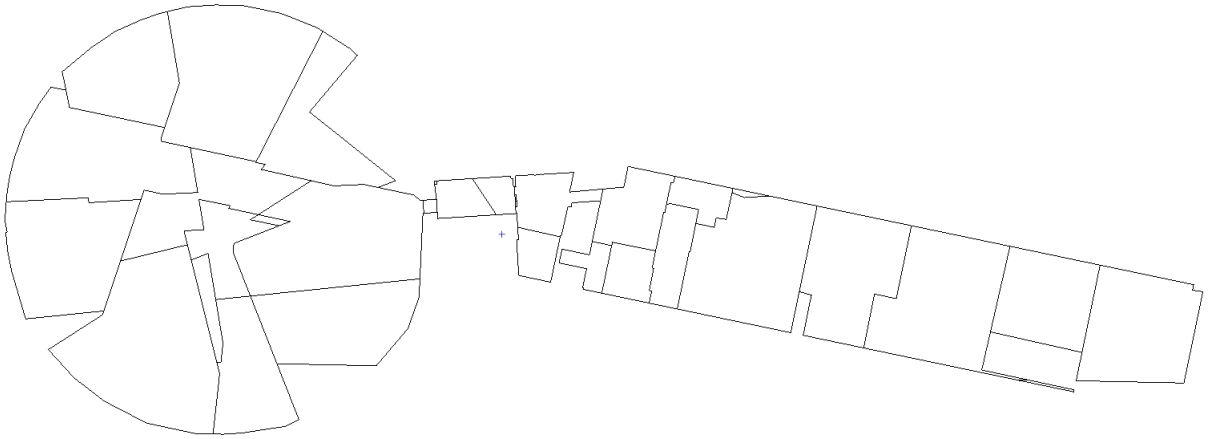


Figure 4.31. Plan view of F2050 reconstruction using 26 ground floor scans w/ blinds down

This demonstrated a successful method of dealing with high levels of noise within a survey in order to achieve an improved geometry reconstruction. Compared with reconstruction attempts that were made when blinds were open in F2050, the circular workshop achieved much greater definition when reconstructed. The rear portion of F2050 had been more heavily segmented than previous attempts and was likely to be a result of processing the whole of F2050 as a single reconstruction rather than in its constituent parts. All of the connecting corridors had been reconstructed so that no gaps between sections of F2050 existed.

As previously mentioned, the use of blinds to establish the volume boundaries for reconstruction prevented the identification of window openings in the outside shell of F2050 and any subsequent BEMS applications needed to manually add these openings back into the F2050 geometry. It was clear from the reconstruction results that further development of the software was still required to fully capture the building geometry without missing segments. The cluttered environment of F2050 caused occlusions of building features during the scan which resulted in sections of missing building geometry. The most prominent example of this was demonstrated in Figure 4.32. In the area, indicated by a red circle, of the F2050 circular workshop a large wing span with a reflective surface, for aerospace research, was installed in a test rig. This made it difficult to see the external wall behind the test rig which generated large amounts of reflective noise during the surveys. If the Point Cloud data was twinned with photography data and image recognition techniques this barrier may be overcome but was outside the scope of this research.

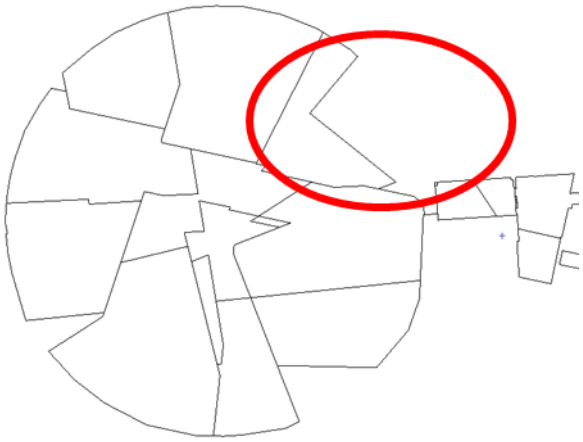


Figure 4.32. Missing segment of building geometry during reconstruction

To combat this additional noise, the subset of 26 scans were passed through the voxel filter to generate E57 files at resolutions of $10mm$ to $100mm$ in $5mm$ increments. These were then automatically processed to reconstruct the geometry of F2050. The 3D views of all successful voxel filtered reconstructions have been provided in Appendix F. Of these reconstructions, only one was deemed sufficiently acceptable to use in BEMS applications. This occurred at a voxel grid resolution of $10mm$ as presented in Figure 4.33 and Figure 4.34. These figures demonstrated an improved reconstruction of the perimeter of the Circular workshop, when compared with Figure 4.32. While geometry was reconstructed at other voxel grid resolutions the visual quality decreased as the voxel grid cubes increased in size. For this reason, only the reconstruction without voxel filtering and a $10mm$ voxel filter were used in further BEMS research.

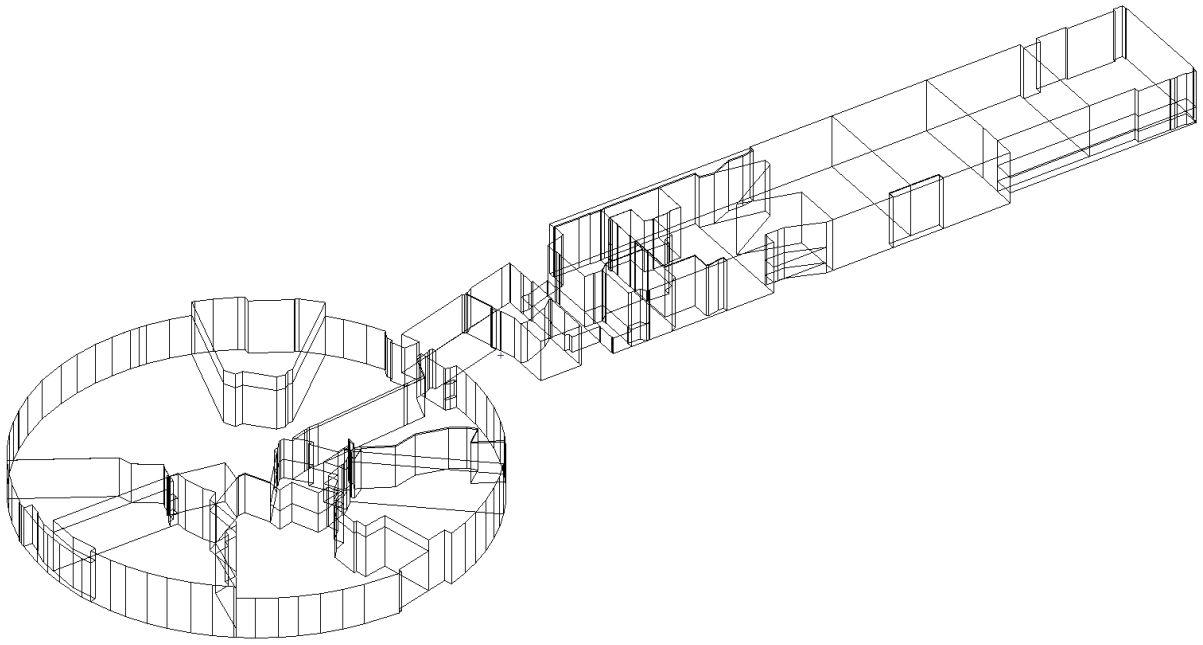


Figure 4.33. F2050 10mm voxel filter reconstruction using 26 ground floor scans w/ blinds down

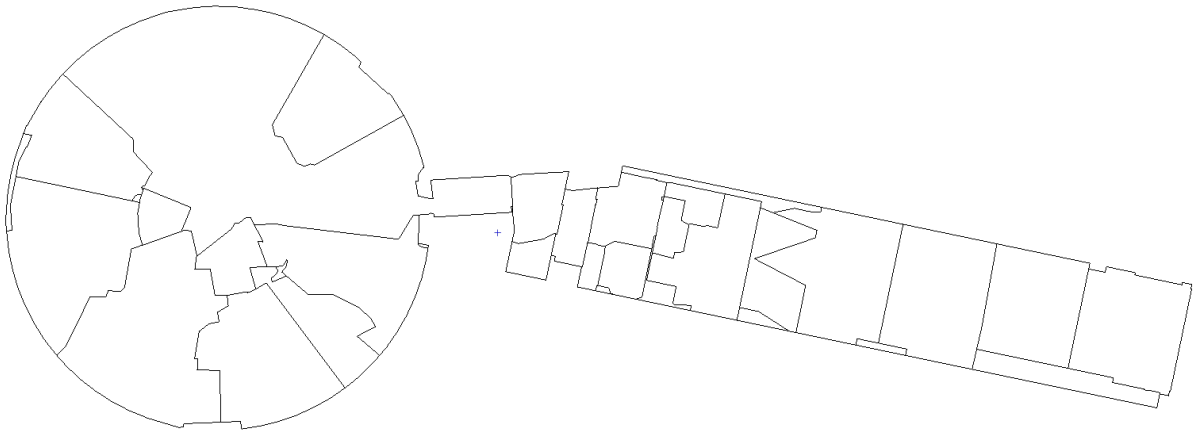


Figure 4.34. Plan view of 10mm voxel filter F2050 reconstruction using 26 ground floor scans w/ blinds down

4.6.3. Limitations

The following limitations in using the presented reconstruction results for BEMS applications were identified;

1. Using only ground floor scans, it was expected that any BEMS model would exhibit a reduced volume and floor area compared to the base line model.
2. The use of window blinds during the survey required BEMS openings in thermal boundaries to be added back in manually.

3. The lack of a reconstructed internal thermal mass (e.g. internal partitions) meant that any BEMS assessments were likely to predict a higher rate of heat loss from the full thermal envelope.
4. To create these geometry reconstructions had taken approximately 2 days which included the site survey, scan registration, pre-processing of E57 files and the automated reconstruction. The reconstructed geometry still required the application of boundary conditions and the identification of openings (e.g. windows, doors, etc.), see Chapter 6.

4.7. Summary

This Chapter described the proposed new workflow to automatically produce geometry reconstruction of buildings. Typical Point Cloud outputs were presented including features of captured Point Clouds to be aware of when used in this format such as noise generated via reflections, blind spots on the scanner and photos taken alongside the laser scan.

A method of Point Cloud registration was documented along with an overview of existing modelling capabilities of commercial Point Cloud processing software and tools available within the academic sphere. An existing prototype was reviewed that supported the use of E57 files and utilised a GUI to display reconstruction results for exporting in an IFC format. The geometry reconstruction process and pre-processing of an E57 file was described. The subsequent processing of an input E57 file was summarised with full details provided in Appendix E. Additional developments were identified as the definition of thermal surfaces and the development of an output format in gbXML.

Following initial reconstruction results attempts were made to reduce survey noise by lowering window blinds on a second site survey and a subset of ground floor scans was selected to reduce the total number of input data points that required processing. This revised method produced recognisable F2050 geometry however some discontinuities existed. A visual improvement in geometry reconstruction was achieved following the application of a voxel filter.

Limitations to the reconstruction method were identified as being limited to a single storey of a building and access requirements during the survey meant some areas were omitted from data capture. In addition, scanning suspended ceilings, did not capture the geometry

of the building roof accurately. Developing the technique to use external scans would capture the full external building envelope. An investigation into the required hardware for external scans identified substantial cost barriers however it was identified that a scanning prototype could be produced at a relatively low cost which was considered first in Chapter 5 before considering the results of BEMS on the reconstructed geometry presented in this Chapter.

Chapter 5.

Hardware Development

5.1. Introduction

In response to the identified substantial cost barriers for performing external LiDAR scan surveys of buildings, this Chapter documents the development of a low-cost LiDAR unit. The challenges posed by environmental clutter and line-of-sight obstructions, when reconstruction of the thermal building volume from internal scans was attempted, have been summarised and used to identify four design requirements for a prototype of a low-cost LiDAR unit as has been described in subsection 5.2.

The Arduino hardware platform and LiDAR sensor used in the hardware design are detailed in subsections 5.3 and 5.4. The design development that enabled rotational LiDAR measurements is described in subsection 5.5. Wireless communication hardware, a portable power supply and data logging functionality were all incorporated onto the prototype as described in subsection 5.6. Incorporation of an IMU for positional tracking is presented in subsection 5.7.

Three primary error sources were identified to aid future development of the prototype in subsection 5.8. An assessment of the prototype costs has been provided in subsection 5.9. The Chapter concludes by summarising the findings of the research into hardware development in subsection 5.10.

5.2. Low-Cost LiDAR

A clear observation made in all of the reconstructions was that when scanning internally, in the presence of cluttered environments and obscured building features, the reconstructions included occlusions. Although voxel filtering was found to help reduce noise, occlusions still existed in the geometry reconstruction. On top of this, as was found during the laser scan surveys described in subsection 4.2, not all spaces within a building volume were accessible during the surveys. This presented the issue of missing volumes from the reconstruction processes.

One solution to this problem would be to conduct external scanning of a building instead as this would have ensured the full thermal envelope was captured. Ideally this would have been conducted using a UAV which was capable of capturing the roof geometry accurately unlike

a terrestrial static scanner. Roof geometry and height were identified to be missed in some cases when scanning internally due to the presence of suspended ceilings, see Figure 5.1.

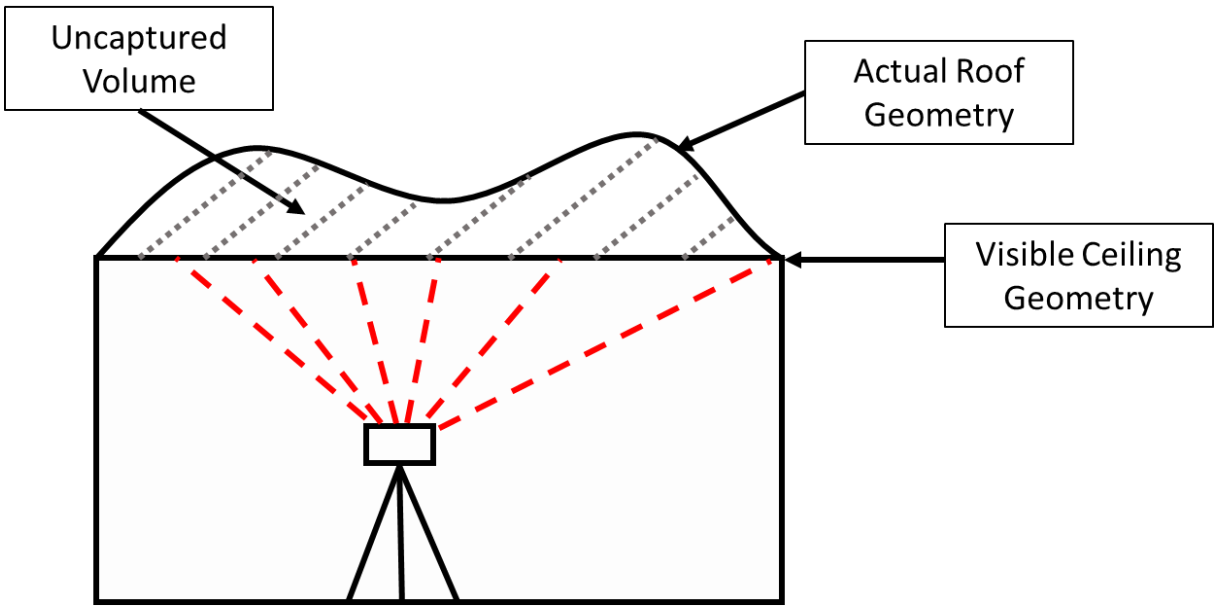


Figure 5.1. Internal scan limitations

A UAV would also have been able to capture the building in a single uninterrupted survey during flight. This would have resulted in a reduced time to conduct the scan to BEMS workflow by removing the need for manual scan registration and it would also have almost removed disruptions to building occupants during the survey.

As discussed in Chapter 1 and Chapter 2, the capital required for UAV mounted LiDAR systems was found to be substantial. However, an opportunity was identified to develop a low-cost LiDAR scanner and an initial prototype was produced and has been documented within this subsection.

In order to achieve this objective, there were four main requirements outlined for the prototype;

1. A prototype shall have an integrated LiDAR sensor for distance measurements from a fixed point in a survey volume.
2. A prototype shall have been capable of distance measurement, via LiDAR, in a 360° field of view.
3. A prototype shall have been able to record its movement in three dimensions to track the centre of rotation and been remotely controlled.

4. The cost of prototype hardware materials shall have been acquired for under £500 which was orders of magnitude less than existing market offerings.

5.3. Hardware Platform

To develop a working prototype quickly and inexpensively, without needing to create the central data processing hardware and architecture, the Arduino platform was selected [166]. It was inexpensive, open source, widely used for electronic hardware projects and required limited knowledge of electronics to start development. The Arduino platform was programmable and allowed for applications that used a wide range of electrical sensors and actuators. To begin prototype development an Arduino Uno board was selected, see Figure 5.2.

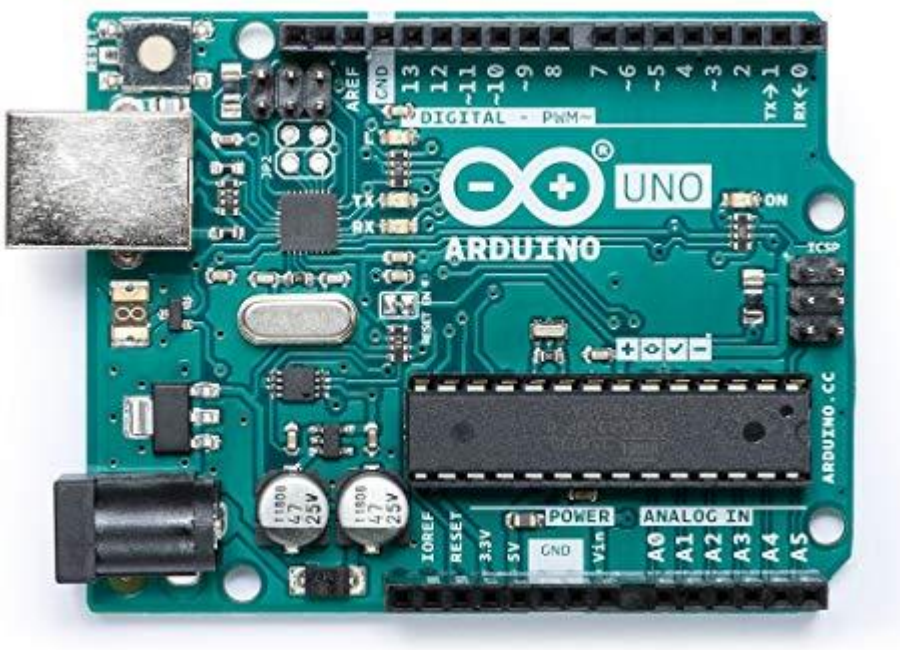


Figure 5.2. Arduino Uno R3 board [167]

To use the Arduino Uno connections had to be made between all hardware and the board using the digital and analog pins. Once the correct hardware had been installed programme code was uploaded to the ATmega328P microcontroller. The code was written using the Arduino language which was effectively the same as the C++ language. The code upload was handled by the Arduino Integrated Development Environment (IDE) which passed the C++ code into a compiler that converted it into binary instructions for the microcontroller.

The first part of any Arduino code defined a list of variables and their values at programme initialisation. This was used to import C++ libraries, define global variables and assign Arduino pins to perform input or output functions.

When first powered, or reset, and the global variables had been defined there were two main functions that ran in the microprocessor. The first function run was called $setup()$ and was a $void$ type function that executed a list of commands but did not return any result on completion. This function was only run once for configuration purposes and was not run again until the board was manually reset.

On completion of the $setup()$ function the $loop()$ function was run. $loop()$ was also a $void$ function type that executed commands but did not return a result. The commands it executed can however have return values contained within the $loop()$ function. Once the $loop()$ function had been completed it was continuously repeated until either power was removed from the Arduino or the board was reset manually. From this basis the Arduino was capable of being programmed to send, receive and process signals from a variety of devices that were used for data logging across many applications.

5.4. LiDAR

Based on a review of existing and commercially available LiDAR sensors the LiDAR-Lite v3 High Performance (LL3HP) sensor was selected for use in prototype development [168], see Figure 5.3. The sensor had a range of $40m$ which removed the need for any UAV to fly near a façade during a survey. The sensor had a low power consumption, its communication interfaces were compatible with an Arduino based prototype, its IPX7–rated housing protected it from dust and moisture ingress if working in an external environment and when operated at its quickest acquired data at a rate of $1.5kHz$.

Figure 5.3. Garmin LL3HP [168]

Full specifications of the LL3HP are included in the operating manual [169].

The two methods available for communicating between the Arduino Uno and the LL3HP were either I2C or PWM. I2C is a bus communication protocol that allowed for multiple devices to communicate using two data lines only; Serial Data Line (SDA) and Serial Clock Line (SCL). The devices on the SDA and SCL lines acted as either master devices or slave devices. Master devices are used to synchronise a communication clock and initiate communication to other devices on the bus. In comparison, slave devices are used to respond on the request of master devices. In theory, the number of master devices was unlimited but steps would have to have been taken to avoid conflicts between multiple master devices attempting to communicate at the same time, and the number of slave devices was limited by the address space. Each slave was assigned an address that is unique to the slave device. The first small data package sent by any master device included the intended slave address so the correct device would respond.

PWM uses pulses of a digital signal to generate an analog signal. When PWM was used the output voltage was varied by adjusting the time that the digital signal was set to high, which taken as a percentage of a single digital switching cycle defined the duty cycle of the PWM output signal. The second parameter to fully define the PWM output was the frequency of switching the digital signal from high to low. The PWM output behaved as a constant analog voltage to power devices. Being an analog signal any device connected to the PWM output would have responded as it is not address specific like I2C.

In the context of LiDAR prototype development I2C was selected. The I2C communication interface enabled the addition of other devices that were needed to meet prototype development requirements. This also ensured that the devices were synchronised to the same clock during communications. The I2C communication for the LL3HP supported $400kHz$ data transfer speeds which was much greater than the maximum acquisition rate of the LL3HP (i.e. $1.5kHz$). A prototype wiring diagram of the LL3HP to the Arduino has been illustrated in Figure 5.4. The connections included $+5\mathcal{V}$ in red, Ground (GND) in black, SDA in blue and SCL in green. The yellow and orange wires of the LL3HP were not required to use I2C. As per the manufacturer's instructions for the I2C configuration, a $680\mu F$ capacitor was placed between the $+5\mathcal{V}$ and GND lines. This ensured a level voltage was applied to the LL3HP to account for any fluctuations in the $+5\mathcal{V}$ line.

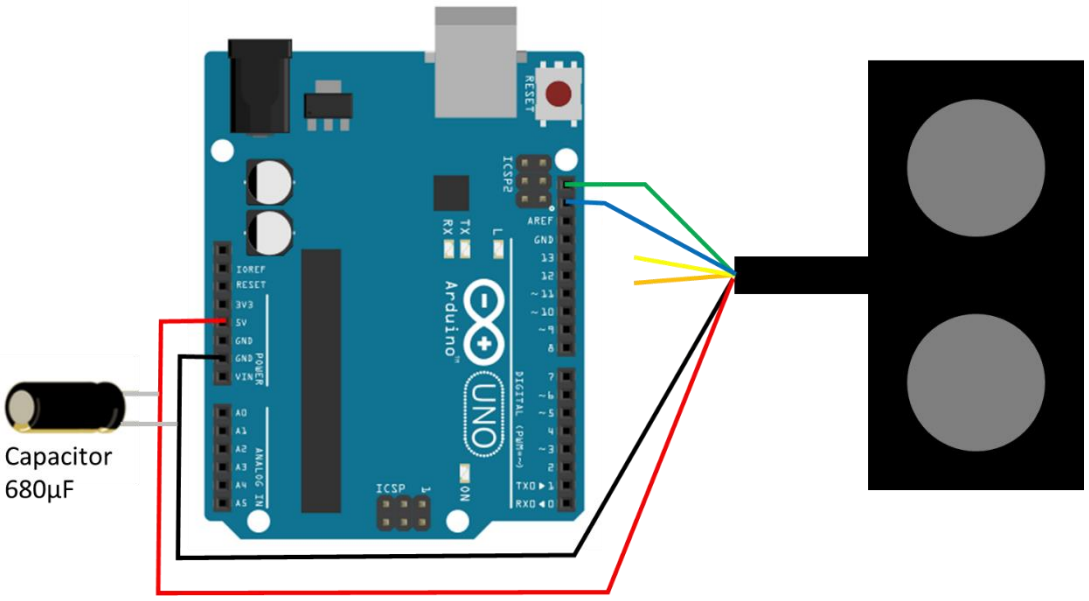


Figure 5.4. LL3HP connection to Arduino Uno

In this configuration the Arduino acted as a master on the I2C interface and the LL3HP acted as a slave device. When powered the LL3HP transmitted an infrared laser beam which when reflected back from a surface was detected at the LL3HP. The time of flight involved in this signal travel was used to determine the distance of a surface away from the sensor providing a One Dimensional distance reading. With a programme written and uploaded to the Arduino, using the IDE, these measurements were viewed on a computer. This development met the first requirement of prototype development. Testing of the sensor was documented upon incorporation of rotational functionality, see subsection 5.5.

5.5. Rotation

The next step in development was to expand the LL3HP capabilities so that it rotated producing a 360° view of measurements in a 2D plane. This was conducted by creating a 3D printed support frame that integrated two pulleys joined by a rubber O-ring. The smaller pulley was driven directly by an electric motor mounted to the frame while the larger pulley was positioned on top of a bearing and slip ring. This configuration, and using a Slip Ring, allowed the LL3HP to rotate freely without twisting the cables connected to the Arduino Uno. A cross-section of this arrangement, designed as part of this research, has been provided in Figure 5.5.

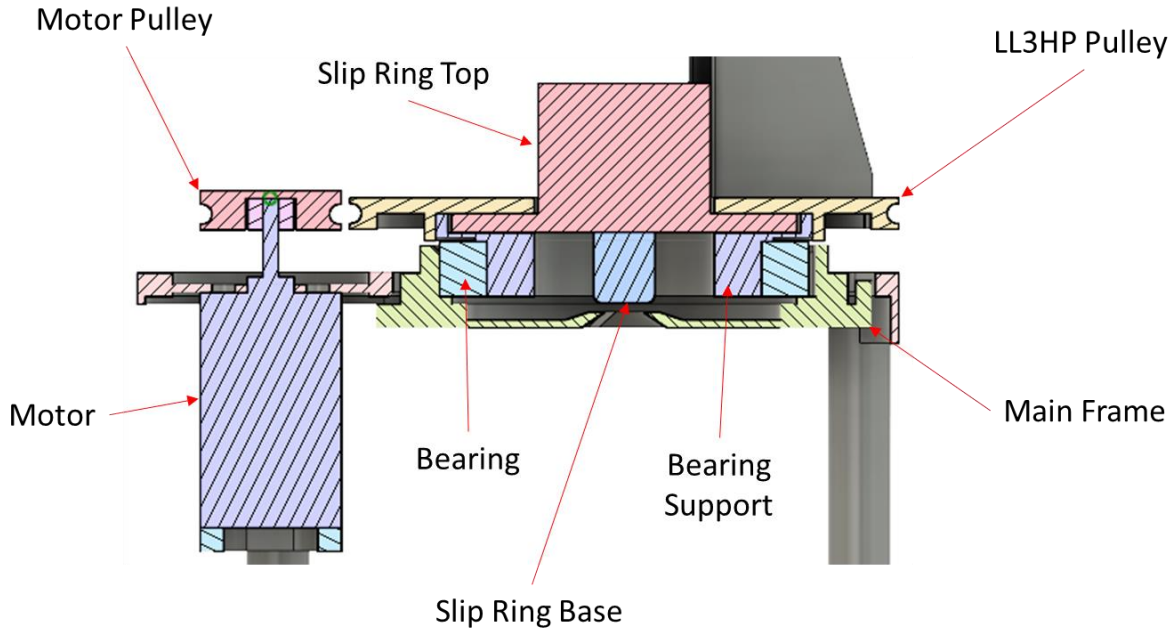


Figure 5.5. Cross-section of prototype rotation mechanism (O-ring removed for clarity)

The LL3HP Pulley incorporated a mounting stand for the LL3HP so that it could be fixed to the pulley with 4xM3 bolts, see Figure 5.6. The position of the LL3HP ensured its cable was centred over the slip ring to make the physical electrical connection easier.

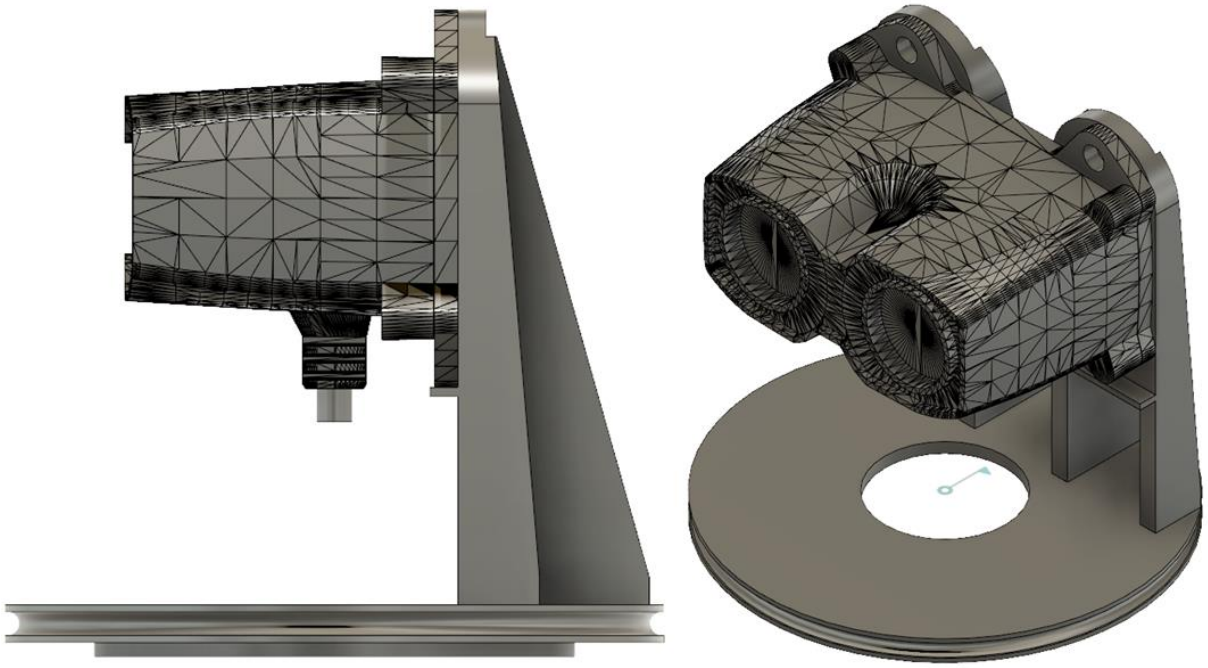


Figure 5.6. Mounting frame for LL3HP

5.5.1. Angle Measurement

Having achieved free rotation of the components it was important that the angle of the LL3HP, relative to a fixed datum on the prototype, was measured. This was achieved using an optical sensor embedded into the LL3HP pulley and an encoder ring which was positioned on top of the Main Frame and below the LL3HP pulley as demonstrated by Grau [170].

The encoder ring, illustrated in Figure 5.7, consisted of 14 standard steps and a single smaller datum step. The datum step was half the length of a standard step making the intervening gap between the datum and first standard step 1.5times greater than other encoder steps. The encoder ring was assembled to the frame using keyways to ensure correct positioning relative to the Main Frame.

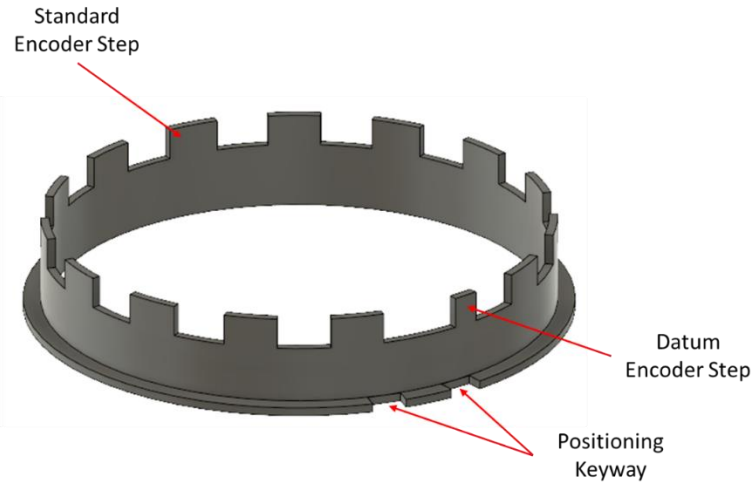


Figure 5.7. Prototype encoder ring

An optical switch was embedded into the LL3HP Pulley that rotated with the LL3HP. The switch was positioned so that its transmitter and receiver were situated on either side of the encoder ring. As the switch was blocked intermittently by the encoder ring steps, an alternating signal of high and low voltage was produced. Assuming a constant rotational speed of the LL3HP the time taken to travel between the datum encoder step and the first standard encoder step differed in comparison with the travel time between two standard encoder steps.

By timing the cycles of this alternating signal, a determination was made as to when the datum had been triggered and an angle measurement of 0° was produced. Each time the optical switch was triggered the angle measurement was incremented by 12°. If a request for the current angle was made during the middle of a transition of an encoder step or gap the timing of the previous transitions were used to estimate the correct angle at that time. At 360° the datum was triggered again, and the angle was reset to 0°.

To achieve this functionality an Omron Transmissive Photo Micro-Sensor EE-SX1320 [171], see Figure 5.8, was incorporated into the LL3HP Pulley as described above.

Figure 5.8. Omron transmissive photo micro-sensor EE-SX1320

The sensor had a gap of $2mm$ between the transmitter side and the receiver side which when powered had an optical signal sent between each side. The sensor was designed to be mounted on a Printed Circuit Board (PCB) using its four soldering pads, see Figure 5.8. At this stage of prototype development, as no PCB was available, jumper wires were soldered onto the sensor, which was glued into place on the LL3HP Pulley, see Figure 5.9. In this position the encoder ring intermittently blocked the signal in the optical switch as the LL3HP Pulley rotated, see Figure 5.10.

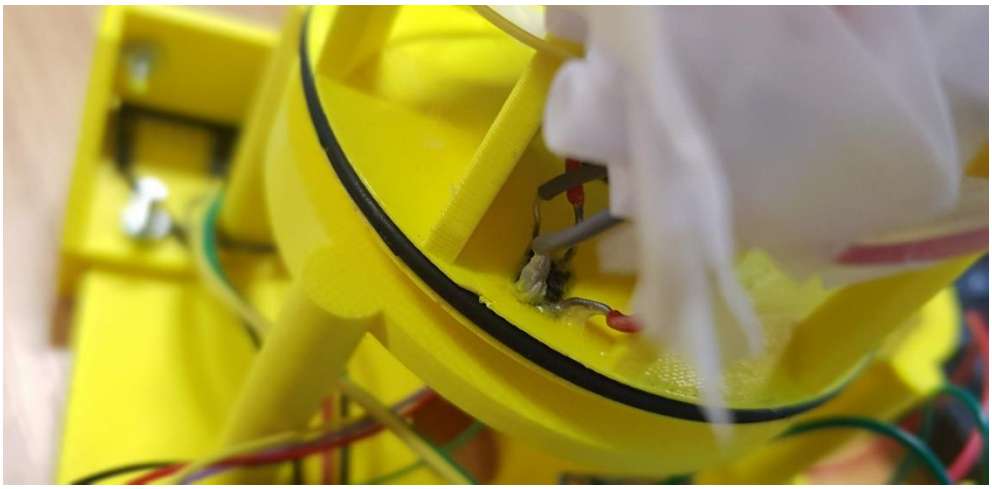


Figure 5.9. Jumper wires soldered onto optical switch and glued into LL3HP Pulley

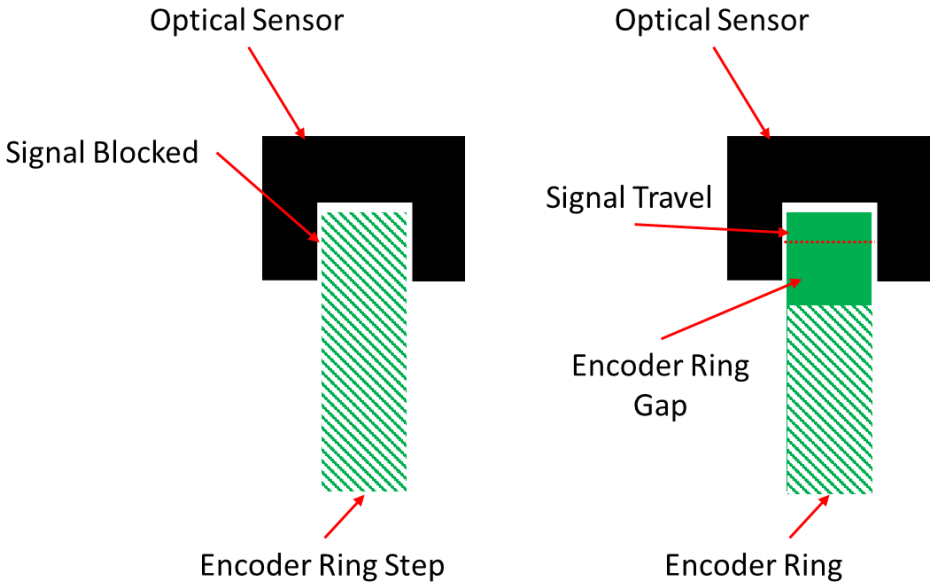


Figure 5.10. Cross section of encoder ring interacting with optical switch

5.5.2. Motor Control

The rotation of the LL3HP was powered by a $6\mathcal{V}$ DC motor that was supplied with a 0.5module spur gear, that had 10 teeth, and was fixed to the motor shaft. The motor pulley was designed for the spur gear to be recessed inside so that the planes of rotation of both pulleys were in alignment.

It was necessary to include a method of motor control in the prototype so that start and stop commands could be used when required. The effect of back-Electro Magnetic Force (EMF) also required consideration. This effect has been observed to occur when the supply voltage from a motor has been switched off and the motor decelerates. During this time, with the armature in motion, the back-EMF has been found to not be exceeded by a supply voltage which has then caused a small reversal of current in the circuit. While this only occurred for a small amount of time repeated current cycles can cause damage to microcontrollers, such as those on the Arduino.

To prevent this damage an L293D microcontroller was incorporated into the prototype. It consisted of 16pins and had the potential for driving a maximum of two motors independently. One end of the microcontroller had a circular dent to enable pin identification, see Figure 5.11; the pins used by the prototype have been highlighted in red.

The L293D contained two H-bridges that were used to control the direction of any attached motor and protect against short circuits during switching through the use of diodes.

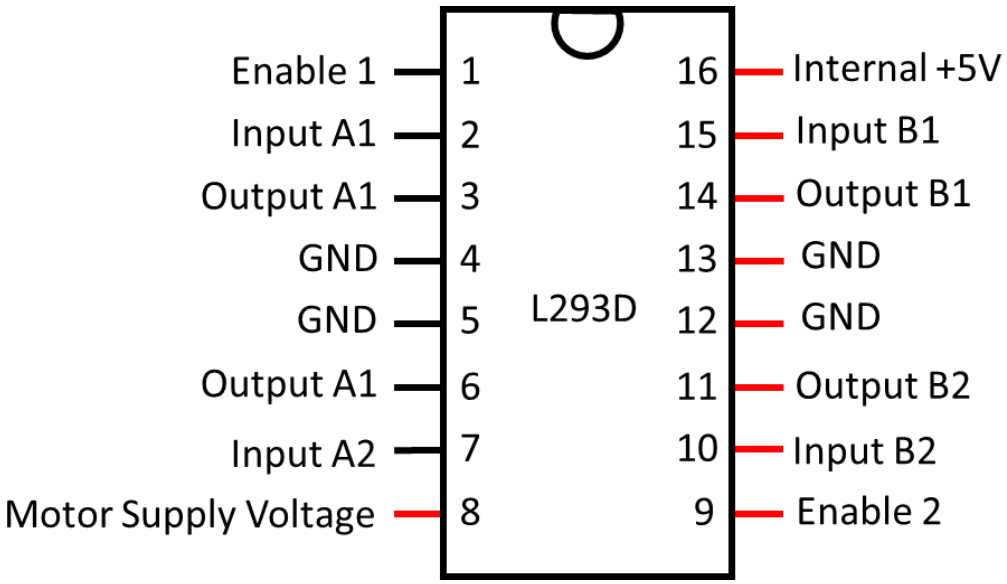


Figure 5.11. Pin layout of L293D microcontroller

Voltage was supplied to pins 8 and 16. A voltage supply on pin 8 was used to drive the motor. A voltage supply on pin 16 provided $+5V$, from the Arduino, to power the internal logic gates of the L293D microcontroller. Pins 11 and 14 were connected directly to the terminals of the motor used in the prototype. Pins 12 and 13 were connected to GND.

The final three pins; 9, 10 and 15 were used to control the logic to the motor. For any motor operation to occur pin 9 needed to be set to high and this was used to turn the motor on and off. The pins 10 and 15 determined the rotational direction of the motor and the pin values opposed each other (e.g. when pin 10 was high, pin 15 was low) for rotation to occur. If the values of both pins were reversed, then the motor changed direction. For development of the prototype only a single rotation direction was required as this ensured a constant datum position with the encoder ring.

All of these features were then combined to achieve rotational functionality of the LL3HP, and the prototype wiring diagram has been provided in Figure 5.12. Resistors were added to the optical switch circuit to ensure the correct logical values are provided to the output wire which returned to the Arduino. Power for this prototype configuration was provided via a Universal Serial Bus (USB) cable that also enabled code to be uploaded onto the Arduino microprocessor.

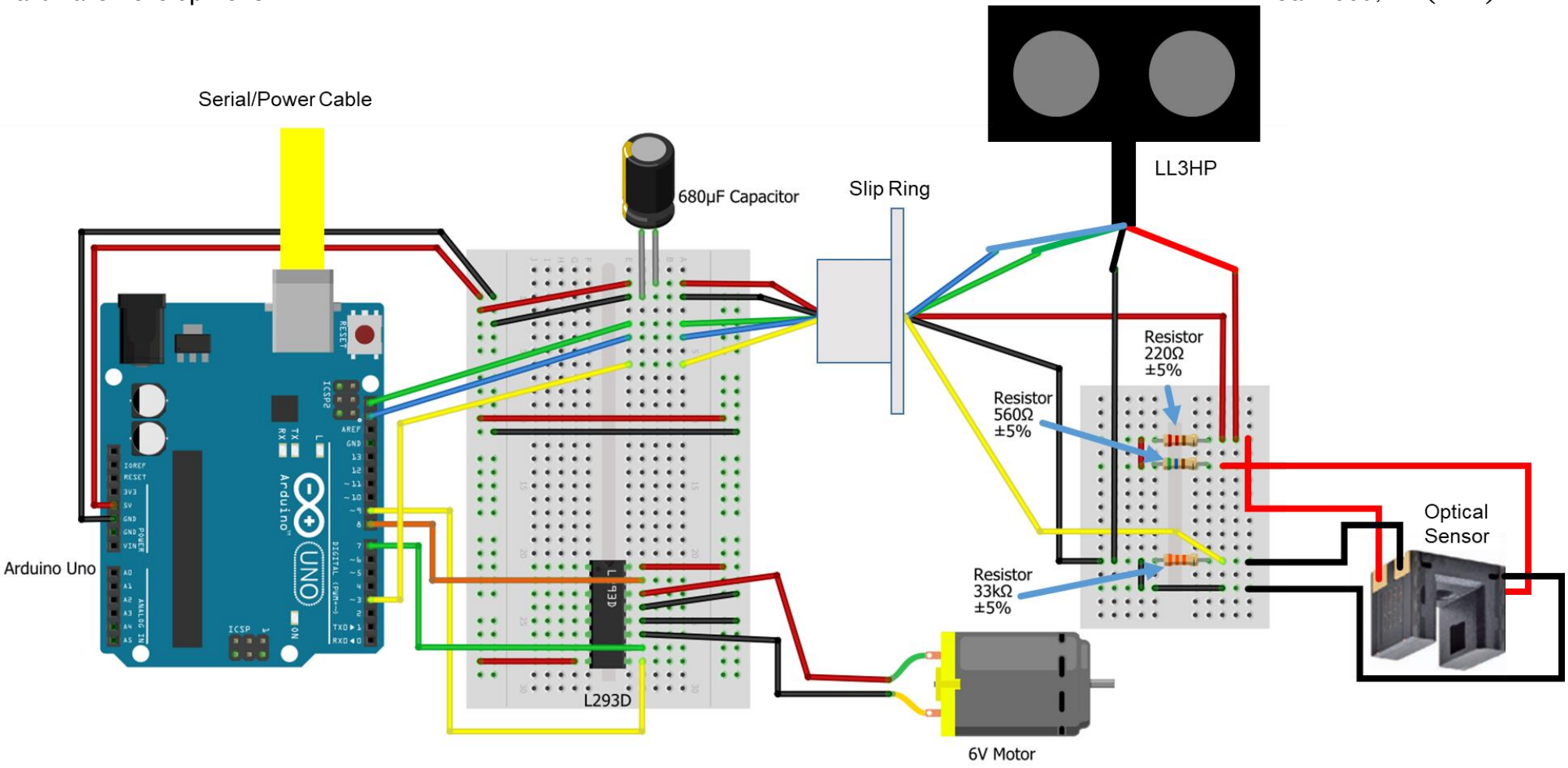


Figure 5.12. Rotation functionality prototyping circuit

Figure 5.12 identified the use of two prototyping breadboards. The smaller breadboard that was used for sending power and signals to the LL3HP and the optical sensor was actually a terminal connection block on the developed prototype. With the code written and uploaded to the Arduino the LL3HP was capable of taking measurements in a 360° 2D plane. With the Arduino tethered to a computer via cable the distance and angle measurements from any readings were streamed quickly for display purposes.

5.5.3. Calibration

Before the prototype was tested the interaction between the encoder and optical switch underwent a calibration. Each time the optical switch changed its output state (i.e. blocked→uncovered or uncovered→blocked) the software uploaded onto the Arduino performed an Interrupt Service Routine (ISR). An ISR is used to respond to an external event by momentarily interrupting the main software. In this application the external event was a change in state of the optical switch.

During the ISR the optical switch output was read to identify if it was in a high or low logical state and the time that the ISR was triggered was recorded. This was compared to the time of the previous ISR call to compute an encoder transition time between optical switch states. As each encoder step covers 12°, the value for the encoder angle was incremented to identify which step or gap of the encoder the switch was currently transitioning across. Based on the geometry of the encoder ring the shortest transition time occurred on the smaller datum step. This transition finished as the optical switch moved from a blocked to uncovered state. For this reason, the time to transition across a step was recorded as a short time step with the time across a gap as a long-time step. This was updated every time that the optical switch state changed.

When a gap between encoder steps had been transitioned the ratio between the existing shorter time step and the newly calculated longer time step was computed. This ratio was used to determine whether the datum of the encoder had just been triggered. When the datum had been triggered the measured prototype, angle was set to 0°. If the datum was not triggered the measured prototype angle was set to match that of the encoder angle and a timer was updated based on the latest step transition time that incremented the angle by 1° each time the timer was looped.

Figure 5.13 provided an algebraic overview of the distance between each of the encoder steps. Under the assumption that the LL3HP was rotating at a constant speed the theoretical encoder ratio when the datum point was reached was calculated as $^1/_3$. For all other ratio calculations across regular sized encoder steps and gaps the ratio was theoretically 1.0.

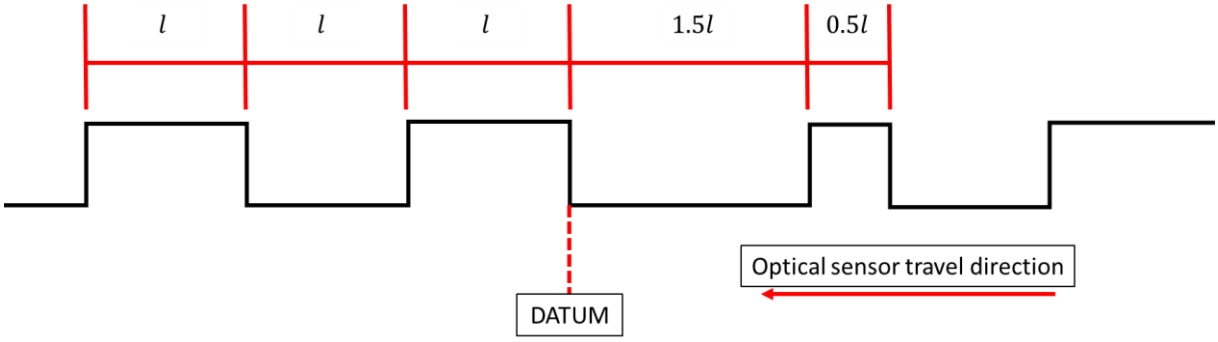


Figure 5.13. 2D layout of encoder ring

In reality, there were several variables that affected the smooth rotation of the LL3HP Pulley which included the bearing friction and any slippages between the rubber O-ring and the driving and driven pulleys. This variation caused, in some circumstances, the datum encoder ratio to be greater than 1/3 and the standard encoder ratio to be less than 1. This led to a requirement to calibrate the encoder ratio at which a datum point was declared.

During calibration the datum encoder scale was tested ranging from 0.1 to 0.9 at 0.1 increments. If the ratio was calculated as less than the calibration value, the datum was declared. Two separate tests took place; 1) Each encoder step transition was timed which adjusted the rate at which the measured angle was incremented by 1°, and 2) the same timing was used as per test 1 but an additional correction was made at each trigger of the optical switch to match the measured angle with the encoder angle. The resulting distance measurements and angles for each calibration test were plotted in 2D and visually compared to identify the most appropriate operation method and encoder scale. The results of the calibration test have been tabulated in Appendix G.

The second calibration test, presented in Appendix G, was only carried out for encoder ratios between 0.3-0.8. The results of the first test outside of this range produced spurious results with no visual identification of room boundaries possible based on timing alone so a meaningful correction in the second test was not expected.

Across each of the encoder scales 0.3 to 0.8 room boundaries were detected and plotted during the scan. These room boundaries tended to be fainter when a lower ratio was used and more defined the higher the ratio was. The encoder scales of 0.6 and 0.7 were seen to produce the strongest visual results. At 0.7 and 0.8 a skipping phenomenon was observed where wall boundaries were ghosted in the scan noise but offset at regular angles around the scan origin.

From the calibration tests it was observed that the second 12° correction had the effect of smoothing out the results when compared against the use of timing only, in the first test, to determine the measured angle. This had the benefit of regularly correcting for any mistiming between encoder steps due to variation in the motor rotational speed. From these results an encoder scale value of 0.66 was selected to distinguish between datum declarations and regular encoder step transitions.

5.5.4. Testing Rotation Functionality

On completion of calibration a lightweight GUI was created that allowed for operation of the prototype from a computer and a 3D display of a scan. The GUI contained a central graphical display, complete with directional axis, to render recorded points from the prototype in real-time. A message console was displayed at the bottom of the screen and the upper left-hand corner contained interactive buttons providing the ability to start the scan process, stop a scan, start a subsequent scan of a single survey, save the scan data and exit the GUI.

To test the rotational functionality, the prototype was mounted onto a tripod with the LiDAR sensor placed horizontally. This enabled the prototype to record a 2D horizontal slice of the environment surrounding the scanner, see Figure 5.14.

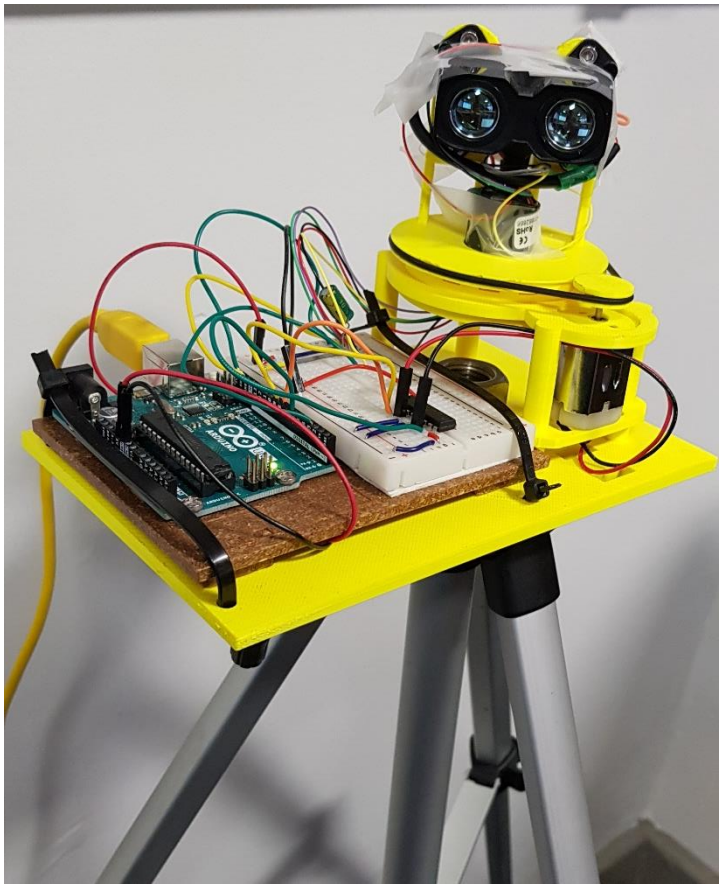


Figure 5.14. Prototype mounted onto tripod

This test was conducted in an indoor lab space. Two tests were conducted; 1) a single static scan from one location, and 2) two scans in total from different locations in the lab as indicated on Figure 5.15.

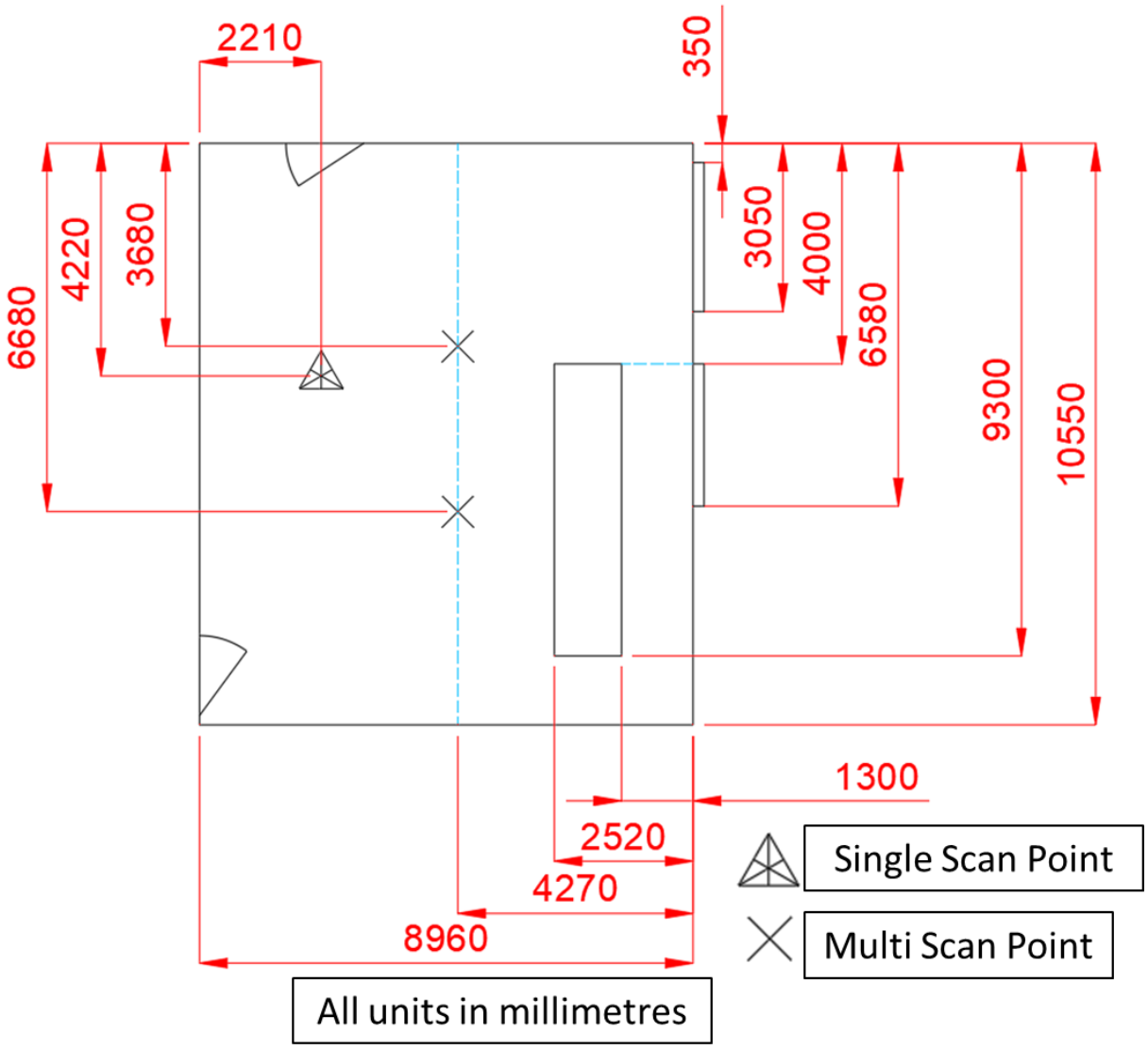


Figure 5.15. Plan of lab for rotational functionality testing

The first static test produced a 2D map of the testing lab, see Figure 5.16. The red lines manually added over the right of this figure have highlighted where the room boundaries have been detected by the scanner.

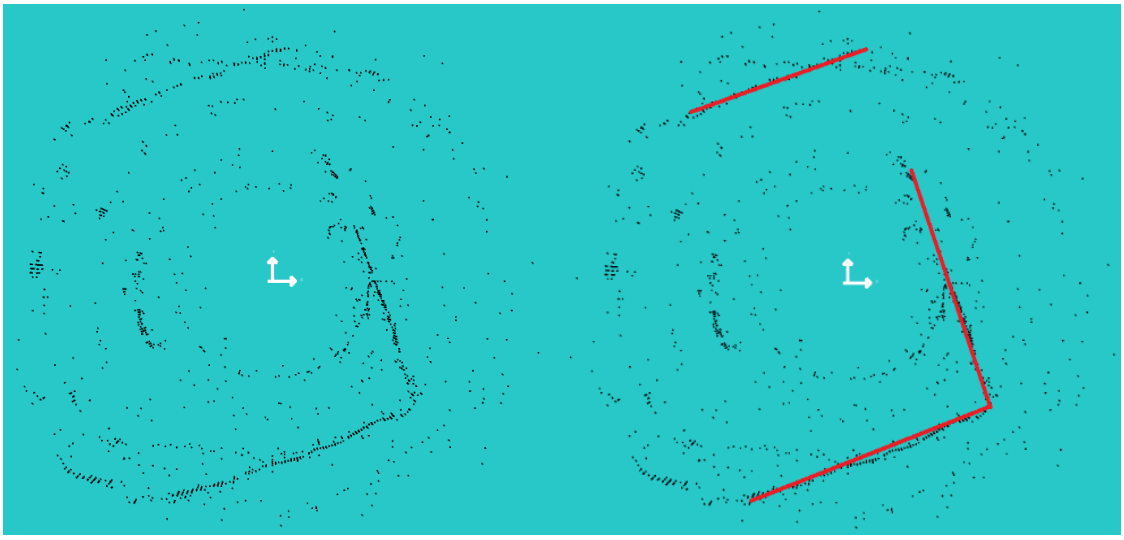


Figure 5.16. (left)Test scan results for single static scan (right) identification of room boundary

Based on knowledge of the scanner position in the room, and size of the room, the accuracy of the prototype was determined. The scanner readings were only interpreted sufficiently to detect two opposing walls in this test. The linear distance between the walls was interpreted by the prototype as $10.54m$. Compared with the known room length of $10.55m$ a discrepancy of approximately 1% was calculated from this dimension.

It is apparent from the result that although room boundaries are visually identifiable the rendered 2D scan maps included high levels of noise. While noise was expected close observation of these maps showed that recorded data formed in concentric circles around the scanner with radii such that the detected wall boundaries were tangential. This was likely caused by a mistiming of the optical sensor in relation to the encoder and could have been a result of variable motor speeds affecting the timing between encoder steps.

The second test involved capturing scans at two distinct locations within the room, see Figure 5.15. The first of these scans produced the 2D Map shown in Figure 5.17. This scan was conducted in a different location and prototype orientation to the first test but the same room boundaries were detected.

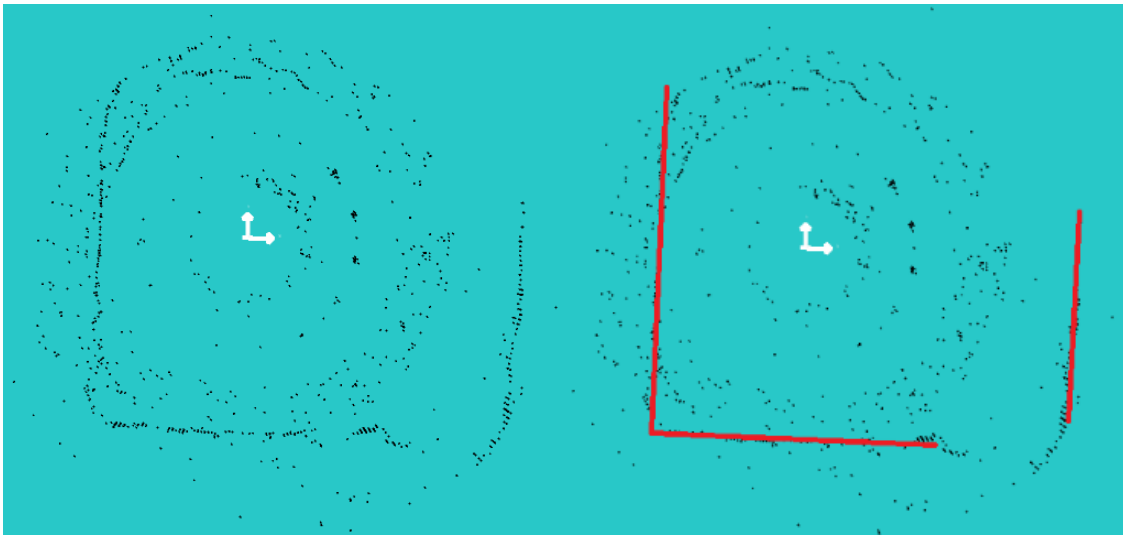


Figure 5.17. (left) Test scan results for 1st of two scans (right) identification of room boundary

On completion of the first scan, the scanner was moved to a second known location in the lab. Scanning was conducted at this second location and the results were manually offset by the distance moved between scan locations. This overlap of two scans enhanced the scan results allowing identification of all four walls. A discrepancy in linear room dimensions was calculated as between 0.29-1.2%.

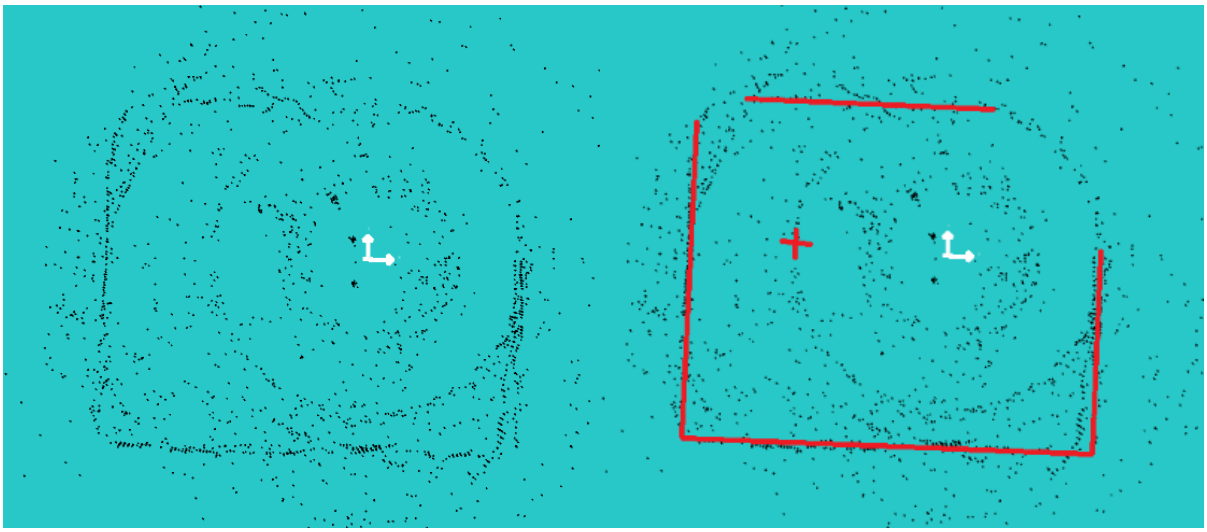


Figure 5.18. (left) Test scan results of two scans (right) identification of room boundary and 1st scan position

The data recorded by the prototype consisted of a distance reading in cm and a direction angle from the prototype datum. This information was displayed in the message console during the scan. On completion of a scan the data containing the raw readings from the prototype, as well as the interpreted coordinates from the scan location, were saved in two

separate text files. Note: The coordinates were produced based on a 2D scan so the third coordinate was set to 0.0 for all readings.

Observation of these outputs indicated that not every prototype reading was written to a matching coordinate and as a result some scan data was lost from final results. This was caused by the prototype being capable of reading scan data quicker than the GUI could process and render it. One method to avoid this data loss was identified as recording scan data directly onto portable media to then be converted into coordinates when data underwent post-processing after the scan.

The development of the 2D scanning prototype described thus far met the second requirement of prototype development by enabling a 360° scan in a single 2D plane. This plane was orientated horizontally but to scan an environment in 3D it was identified that a vertical 2D plane would be better suited. If translation and orientation of the scan origin was tracked, then the 2D scanning plane could be moved through an environment. Thus a 3D scan, consisting of several 2D slices, would have been constructed. It was also observed that in a 2D scan any mistiming of the optical sensor, that created noise/erroneous readings, displayed all noise on a single 2D plane. This meant that as the scan time increased more erroneous points would be produced in a single plane. By moving to a 3D scan configuration it was predicted that these erroneous points would be distributed through a larger volume and as such may not have as much prominence in a 3D data set.

5.6. Mobility

Following prototype development to produce rotational functionality the limitations for the prototype to be used in a mobile manner were identified. The rotation prototype was tethered to a computer which provided the ability to power, upload Arduino code and stream data live during a survey but made moving around impractical. A method of portable power and communication was required. As previously mentioned, several data packets recorded by the prototype were lost when rendered on screen and a means of on-board data recording was needed. The prototype was only capable of individual static scans offset by a known amount. If the translation and orientation of the prototype was recorded it would have allowed the scanner to move freely during a survey to capture a full environment which would be much more suited to UAV surveys.

5.6.1. Power Supply

The Arduino operated on $5\mathcal{V}$ input and the prototype motor was rated at $6\mathcal{V}$. These two components of the prototype operated in parallel. To accommodate these power requirements it was determined that a $12\mathcal{V}$ power supply would be sufficient. Eight $1.5\mathcal{V}$ AA batteries were inserted into a battery box that incorporated a plug PP3 electrical connector. A battery enclosure was 3D printed and bolted to the prototype that included a locking piece to keep the batteries in place, see Figure 5.19.

Figure 5.19. Portable power supply for prototype

Using a common power supply for both the Arduino and motor meant that both used a common GND. As the Arduino had sensitive integrated circuits it required protection against any inadvertent back voltages created when operated. For this reason a diode, 1N-4007, was added between the battery +$12\mathcal{V}$ line and the $\mathcal{V}_{in}$ pin on the Arduino so that current only flowed in a single direction and any delicate electronics were protected.

With a $12\mathcal{V}$ input applied to the Arduino at the $\mathcal{V}_{in}$ pin its internal regulator created a $5\mathcal{V}$ supply to operate the Arduino and associated connected peripheral components. In parallel, $12\mathcal{V}$ was also applied to pin 8 on the L293D integrated circuit to provide power to the motor when requested by the Arduino logic.

5.6.2. Remote Control & Data Logging

To create a prototype capable of mobile scanning, wireless communication between a computer and the prototype was established. To avoid the data loss observed when survey data was streamed to a computer all measured data was written to an on-board SD card.

This required a simpler text-based user interface for commands to be sent from a computer to the prototype.

One advantage of using Arduino as the hardware platform for prototype development was that standardised modules, known as shields, could be stacked on top of the Arduino Uno to increase its capabilities. For wireless communication a pair of devices; one mobile and one tethered, were required to act as transmitters and receivers of user commands and data. Two Arduino Wireless SD Shields, each in combination with an Arduino Uno were combined for this purpose.

The Arduino Wireless SD Shield plugged into the top of the Arduino Uno maintaining access to the majority of the digital pins used in previous development, see Figure 5.20. The shield contained a set of 20 socket headers that accepted a wireless communication module and a carriage to insert a micro-SD card.

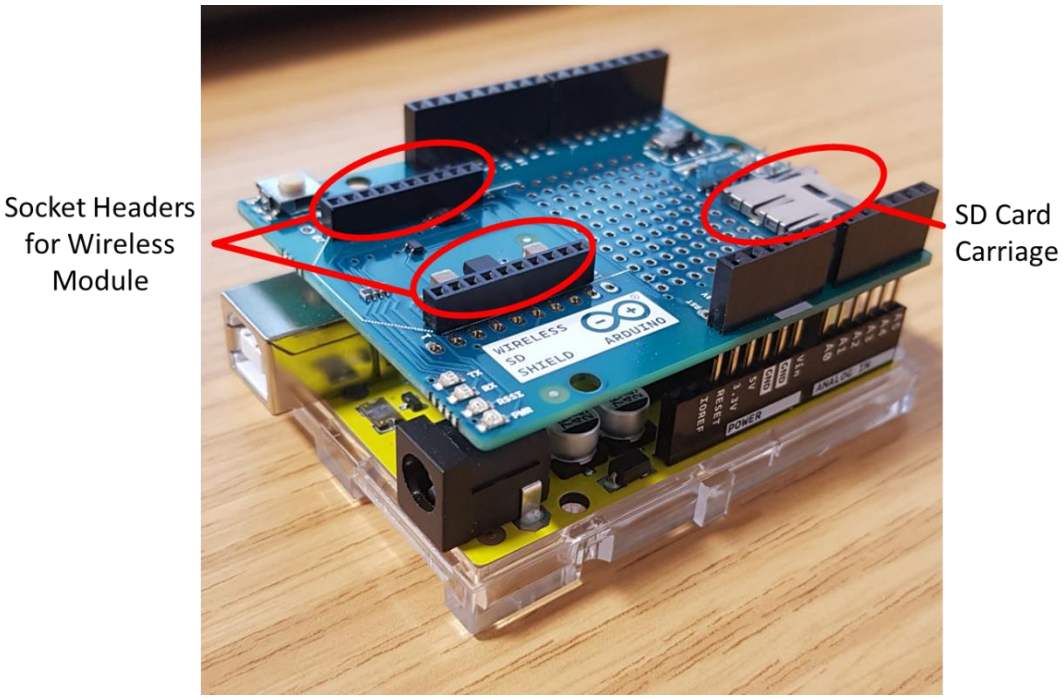


Figure 5.20. Arduino Uno with Arduino wireless SD shield

To achieve two-way communication between a tethered transmitter Arduino and the Arduino incorporated into the prototype an Adafruit XBee Series 1 Module was added into each of the Arduino Wireless SD Shields, see Figure 5.21. This wireless module was able to automatically detect and sync with the same module type when in range. This enabled information to be sent between a computer and the prototype via a serial interface.

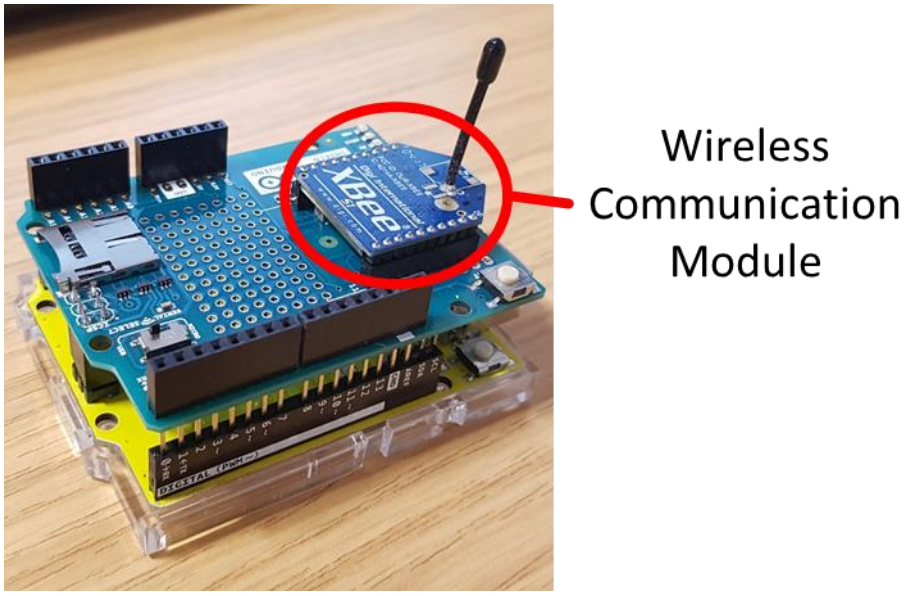


Figure 5.21. Adafruit XBee module, Series 1, 1mW with wire antenna

The code used to conduct logic processing on the prototype was uploaded onto the microprocessor of the Arduino Uno board. When powered the Arduino started running this programme and the first task included confirming an open serial connection between the prototype and the computer. This was performed by sending commands to the tethered Arduino using the Arduino IDE. For this process to be effective the microprocessor on the tethered Arduino was removed, see Figure 5.22, which allowed any commands sent from the computer to pass through the tethered Arduino, straight into the wireless module which was transmitted to the Arduino in the prototype to act on and vice versa.

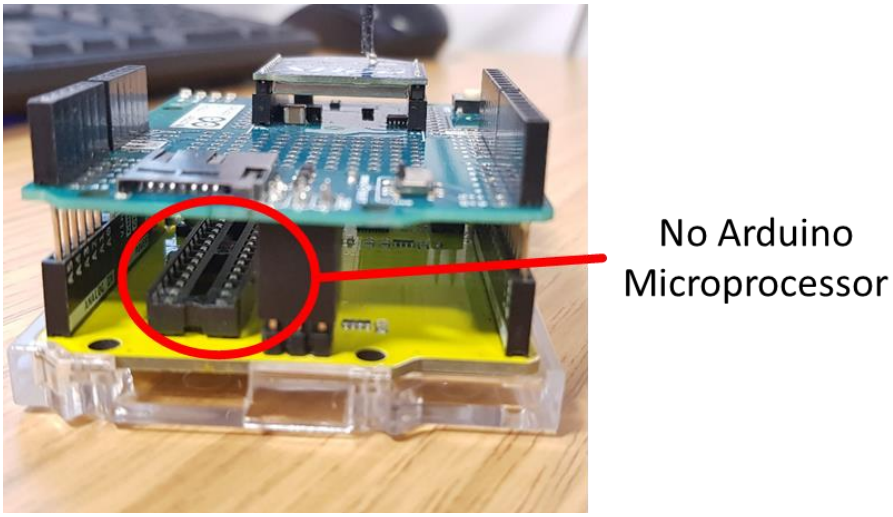


Figure 5.22. Removed microprocessor on tethered Arduino

With the prototype developed to communicate and operate wirelessly it was identified that the recorded data could not be sent at a high enough speed over the wireless connection. Thus, a means of on-board data logging was deployed via use of the micro-SD card carriage on the Arduino Wireless SD Shield of the prototype. This required the use of SD code libraries developed for Arduino to achieve effective read and write functionality. One observation in using the SD library was that approximately 50% of the dynamic memory available in the microprocessor of the Arduino Uno was required for this functionality. While this did not prevent correct operation of the prototype at this stage of development it highlighted that the memory constraints in using the Arduino Uno would become detrimental if too many additional features were used. The prototyping circuit that enabled a mobile prototype is shown in Figure 5.23.

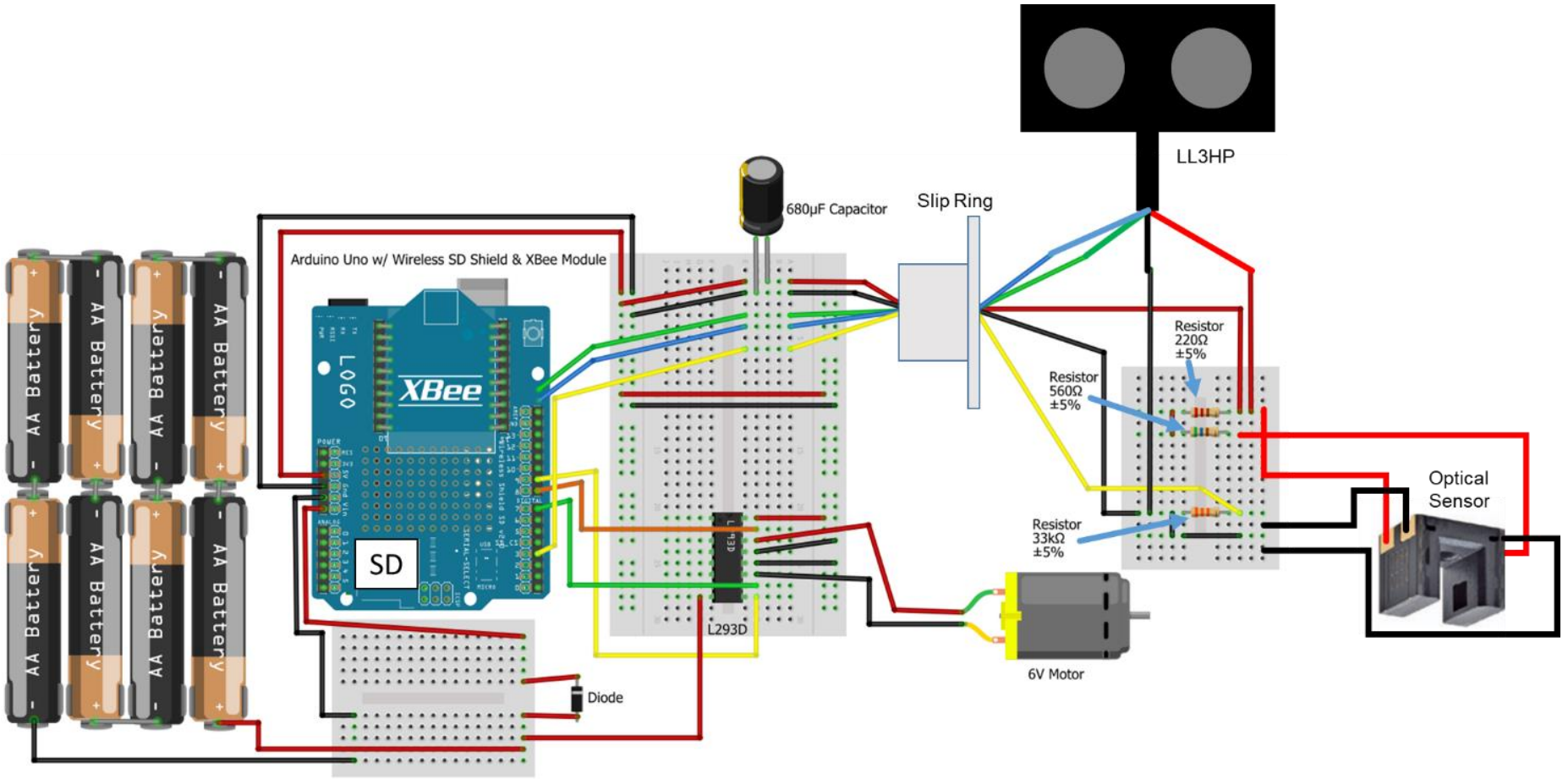


Figure 5.23. Prototype circuit to achieve mobile rotation functionality

Following incorporation of the SD card interface the prototype was operated remotely from the scan origin. With a powered prototype and a tethered Arduino successfully communicating wirelessly, the prototype sent instructions for selecting an operation command; continuous distance measurement, rotating distance measurement or stop a measurement. The operation instruction was then sent over the wireless serial communication link.

This development partially fulfilled the 3rd prototype requirement in enabling remote control of the scanning operations. To meet the full requirement motion tracking of the prototype was also required, see subsection 5.7.

5.7. Motion Tracking

A range of motion detection methods were identified for a small electronic equipment. These were found to range in sophistication and in some cases required substantial infrastructure. Some of these methods included the use of GPS sensors, visual tracking with cameras and on-board motion detection. While GPS was a powerful technology its realistic minimum positional accuracy was found to be approximately $3m$ which increased when indoors [172,173]. On investigation of visual tracking technologies from a base station it was clear that mobility would be limited to line of sight applications and require data to be recorded in multiple locations. On-board motion detection of equipment was found to be widely used in consumer electronics (e.g. smart phones) with a combination of sensors to measure motion and orientation of a device.

The prototype developed for this research was intended for use with UAVs. UAV technology had become sophisticated in recent years with the flight path of a UAV accurately recorded in all three dimensions. Ideally, the motion capture data streams would have been taken directly from a UAV flight recorder, but this data was only available through expensive hardware developer licenses. For this reason, a method of incorporating built-in motion detection was explored for the prototype.

An IMU was identified as a potential sensor that could accurately measure the acceleration, orientation and heading of the prototype while it was moving during a scan. An IMU consists of two or three different types of sensor; 1) accelerometer, 2) gyroscope and 3) magnetometer. The magnetometer sensor is not always present in an IMU. Depending on

the sophistication of an IMU it has individual sensors on a number of axes. For example, a 9-axis IMU has 3 accelerometers, 3 gyroscopes and 3 magnetometer sensors aligned to three orthogonal common axes of the IMU.

5.7.1. Acceleration, Orientation and Heading

When an accelerometer is mounted onto equipment it is used to measure the acceleration along the axis which it was aligned with. Exact position changes cannot be measured with an accelerometer however, acceleration a $(m\ s^{-2})$ is directly related to position x (m) via the double integral presented Equation 5.1. Theoretically, knowledge of exact accelerations over time has enabled exact positional change to be calculated. In reality, IMU acceleration readings from an accelerometer are observed to exhibit noise and therefore error (err). In Equation 5.1 the error term was also integrated twice to determine position changes which would have led the error term to quickly become unwieldly and produce large drifts in position estimation.

$$x = \iint a\ dt + \iint err$$

Equation 5.1. Relation between linear position and acceleration over time

A gyroscope is used to measure the orientation of an object and they are useful for navigational purposes. An object's rotation around a single axis is measured with a gyroscope. Three gyroscope sensors are commonly used together, on separate orthogonal directions, to provide readings of Pitch, Yaw and Roll Euler angles for an object, see Figure 5.24.

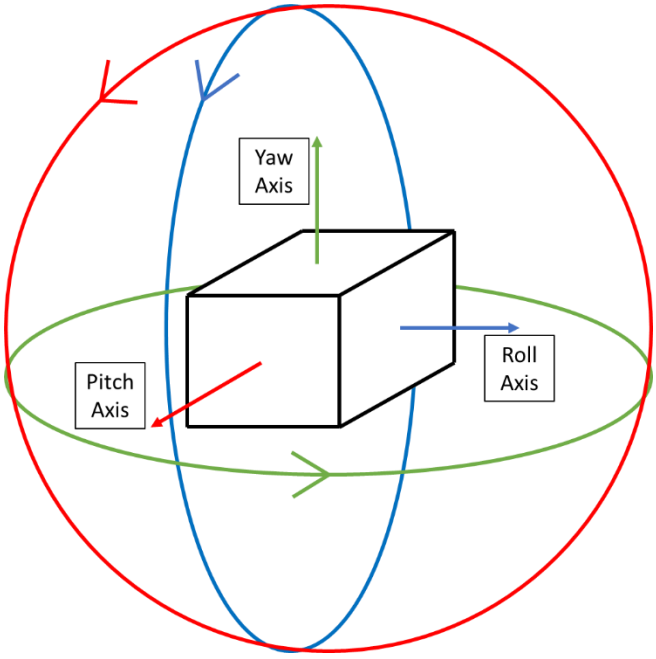


Figure 5.24. Pitch, roll and yaw axes

When used for navigational purposes Euler angles are liable to exhibit gimbal lock. This phenomenon has occurred when two of the three rotational axes have become aligned and a degree of freedom for the gyroscope system is lost. To overcome this phenomenon with digital gyroscopes, it has been common to use quaternions instead to represent the orientation or change in orientation of an object in a 3D coordinate system.

Quaternions rely on using complex numbers, consisting of four elements, to define a rotation in 3D space. The general equation that defined a quaternion $\boldsymbol{q}$ has been provided in Equation 5.2 where $\boldsymbol{q_0}$defined the real part of the quaternion and $\boldsymbol{q_1}$, $\boldsymbol{q_2}$and $\boldsymbol{q_3}$ defined the three imaginary elements of the quaternion where $\boldsymbol{q_0}$, $\boldsymbol{q_1}$, $\boldsymbol{q_2}$ and $\boldsymbol{q_3}$ are all real numbers.

$$\boldsymbol{q = q_0 + q_1 i + q_2 j + q_3 k}$$

Equation 5.2. General form of quaternion

Quaternions have been used to represent the rotation of an object from one orientation to another and have had wide applications across several industries that include navigation and computer graphics. Appendix L has provided further information on the mathematics behind quaternions. Quaternions can readily be converted back into rotation matrices or Euler angles to simplify how orientations are presented. As with accelerometers, sensor readings from gyroscopes have also been observed to exhibit noise and therefore when used in isolation their outputs contain erroneous data.

Three-axis magnetometers are used to measure the magnetic field that a sensor was located in across three orthogonal directions. This has been used to measure the magnetic field of the Earth and determine the heading of the sensor's direction relative to magnetic North. Many magnetometers utilise the Hall Effect defined as where a voltage is produced across a conductor when subjected to a magnetic field that runs perpendicular to current flowing through the conductor. The Hall Voltage across each of the three orthogonal axes was measured to calculate the heading of the sensor. As with the other two sensors presented, in an IMU the magnetometers have also been prone to record signal noise and the readings must be carefully considered.

Unlike the other two IMU sensors magnetometers have been observed to be sensitive to the presence of large electrical equipment located in close proximity to the sensors. The magnetic field produced by most electronic appliances are approximately 10^2 greater than that of the Earth's natural magnetic field. This has caused difficulty when used indoors due to the presence of electrical circuits and transformers. Outside, in a clear and open space, magnetometers have been observed to be less likely to receive interference from these stronger sources.

As discussed, each of the sensor types used in a 9-axis IMU have been subject to noise generation such that when used in isolation they have exhibited drift phenomena that has reduced the reliability of their outputs. Sensor fusion was identified as a technique used to combine data from multiple input streams and sensors by comparing sensor readings against reference signals. When a 9-axis IMU was considered the sensors recorded two reference signals that did not change noticeably as the IMU was moved in a 3D space. These two reference signals included the acceleration due to gravity g and the Earth's magnetic field $\mathcal{B}$. While both terms have been measured to vary across the globe, they have been widely considered constant within a local area such as a survey of a building.

Sensor fusion has enabled gyroscopes to use the acceleration due to gravity, from the accelerometers to provide a reference down direction to minimise errors in orientation readings. The acceleration errors were not dealt with in this fusion combination which has led to the common use of GPS to militate against any acceleration drifting errors. Similarly,

the use of magnetometers has aided reduction in gyroscope drift too by providing a reference heading to compare orientation signals against.

5.7.2. MPU-9250

This stage of prototype development focused on use of the IMU only, without the addition of GPS data. An MPU-9250 was incorporated into the prototype as a 9-axis IMU that contained a 3-axis accelerometer, a 3-axis gyroscope and a 3-axis magnetometer on a single PCB, see Figure 5.25.

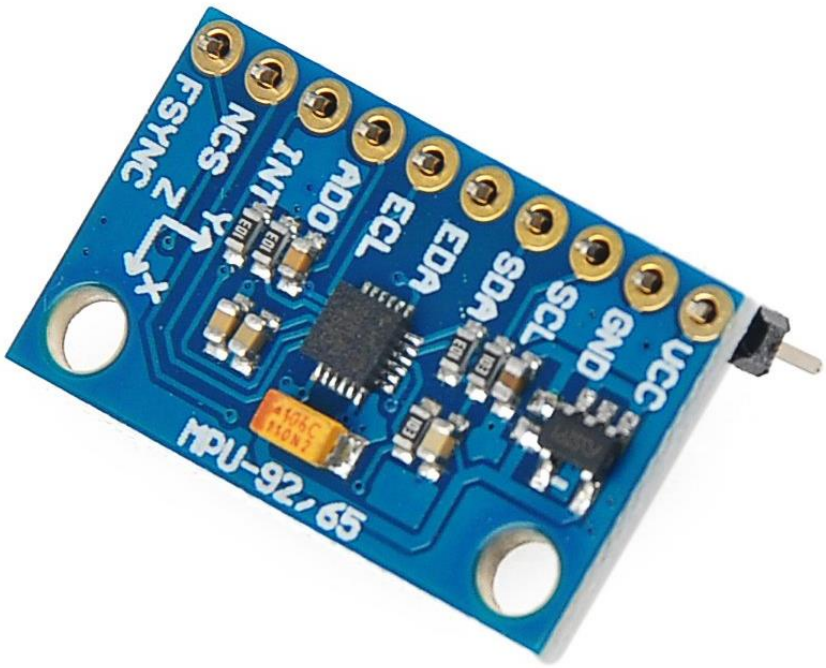


Figure 5.25. MPU-9250

As the prototype was not expected to see large and highly dynamic movements during use the MPU-9250 was configured to its most sensitive detection range settings across all axes in order to detect a smaller range of accelerations and rate of rotations. For the accelerometer this was selected as $\pm 2g$ and the gyroscope was configured to $\pm 250° \, s^{-1}$. The magnetometer only had a single default range detection setting of $\pm 4800 \mu \mathcal{T}$.

Similar to the LL3HP, discussed in subsection 5.4, the MPU-9250 also communicated via the I2C communication protocol enabling a $400 kHz$ data transfer frequency. This allowed it to be used on the prototype in parallel with the LL3HP. The output data rates of the sensors on the MPU-9250 were $4\, kHz$, $1\, kHz$ and $8\, Hz$ for the accelerometer, gyroscope and magnetometer respectively. It was noted that the magnetometer operated at a frequency of

the order 10^3 lower than the other two IMU sensor types. The discrepancy in data output frequencies meant that magnetometer readings did not update as quickly as the other sensors.

As magnetometers are prone to interference the MPU-9250 was mounted onto the prototype such that its distance from any other electrical components were maximised. This was especially crucial regarding the motor which generated a substantial magnetic field during operation compared to the Earth's magnetic field.

The MPU-9250 was wired into the prototype using five signal lines; 1) VCC to receive a $5V$ voltage from the Arduino, 2) GND to ground the MPU-9250, 3) SCL for I2C clock communication with the Arduino, 4) SDA for I2C data communication with the Arduino and 5) FSYNC which was connected to GND as per the manufacturer's instructions, see Figure 5.26.

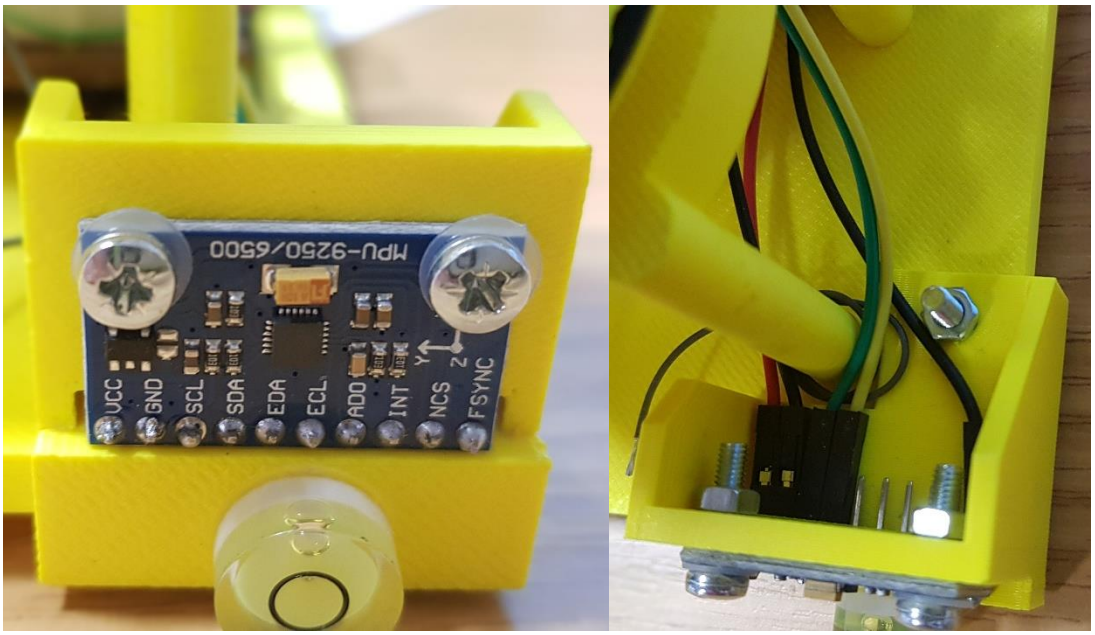


Figure 5.26. (left) MPU-9250 mounted on prototype, (right) MPU-9250 wiring configuration

Figure 5.27 has illustrated the prototyping circuit used to incorporate the MPU-9250 into the prototype development. To utilise the MPU-9250 with the Arduino the firmware was updated to use an open source library specific for the MPU-9250 by Bolder Flight Systems [174]. When the Arduino was connected to its power supply, along with the components previously described in this Chapter, the MPU-9250 was initialised by setting its operational parameters on sensitivity that were previously described. Further

information on calibration and data capture algorithms of the MPU-9250 has been provided in Appendix H.

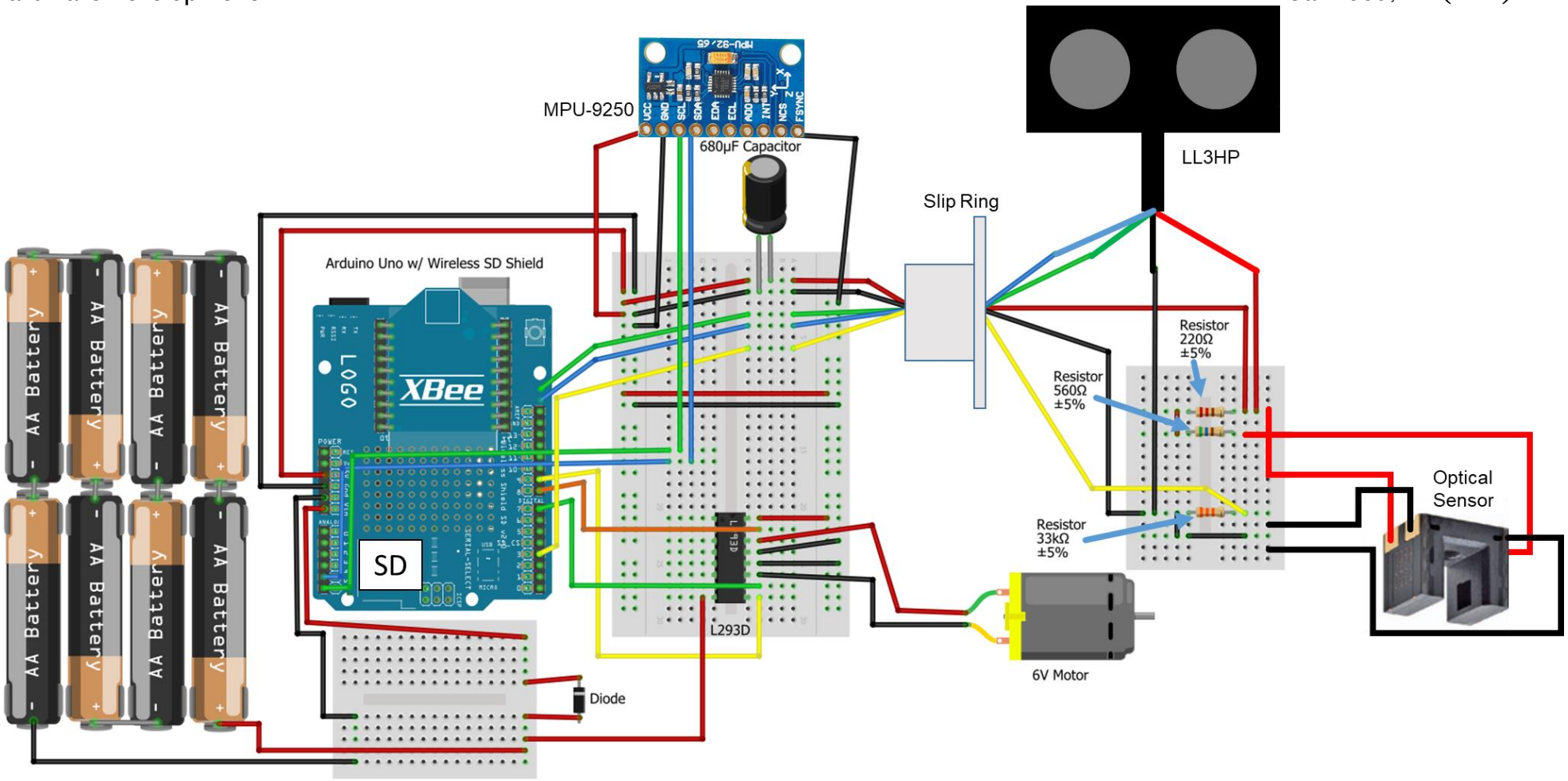


Figure 5.27. Prototype circuit for motion tracking

5.7.3. Sensor Fusion Algorithms

Data fusion techniques were applied to the recorded data set in an effort to account for and mitigate accumulating errors and noise included in recorded data across the accelerometers, gyroscopes and magnetometers. Three data fusion algorithms presented by Kalman [120], Mahony *et al.* [121] and Madgwick [118] have been identified to be used in many applications for fusing IMU data.

The Kalman [120] filter is a noise reduction filter. It has used knowledge of past readings to predict future readings which has cleaned data such that the deviation from trend lines are reduced. This effect has been demonstrated in Figure 5.28 for gyroscope, acceleration and magnetometer readings taken from the developed prototype on a single axis while being carried.

While noise reduction was achieved in the overall signals, the new filtered signals had not taken advantage of reference signals in acceleration or magnetic fields to account for accumulating errors. The data streams remained separate and were stored with respect to the local prototype coordinates.

In comparison, the Mahony *et al.* [121] and Madgwick [118] filters considered data streams across at least two sensor types together. Both filters have been applied to data that either includes accelerometer and gyroscope data or data from all three sensors; accelerometers, gyroscopes and magnetometers. Data stream errors were iteratively calculated at each time point by comparing the linear acceleration and magnetic field in each orthogonal direction to provide an estimate for the direction of gravity and the Earth's magnetic field relative to the sensor axes.

While the two techniques are similar Mahony *et al.* [121] defined proportional and integral gain constants which were multiplied by calculated errors to adjust the gyroscope data at a specific time point. The revised gyroscope data was then used to calculate the rate of change of rotation in quaternion form. In comparison, the technique presented by Madgwick [118] defined a single constant β which was multiplied against a calculated error over each time step with the result then removed from the calculation of the rate of change of rotation in quaternion form. At this point the two methods were found to operate in a similar manner again where the rate of change in orientation was integrated which yielded a new quaternion

for the orientation of the prototype that had been corrected against gravity and/or magnetic reference fields. The methods were used for correcting noise in gyroscope data of the prototype and are described in Appendix I.

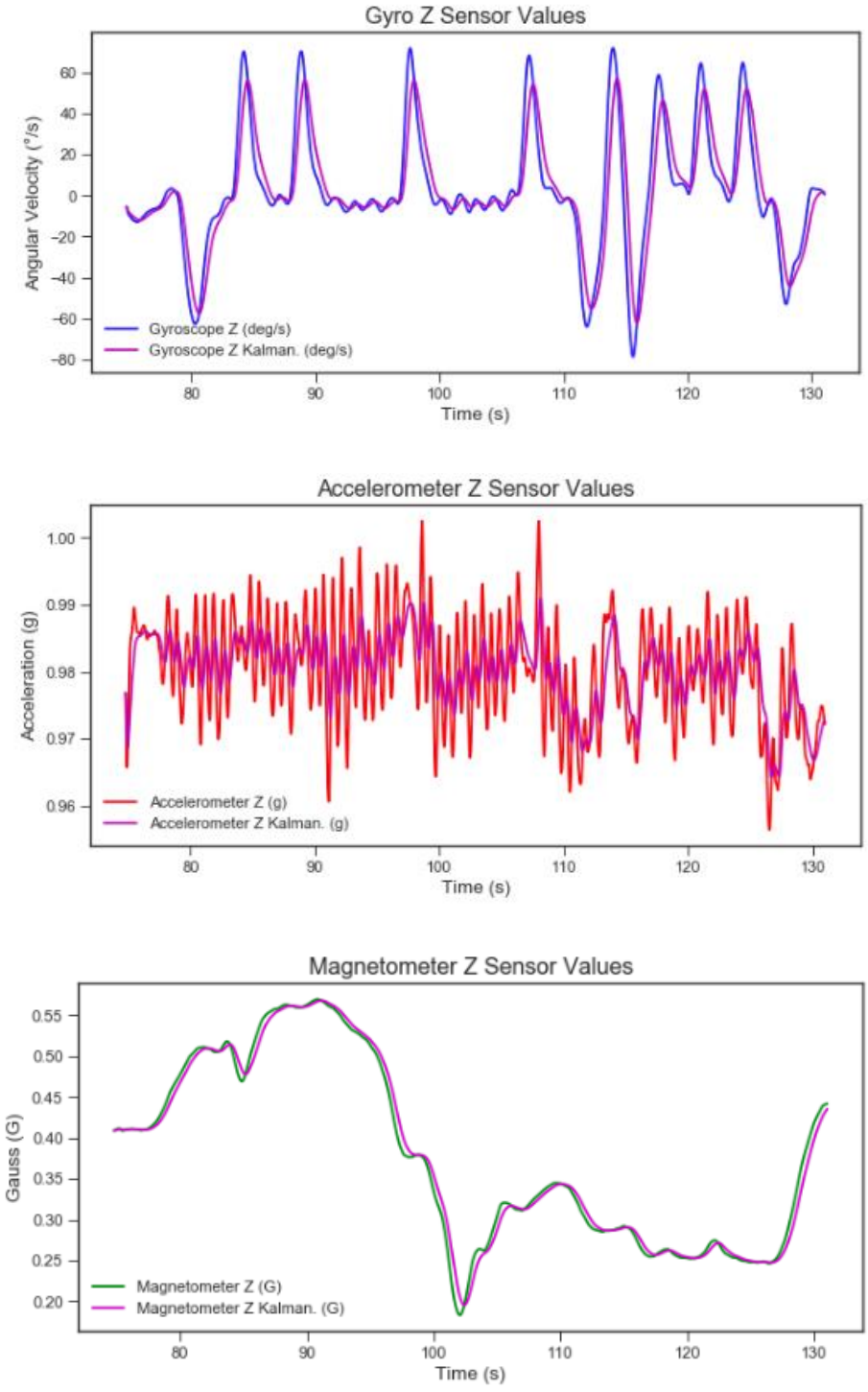


Figure 5.28. Effect of applying Kalman filter to data that exhibits noise

While the Mahony and Madgwick techniques provided a useful method of data fusion they only enabled correction of the gyroscope orientation readings. It is common to find accelerometer readings corrected with an additional data source such as GPS. As the prototype was being developed and tested indoors the addition of GPS capabilities was not deemed practical. For this reason, an alternative method of drift correction in accelerometer data was required to accurately track the position of the prototype in 3D space. One potential method to achieve this was identified as Dead Reckoning.

5.7.4. Dead Reckoning

Dead Reckoning was defined as a method to compensate for any errors and subsequent drift included in positional calculations that have been derived from IMU data. Dead Reckoning was a useful technique in lieu of an additional sensor input, such as GPS, but it required the use of assumptions to determine position changes over time. Madgwick *et al.* [119] presented preliminary results using dead reckoning to accurately track the 3D position of a foot over a $3m$ walking distance. Although not published in academic literature the source code was made available in MATLAB format [175] with additional datasets for testing which was reproduced in the Python programming language for this research as described in Appendix J.

Having identified a suitable post-processing algorithm for survey data and successfully recreated its functions on test data the dead reckoning technique was applied to data captured using the prototype in a lab space. The lab space used for the test has been illustrated in Figure 5.29 as well as the direction of travel during data capture with the prototype. The results from testing are presented in Appendix K.

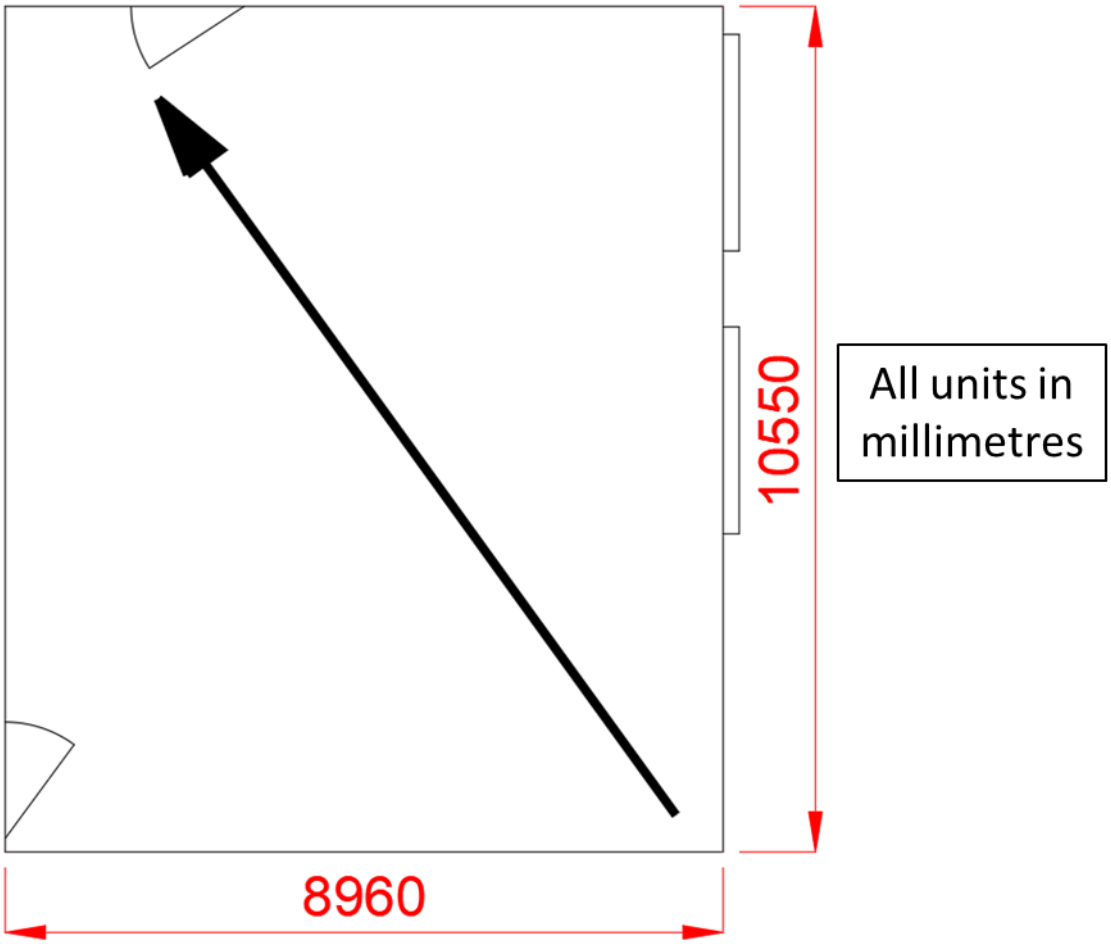


Figure 5.29. Lab space used for motion tracking test on prototype

In a similar manner to the sample data, the data captured by the prototype was processed by the Mahony and Madgwick filters, as described in subsection 5.7.3. This tested whether the prototype motion could be accurately tracked during data capture. Following correction of the orientation via one of these filters the dead reckoning method, as described in Appendix J, was applied to account for drift in stationary and non-stationary periods to calculate the velocity and position of the prototype over time. The plan view of the calculated 3D position from data recorded on the prototype are presented for both variations of the Mahony and Madgwick filters in Figure 5.30. The results from data recorded by the prototype were found to be unsuccessful in comparison with the sample data that was processed in an identical manner.

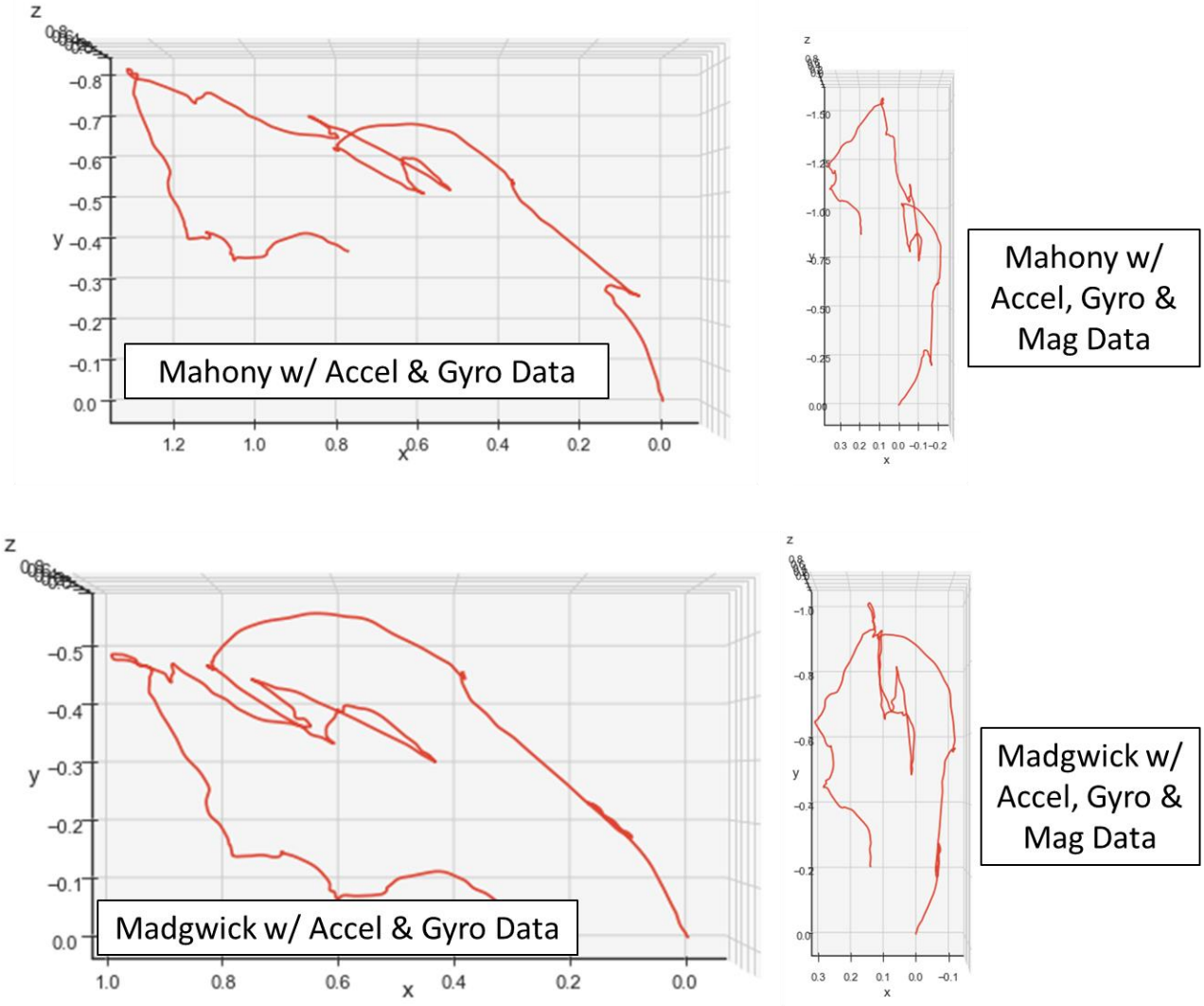


Figure 5.30. Plan view of calculated prototype position during lab test

To understand why the prototype data was unable to replicate similar results to the sample data the raw data was analysed. Upon investigation it was found that the sample test data was recorded at a frequency of $256Hz$. The data recorded by the prototype was only achieved at a maximum frequency of $76Hz$. This meant that the sample data was of a much higher resolution that captured approximately 3.4 times more data over an equivalent time period. It was possible that the higher recording frequency in the sample data was also responsible for enabling the extreme data ranges to be captured which may have been missed by the prototype. While data rows were recorded at different frequencies between the sample and prototype data the frequencies of data on the individual sensors also differed.

In the sample data set the accelerometer and gyroscope readings appeared to match the $256Hz$ of the overall data set with a new data value on at least one of the three orthogonal directions in each row. The magnetometer in the sample data set was found to have repeat readings on all three axes over approximately 3 data rows resulting in a magnetometer refresh rate of approximately 65-85Hz.

When the prototype data was considered the gyroscope and accelerometer readings were found to match the mean $76Hz$ of the prototype update rate. As with the sample data the magnetometer used on the prototype also had a lower frequency than the other two sensors which was estimated between 7.6-8.4Hz. As previously described the MPU-9250 has sensor reading frequency specifications of 4kHz, 1kHz and 8Hz for the accelerometer, gyroscope and magnetometer respectively. As such the recorded result frequencies were expected for the prototype.

Having achieved an improved understanding of the data capture rates of the prototype it was concluded that the much slower rate, in comparison with the sample data, effectively created large gaps in the prototype data collection. This meant that any drift correction from raw sensor data was conducted using lower resolution data which increased the risk of errors occurring when interpolating between data points and reducing the effect of corrections. As such the prototype objective of being able to track its position in 3D space was not achieved during this research. During development of the prototype, that has been described in this thesis, a number of error sources were identified to aid further development in future work.

5.8. Sources of Error

As described in subsection 5.5 the 2D rotation of the prototype was capable of detecting room boundary walls but also generated substantial levels of noise. It was recognised that any future developments should try to mitigate this noise.

The dead reckoning and sensor fusion techniques were outlined as viable solutions however when dead reckoning is used, with the prototype held at chest height, the wider applications of the prototype was limited. It involved a non-intuitive data capture method, via intermittent walking, that was liable to introducing error in the survey and would not have been directly applicable to use on a UAV platform to aid the external survey of a building. The data fusion

techniques that were assessed only provided a reference signal for orientation and not acceleration. The addition of another data signal such as position using GPS may have aided in providing a reference acceleration signal to compare with the accelerometer, but the limited available dynamic memory prevented this during this research.

The low data sampling rate that was identified on the prototype meant that substantial data loss occurred, and drift corrections were not as effective on reduced data. Increasing the sampling rate would require an overhaul of prototype hardware. With the primary error sources of the current prototype development identified an assessment was made on the cost of development and the expected costs of reducing error sources.

5.9. Cost

The final requirement of prototype development was to keep the LiDAR unit cost as low as possible within a target budget of £500. While the motion tracking element of the prototype was not successfully incorporated into the prototype the cost of existing components in the first development iteration of the prototype described in this research were measured. In addition, by identifying areas for development on future prototype iterations the cost of developing those iterations were also estimated relative to current costs. These costs for the developed prototype have been tabulated in Table 5.1.

Table 5.1. Prototype bill of materials and cost

Part	Quantity	Unit Cost	Total Cost
LL3HP	1	£106.01	£106.01
XBee Series 1 Module	2	£28.96	£57.92
Arduino Uno	2	£26.28	£52.56
Arduino Wireless SD Shield	2	£18.67	£37.34
AA Battery	8	£2.20	£17.60
SD Card	1	£12.49	£12.49
Adafruit Slip Ring with Flange (Part No:736)	1	£11.03	£11.03
Prototype Breadboard	1	£7.32	£7.32
MPU-9250	1	£6.99	£6.99
L293D IC	1	£6.42	£6.42
Spool of ABS 3D Printing Wire	~1/3	£12.50	£5.00
Terminal Strip	7	£0.40	£2.80
Serial Cable	1	£2.24	£2.24
6807 ZZ Metal Shielded Thin Wall Deep Groove Ball Bearing 35x47x7mm (61807)	1	£2.20	£2.20
Jumper Wires	25	£0.08	£2.00
Battery Holder	1	£1.76	£1.76
6$\mathcal{V}$ DC Motor	1	£1.49	£1.49
9v Battery Connector	1	£1.39	£1.39
Omron Transmissive Photo Micro-Sensor EE-SX1320	1	£0.92	£0.92
70mm x 2mm O-rings	1	£0.53	£0.53
Capacitor Radial 10v 680uf	1	£0.20	£0.20
Diode 1N4007	1	£0.15	£0.15
220Ω Resistor	1	£0.06	£0.06
560Ω Resistor	1	£0.06	£0.06
33kΩ Resistor	1	£0.02	£0.02
	Total Prototype Cost		**£336.50**

The version of the prototype developed as part of this research was produced at a raw material cost of £336.50 which was 67.3% of the proposed prototype budget of £500. It should be noted that the materials were acquired based on a small volume orders and that cost savings were expected with increased order sizes if mass production was ever achieved. A close inspection of the component's costs revealed that the largest costs in the prototype were associated with the LL3HP, XBee Modules, Arduino Unos and the Arduino Wireless SD Shields which accounted for £253.83 (75.4%) of the total prototype cost. These components were commercially available products that were suited for a wide range of applications, with the exception of the LL3HP, and an opportunity for further cost reduction

was identified as replacing them with bespoke purpose-built components. The first prototype development enabled identification of a functional specification to apply to future design iterations.

As highlighted the key priority areas to reduce error sources for future development included data processing speeds, sensor fusion and reduction of scan noise generation. The main potential components that were identified to aid this additional development would have required GPS functionality, additional data processing capability and another optical sensor as a minimum. In addition, an assortment of miscellaneous components would have been required to implement the additional key functionalities (e.g. jumper wires and resistors). The estimated costs of possible components that could have been used to aid these developments have been tabulated in Table 5.2 as well as an additional 5% to cover the miscellaneous parts.

Table 5.2. Expected costs of prototype future development

Part	Quantity	Approximate Unit Cost	Approximate Total Cost
GPS Arduino Shield	1	£30.00	£30.00
Arduino Uno (or other Arduino board)	1	£30.00	£30.00
Omron Transmissive Photo Micro-Sensor EE-SX1320	2	£0.92	£1.84
Miscellaneous	-	-	+5%
Total Prototype Cost			**£64.93**

If the future development of the prototype incorporated these three key functionalities the total cost of materials has been estimated as £401.43 which remained below the budget threshold of £500 which met the final requirement of the prototype development in this research. Compared with Table 1.1, in which the lowest cost of a LiDAR unit is listed as US$4,500, the developed prototype represented a reduction in unit cost of approximately 90% (based on an exchange rate of £1 = US$1.31). Although the list price of the existing LiDAR units did not represent the total material cost of the unit it was assumed very unlikely that they would be sold at such a large profit margin as to be of a comparable cost to the developed prototype.

5.10. Summary

This Chapter has documented the first development iteration of a low-cost LiDAR unit to aid in the external laser scan of a building for BEMS applications. Geometry reconstruction from internal building scans identified occlusions and omitted areas in the final data sets with external scanning identified as offering a potential solution to these shortfalls. Four design requirements were identified for the prototype towards achieving the overarching aim and objectives of this thesis set out in Chapter 1.

The prototype was developed based on the use of an Arduino Uno to handle the processing of data capture and was twinned with a LL3HP sensor to provide distance measurements which took measurements at a centimetre resolution. Rotational functionality was also developed and calibrated. Mobility of the prototype was obtained through the use of a battery as part of the prototype and two-way wireless communication modules between a transmitter, tethered to a computer, and receiver on the prototype.

When applied to IMU data captured by the prototype the sensor fusion algorithms were not found to be as successful. Sources of error in prototype data were identified as rotational accuracy, low data sampling rates and low available dynamic memory for further developments. The cost of components used in the construction of the prototype were tabulated with the total cost reaching £336.50. This was below the target budget of £500 thereby meeting the final prototype development requirement.

This Chapter has partially demonstrated a proof of concept low-cost LiDAR scanner to be further developed to aid the reduction or removal of barriers to external scanning of buildings for BEMS applications. In lieu of capturing external scan data of F2050 the existing geometry reconstructions, see Chapter 4, were assessed using BEMS and industry accepted validation techniques.

Chapter 6.

Results and Discussion

6.1. Introduction

This Chapter details the results of conducting BEMS using the automated reconstruction of F2050 as described in Chapter 4. The reconstructed geometry output from Chapter 4 is reintroduced in subsection 6.2 followed by model definition methods used to enable the application of BEMS to the reconstructed geometry as described in subsection 6.3.

Subsection 6.4 compares the reconstruction workflows of the proposed automated geometry reconstruction method with the traditional manual workflow applied to the benchmark model. The results of BEMS on the two full reconstructions of F2050 are presented in subsection 6.5. Based on the BEMS results of the full F2050 reconstructions a calibration was conducted on the maximum air change rate, see subsection 6.6

Subsection 6.7 describes the replacement of the more complex area of the F2050 reconstructions with geometry from the benchmark model to create a hybrid reconstruction BEMS. Following the presentation of all simulation results, subsection 6.8 provides a discussion of the research findings, the challenges faced by the proposed method and identifies areas for further development in future research. Subsection 6.9 summarises the contents of this Chapter and highlighted the key findings across the simulation results and the broader research conducted as documented in this thesis.

6.2. Reconstructed Geometry

Following on from the geometry reconstructions that were presented in Chapter 4, the results of applying BEMS to the reconstructions has been documented in this subsection. Two reconstructions of F2050 were selected for further BEMS research which were both obtained by using 26 scans from the ground floor of F2050 while window blinds had been used to mask large glazed areas. The two reconstructions differed in that one used the unfiltered input scans and the second had a $10mm$ voxel filter applied to down sample the number of data points. The resulting reconstructed geometries have been presented again in Figure 6.1 and Figure 6.2 for clarity; Figure 6.2 had the voxel filter applied.

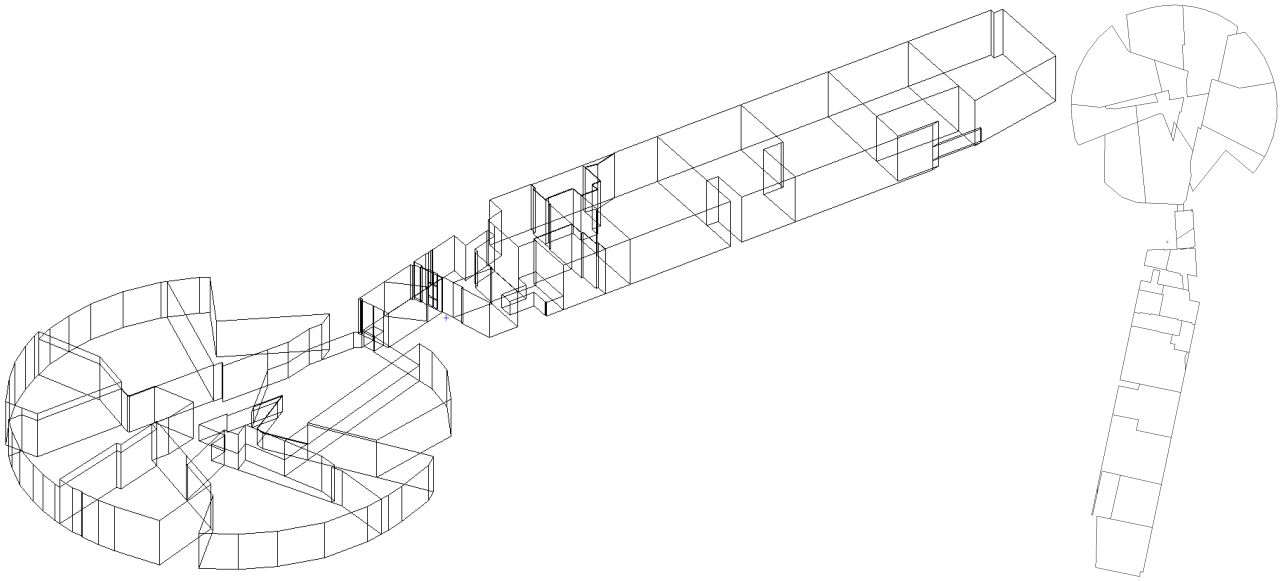


Figure 6.1. F2050 reconstruction of 26 scans w/ blinds down (left) isometric view, (right) plan view

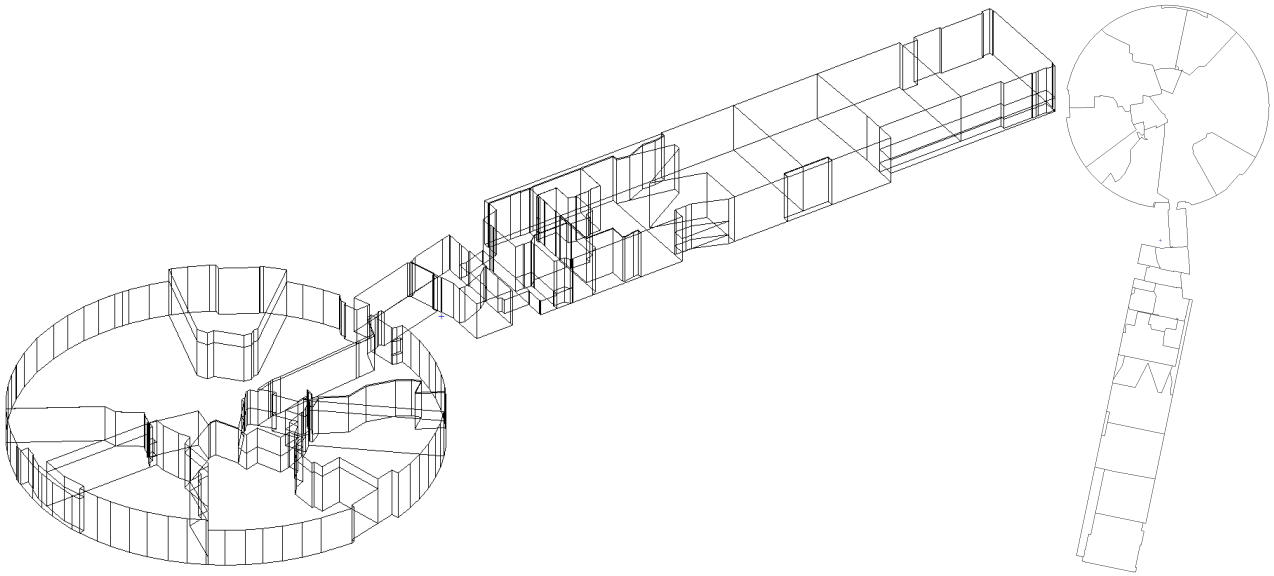


Figure 6.2. F2050 10mm voxel filter reconstruction using 26 scans w/ blinds down (left) isometric view, (right) plan view

It was evident from the reconstructions presented in Figure 6.1 and Figure 6.2 that the use of a voxel filter resulted in the more visually recognisable build form of F2050. This was especially evident for the F2050 circular workshop in which less gaps were observed following application of the voxel filter.

To assess the results of these reconstructions for BEMS applications the energy models needed to have their boundary conditions defined prior to any simulation. It was also noted that the effect of using window blinds meant that windows needed adding into the model manually. In addition, it was identified that the reconstructions associated with individual

scans did not always automatically merge adjacent thermal volumes when a virtual partition was created.

6.3. Model Definition

Both reconstructions were stored in a gbXML format which were imported into IES VE [16]. The model geometries consisted of several thermal zones separated by virtual partitions. These were combined using the "*Connect/Merge space*" tool in IES VE to reduce the number of thermal volumes to four representing the main thermal volumes of F2050.

In addition, any glazing or doors that had not been detected during the reconstruction were manually added back on each of the reconstructions. The large levels of glazing present in the external walls of F2050 meant that this process was likely to take considerable time if the exact glazing pattern of the benchmark model was used. To overcome this time constraint, the external wall surfaces of the circular workshop had glazing applied as a percentage that matched that of the benchmark model. The geometry of F2050 meant that the circular workshop was neatly divided into 24 sectors. The reconstruction results, see Figure 6.1 and Figure 6.2, generated many more external wall panels that made applying the same glazing pattern impractical but the benchmark glazing was calculated to cover 82.5% of each wall and this glazing ratio was also applied to the reconstruction walls.

Doors also needed adding in manually to the BEMS geometry that had not been identified in the reconstruction. When the rear rectangular portion of F2050 was considered the southern façade contained two double doorways. Small discrepancies in the performance of the reconstruction process meant that one of these doorways could not be added in manually in the correct place. The doorway was shifted along the wall slightly in the reconstructed model to a suitable area where it could be modelled as its presence was deemed more important than its specific location.

Following the merging of thermal volumes and definition of F2050 glazing and doors on the reconstructed geometry, the boundary conditions that were applied to the benchmark model were also applied to the reconstructed geometry. This resulted in two reconstructed F2050 models that had fully defined geometry, construction materials and boundary conditions, as described in Chapter 3. These two IES VE [16] models have been depicted in Figure 6.3 and Figure 6.4.

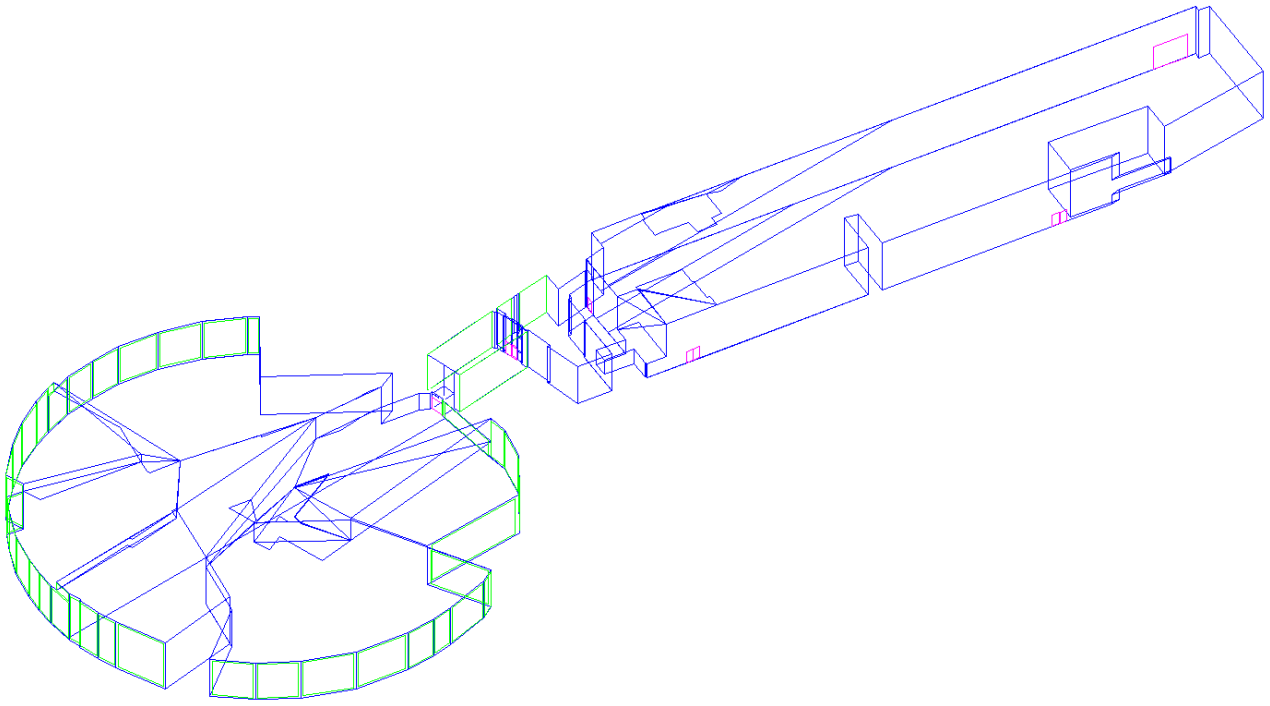


Figure 6.3. IES VE model of F2050 reconstruction without voxel filter

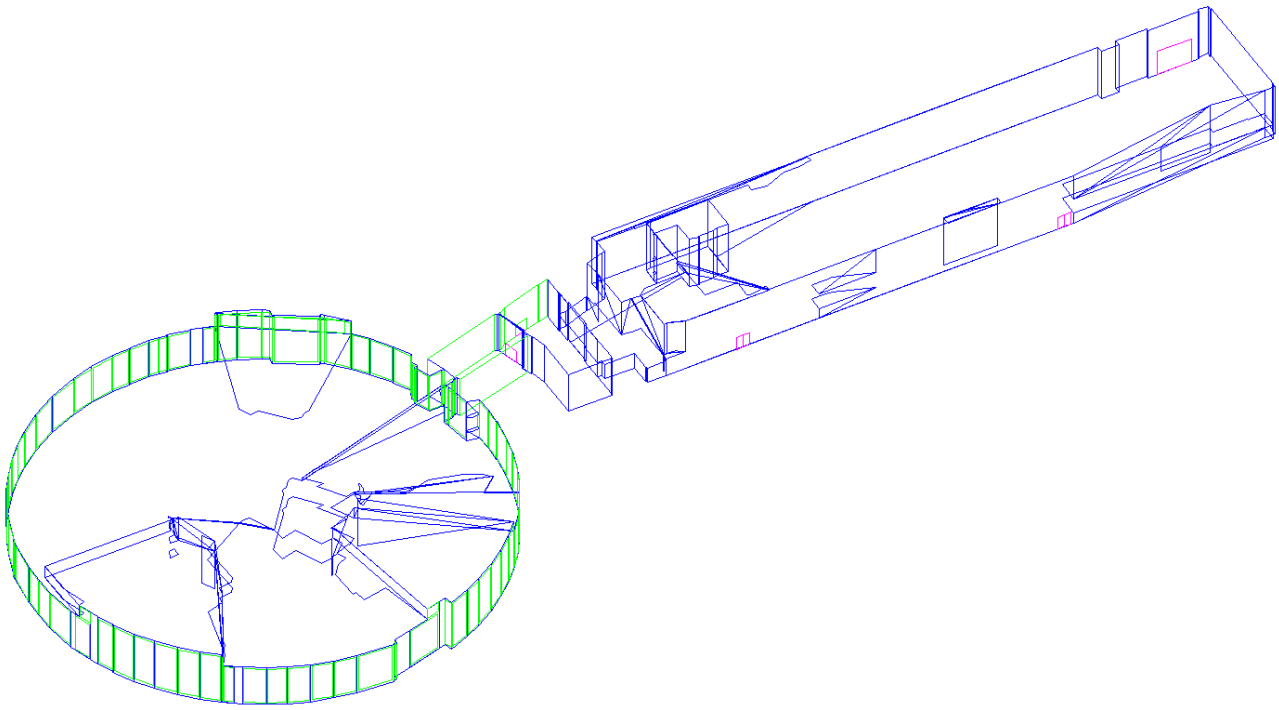


Figure 6.4. IES VE of F2050 reconstruction with 10mm voxel filter

Observations on the reconstructed BEMS models, shown in Figure 6.3 and Figure 6.4, included that where thermal spaces have been merged the geometry of F2050 became slightly corrupted in appearance. This phenomenon was evident on the ceiling surface of

F2050 and was likely caused by the combined effect of complex building geometry (i.e. the circular workshop) and the differing heights of adjacent wall surfaces such that when automatically merged the software attempted to optimise the connection between thermal volumes which may have resulted in non-uniform geometry transitions.

When the reconstructed models were compared with the benchmark model differences were observed in the high-level parameters used to describe each model. These differences are tabulated in Table 6.1 which shows considerable differences between the benchmark and reconstructed geometry.

Table 6.1. Differences between benchmark and reconstructed geometry

Parameter	Benchmark	Reconstruction (No Voxel Filter)	Reconstruction (10mm Voxel Filter)
Volume (m^3)	53,000	36,000	38,000
Floor Area (m^2)	8,700	5,100	5,600
Ext. Wall Area (m^2)	4,000	4,800	4,300
Ext. Opening Area (m^2)	1,700	1,300	1,400

Table 6.1 has demonstrated that the reconstructed F2050 geometry has a reduced thermal volume (-28 to -32%) and floor area (-35 to -41%) compared with the benchmark model. Inspection of both reconstructed geometry cases revealed missing volumes throughout F2050 which, in addition to only using ground floor scans, would have resulted in reduced thermal volumes in the thermal model. The fact that the internal office spaces, within the circular workshop, were not included in either reconstruction was also responsible for a reduced floor area when compared to the benchmark models. The external opening areas was also observed to be reduced in the reconstructions which was expected with the use of only ground floor scans. Interestingly, the external wall area was found to be larger than that of the benchmark model for both reconstructions which was likely caused by the increased perimeter around missing volumes. Across all four parameters the reconstruction that was created following the application of a 10mm voxel grid filter had parameter values closer to the benchmark model than the reconstruction without the voxel filter. The smoothing effect of the voxel filter that was observed in the F2050 reconstructions is likely to have been responsible for this result with much smaller missing volumes and less discontinuities where thermal volumes have been merged in the IES VE [16] model.

6.4. BEMS Workflow Improvement

As discussed throughout this research the traditional workflows of building geometry creation for BEMS application were costly and time consuming. The methods of automated geometry reconstruction that were presented observed a considerable improvement to the time taken to generate building geometry in BEMS software, see Table 6.2 [176]. The traditional workflow was estimated as taking $120h$ to complete whereas the automated method workflow was estimated at $18.3h$ to complete. This established the ability to produce building geometry quickly with an approximate reduction of 85% in workflow timescales.

Table 6.2. Comparison of manual and automated workflows

Method	Activity	Time
Traditional & Manual	Geometry Production	3 weeks ($120h$)
Automated	Site Survey w/ Laser Scanner – 26 scans (see subsection 4.2)	$4.3h$
	Scan Registration (see subsection 4.3)	1 day ($8h$)
	Pre-processing & Geometry Reconstruction (see subsection 4.5)	0.5 days ($4h$)
	Apply Boundary Conditions (see subsection 6.3)	0.25 days ($2h$)
	Total	**$18.3h$**

Having established that the new automated method of building geometry reconstruction provided an improved workflow timescale to complete, the reconstructed F2050 geometries needed to be simulated in BEMS software for comparison with the benchmark BEMS and the measured data.

6.5. Simulation Results

Once the two reconstructed models had been fully defined, a simulation was run on each for comparison against the metered energy data of F2050 over the same 12-month period in 2016. The F2050 reconstruction that did not utilise a voxel filter produced an annual prediction of energy use that was 1.8% lower than that measured via utility bills for F2050. The monthly comparison between the simulated and measured data has been provided in Figure 6.5. One immediate observation of these results was that the simulation predicted much greater energy use in winter months than compared with the measured data and

conversely, much lower predictions of energy use in the summer than in reality. Despite this observation the overall annual energy use in the prediction was closer to measured data than the previously validated benchmark model.

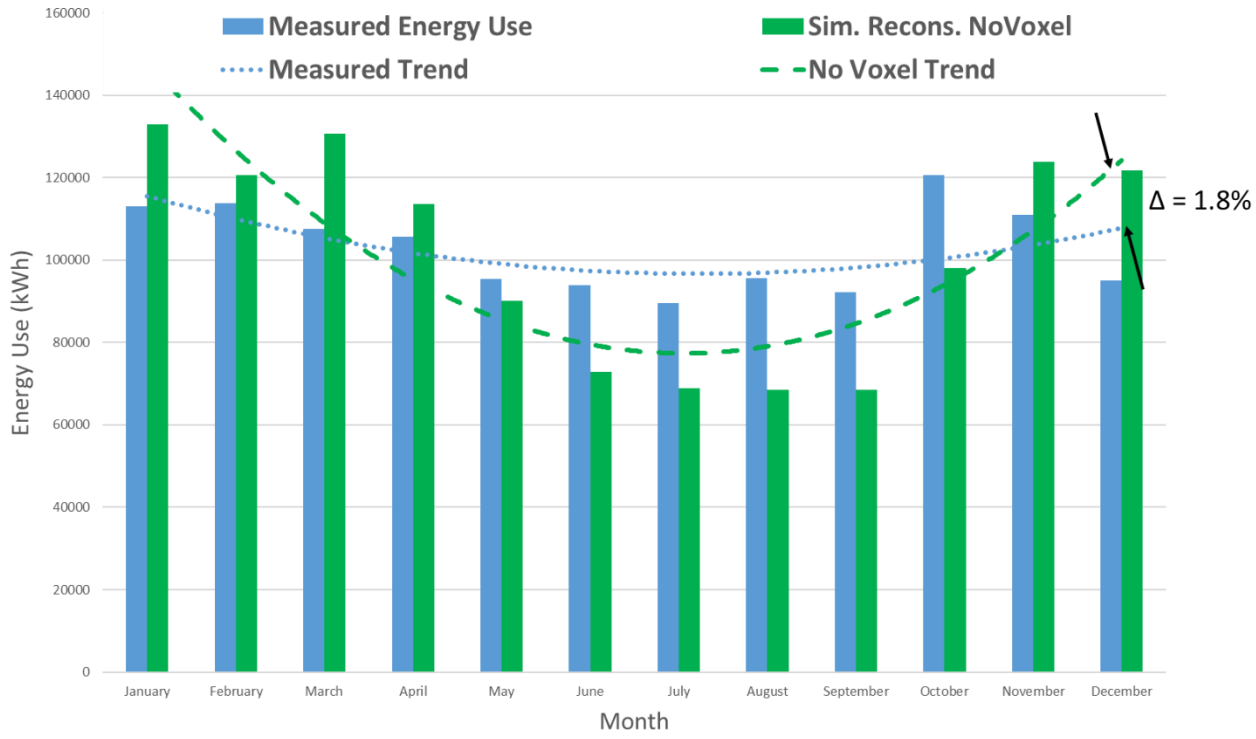


Figure 6.5. Reconstructed BEMS results compared with measured data (no voxel filter)

The validation indices of NMBE and CVRMSE, as specified in ASHRAE Guideline 14 [51], were calculated for the reconstruction results without the use of a voxel filter. These indices were calculated as 1.99% for NMBE and 19.98% for CVRMSE. The NMBE result fell below the maximum threshold of the validation criteria but the CVRMSE did not. As such the reconstructed BEMS, without a voxel filter and using the same boundary conditions as the benchmark model, was not considered to be validated.

As previously discussed ASHRAE Guideline 14 [51] did not provide maximum limits on the simulation discrepancies for individual months and it was observed that six months out of twelve exhibited at least a 20% discrepancy with the measured data. These discrepancies caused the large differences observed in trend lines of Figure 6.5 at the middle and ends of the simulation year even though the overall energy use prediction was found to be within 1.8% of the measured data. A limitation to using NMBE is that it is prone to cancellation

effects, as discussed in Chapter 3, when over- and under-use of energy is predicted across a simulation window which may have caused an acceptable validation index to be achieved.

The same simulation was conducted on the geometry reconstruction that had utilised a $10mm$ voxel filter. Over the 12-month simulation window the energy prediction differed by 11.2% when compared with measured data. Although a similar pattern of over or underestimating energy use on individual months was observed the differences against measured data were greater for the reconstruction that used the voxel filter. The comparison of simulation results and measured data when a voxel filter had been applied to the reconstruction process have been provided in Figure 6.6.

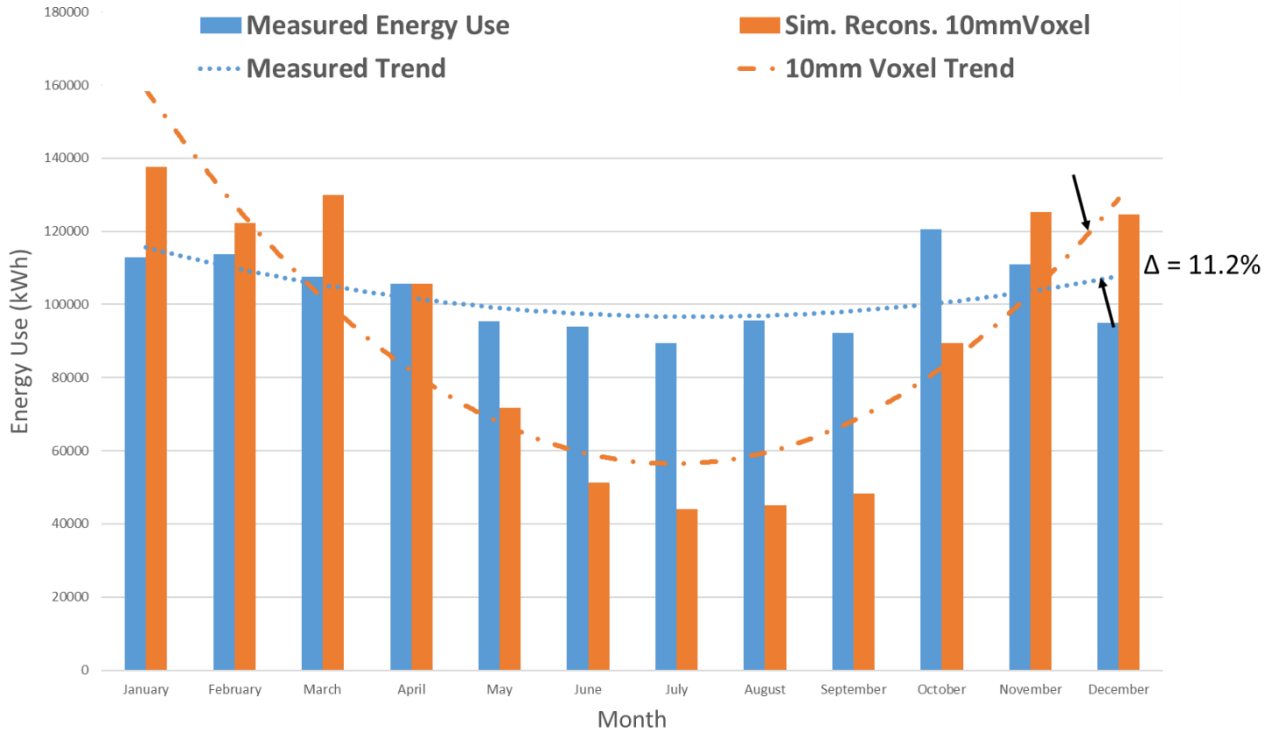


Figure 6.6. Reconstructed BEMS results compared with measured data (10mm voxel filter)

In a similar manner the validation indices NMBE and CVRMSE were calculated as specified in ASHRAE Guideline 14 [51]. For the results of energy simulation of reconstructed F2050 geometry, that used a $10mm$ voxel filter, the NMBE was calculated as 12.19% and the CVRMSE was calculated as 32.34% both of which exceed the index thresholds of 5% and 15% respectively. As a result, the simulation of the voxel filter reconstruction was also not considered to be validated. An interesting observation in these results was that the reconstruction that visually looked more representative of F2050 achieved worse results.

One potential reason for this was that with an improved external geometry model the effects of having no internal thermal mass became more prevalent and increased the rate of heaty loss from F2050. Table 6.3 has tabulated the monthly energy predictions across all simulations and compared these with the corresponding measured monthly data.

Table 6.3. Differences between monthly simulation results and metered data

Month	Metered Data (MWh)	Benchmark (MWh)	Reconstruction w/ no filter (MWh)	Reconstruction w/ 10mm voxel filter (MWh)
January	112.9	113.2 (+0.3%)	132.9 (+18%)	137.6 (+22%)
February	113.7	101.4 (-11%)	120.7 (+6%)	122.3 (+8%)
March	107.5	108.5 (+1%)	130.6 (+22%)	129.8 (+21%)
April	105.6	103.5 (-2%)	113.7 (+8%)	105.7 (+0.1%)
May	95.5	96.9 (+1%)	90.1 (-6%)	71.7 (-25%)
June	93.9	88.0 (-6%)	72.9 (-22%)	51.4 (-45%)
July	89.5	93.6 (+5%)	68.8 (-23%)	44.1 (-49%)
August	95.6	86.7 (-9%)	68.5 (-28%)	45.1 (-53%)
September	92.2	89.1 (-3%)	68.4 (-26%)	48.3 (-48%)
October	120.6	100.0 (-17%)	98.1 (-19%)	89.4 (-26%)
November	111.0	101.8 (-8%)	123.9 (+12%)	125.3 (+13%)
December	95.0	109.0 (+15%)	121.8 (+28%)	124.5 (+31%)
Total	**1,233**	**1,192 (-3.3%)**	**1,210 (-1.8%)**	**1,095 (-11%)**

An observation of the data in Table 6.3 revealed that, when reconstructions were compared, when an overestimate of energy use was predicted against measured data the reconstruction results ranged 7.9 percentage points as a maximum. However, the range increased considerably when an underestimate of energy use was considered which

resulted in reconstruction results ranging by a maximum of 25 percentage points with the voxel filtered reconstruction underestimating substantially more than the reconstruction without voxel filtering when compared against measured data.

The largest discrepancy between the two reconstructions were seen in the simulation results of the months May-September 2016. During these months the benchmark model deviated by a maximum of 9% from the measured data while the reconstructions exhibited maximum deviations of 28% and 53% for the reconstructions without and with a voxel filter applied respectively.

Across both reconstructions it was evident that during the colder winter months the heating demand was considerably greater than reality which suggested a greater loss of heat rate from the thermal volumes of the reconstructions. This was also supported by the considerably smaller cooling demands exhibited in the hotter summer months than seen in the measured data.

The maximum air infiltration rate that was defined for the simulations was set to $0.167ach$ based on the known volume and surface areas of F2050. As the thermal volume and surface area parameters differed in the reconstructions, see Table 6.1, the ach value used will have resulted in a different infiltration rates during the simulations. This is evident when the two reconstructions were compared as the reconstruction that used the $10mm$ voxel filter visually appeared to better resemble the F2050 geometry than the reconstruction without a filter. This resulted in a larger thermal volume so that when $0.167ach$ was applied to both reconstructions under identical environmental conditions the air infiltration rate $(m^3\ s^{-1})$ would have been 5.6% greater on the voxel filtered reconstruction. As such it was deemed worthwhile to calibrate the ach value on both reconstructions to account for any modelling deficiencies.

6.6. Air Infiltration Calibration

Based on the results of the simulation on the F2050 reconstructions it was determined that an increase to the maximum ach value applied to the simulation may increase the total energy use. When F2050 had a heating demand, an increased air change rate meant that more heating of the thermal volume would be required to achieve the HVAC set point temperature of $21°C$ to compensate for heat lost through infiltration losses. Conversely,

when F2050 experienced a cooling demand, an increase to the maximum ach value would have required more cooling to the thermal volume to account for the higher rate of warm air infiltrating the building envelope.

The chosen value of 0.167ach from the benchmark BEMS of F2050 was selected through a simplified air infiltration method. A more detailed method was outlined and found that the simplified calculation results matched closely with the median ach ranges calculated via the detailed method. As part of calibrating the reconstructions the upper quartile value of 0.263ach from the detailed method was selected as this was calculated based on the F2050 infiltration paths and average environmental conditions.

When the 0.263ach calibration was applied to the F2050 reconstruction without a voxel filter the annual energy use was predicted to within 9.4% of measured data which was an increased from the 1.8% observed pre-calibration, see Figure 6.7. The validation indices, specified by ASHRAE Guideline 14 [51], NMBE and CVRMSE were calculated as 10.25% and 26.89% respectively which were increased values that still did not fall below the validation thresholds.

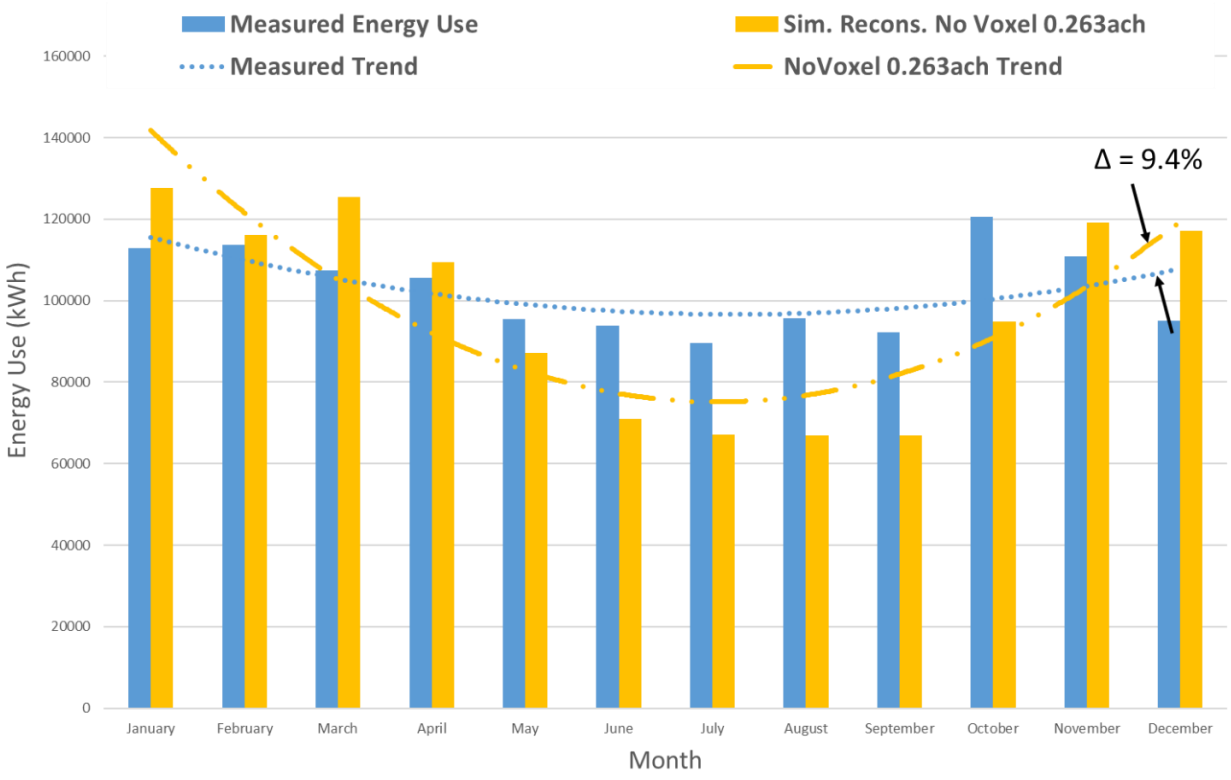


Figure 6.7. Calibrated ach results on no voxel F2050 reconstruction compared with measured data

When this calibration was applied the total annual energy prediction for the voxel filtered reconstruction was increased to within 1.4% of the annual measured energy use of F2050 which improved upon over the 11.2% discrepancy that was observed pre-calibration, see Figure 6.8. The validation indices, specified by ASHRAE Guideline 14 [51], NMBE and CVRMSE were calculated as 1.57% and 35.00% respectively. While an improvement was observed to the NMBE value which then met the required validation criteria the CVRMSE value remained outside the validation criteria.

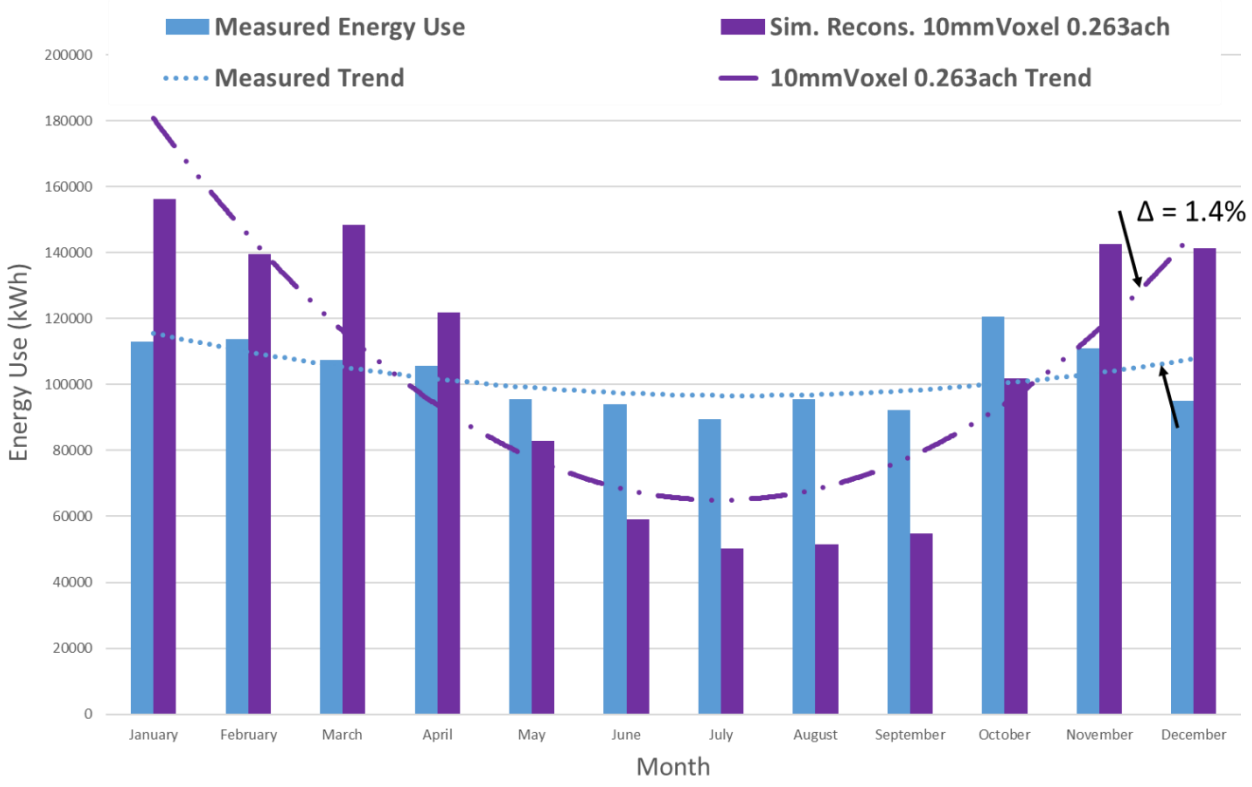


Figure 6.8. Calibrated ach results on 10mm voxel F2050 reconstruction compared with measured data

Both calibrated reconstructions did not achieve validated BEMS against ASHRAE Guideline 14 [51] and the individual monthly discrepancies of both models were observed to differ from measured data by at least 20% regularly. Overestimates for energy use tended to be provided for winter months while in summer months they were underestimated for the reconstructed BEMS. Despite this, over the course of the year the overall energy use predicted in both reconstructions matched closely to measured energy data even if it varied around the measured data points.

Having identified these large monthly discrepancies, the monthly data was plotted as a Kernel Density Estimation (KDE) plot to observe how the simulation results deviated in comparison to the benchmark BEMS and measured data, see Figure 6.9. A KDE has been used to provide a visual representation of the probabilistic density of a continuous variable which, in the case of this research, is the monthly energy use taken from the reconstruction BEMS.

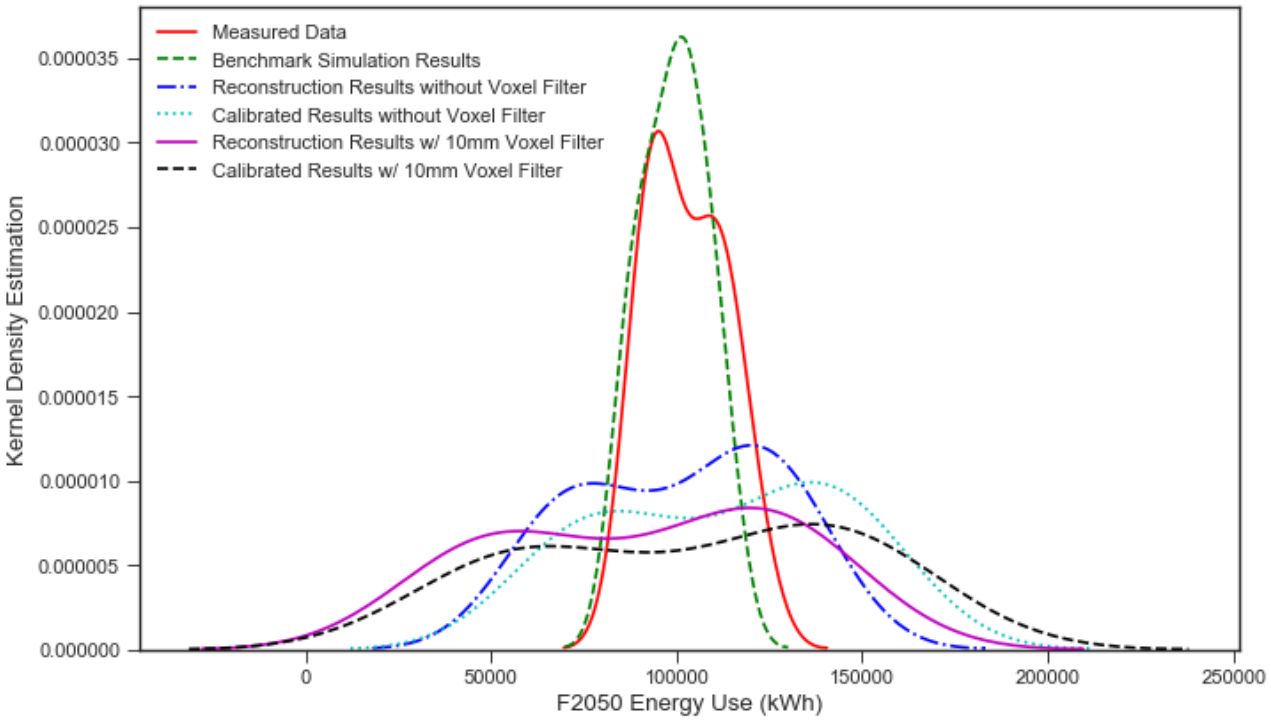


Figure 6.9. KDE plot of simulated results and measured data

Figure 6.9 demonstrated that the benchmark simulation results had a similar standard deviation compared with the measured data. When the distribution of the reconstructed geometry simulations was interrogated several observations were made. The distribution of monthly results for the reconstruction without a voxel filter demonstrated a much greater variance in comparison with both the benchmark and measured data with a reduced maximum KDE value calculated. When the reconstruction that included a voxel filter, pre-calibration, was investigated the distribution of results was observed to be greater still with the curves extending below $0kWh$. Following calibration of both reconstructions, a similar distribution of data was observed however peaks in the data were shifted to higher energy levels as expected.

One of the key challenges associated with using F2050 as a case study was the complex geometry of its circular workshop. A limitation of the presented method was that only ground floor scans could be used under the assumption that the circular workshop geometry accounted for the majority of the thermal volume that included the central offices over multiple floors in the core of the circular workshop. As no validated reconstructed BEMS model of F2050 was achieved a subsequent hybrid approach was considered.

6.7. Hybrid BEMS Sensitivity

The Hybrid sensitivity investigation of F2050 was conducted on both reconstructions by removing the circular workshop from the IES VE [16] models presented in Figure 6.3 and Figure 6.4 and importing the circular workshop from the benchmark IES VE model, see Figure 6.10 and Figure 6.11. This had the effect of including the complexity of the circular workshop accurately in the hybrid BEMS and assessing the reconstruction method on the rear portion of F2050. Both reconstructions were initially simulated using the identical boundary conditions of F2050.

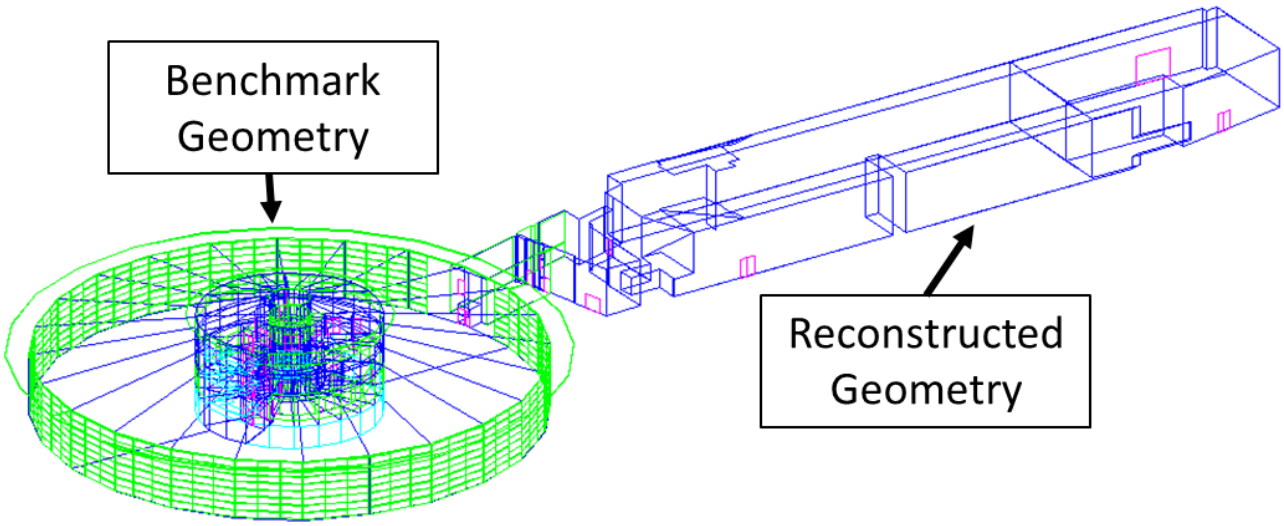


Figure 6.10. Hybrid BEMS of F2050 (no voxel filter)

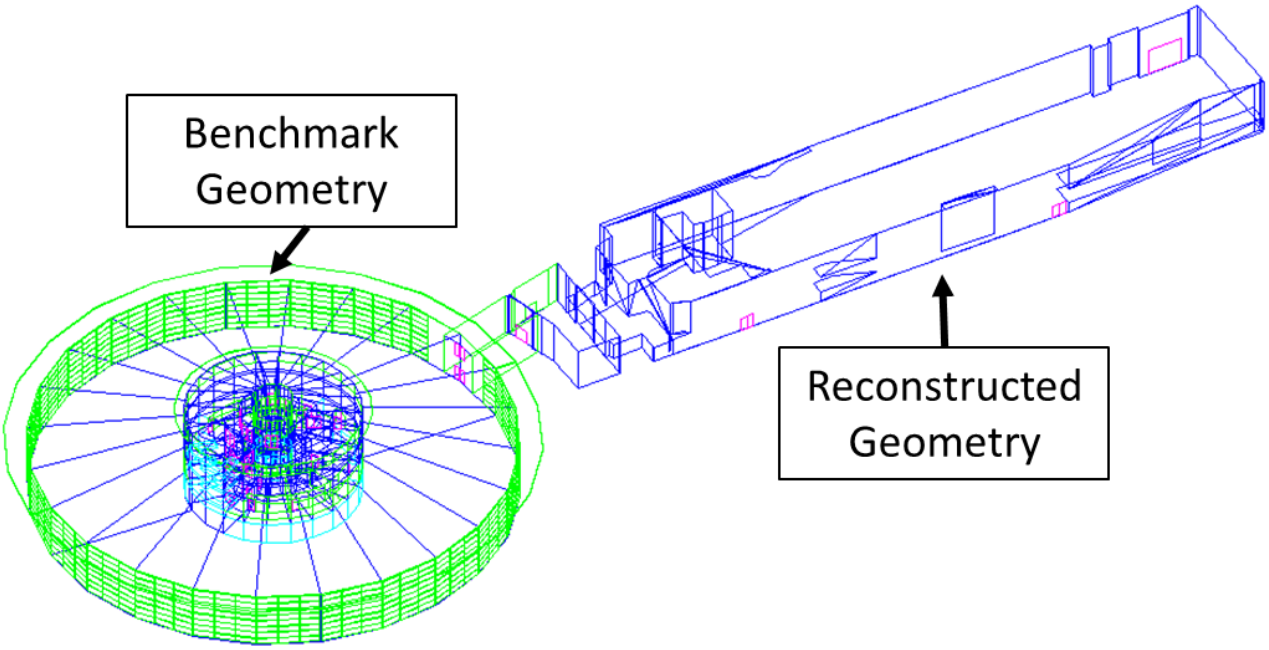


Figure 6.11. Hybrid BEMS of F2050 (10mm voxel filter)

When BEMS was conducted on the hybrid reconstruction without a voxel filter the simulation predicted the annual energy use to within 8.4% of the measured energy data, see Figure 6.12. This simulation was conducted with 0.167ach as per the benchmark model. The ASHARE Guideline 14 [51] validation indices were calculated as 9.13% for NMBE and 12.17% for CVRMSE so while the NMBE fell outside of the validation indices this F2050 reconstruction achieved a valid CVRMSE value below the 15% threshold.

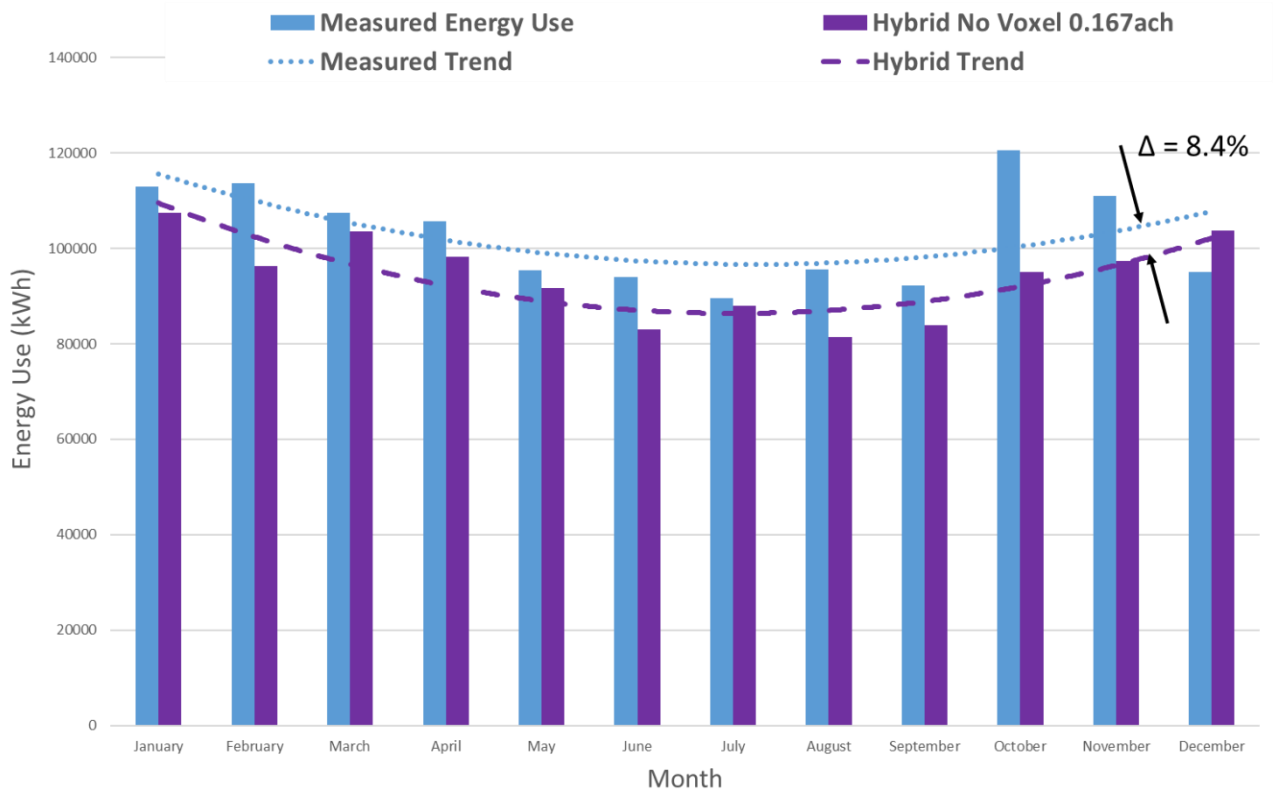


Figure 6.12. Hybrid BEMS results of no voxel F2050 compared with measured data (0.167ach)

The same simulation was conducted on the hybrid reconstruction that used a $10mm$ voxel filter but the results were not favourable. The simulation predicted the annual energy to be 37% different to the measured data, see Figure 6.13. The ASHARE Guideline 14 [51] validation indices were calculated as 40.24% for NMBE and 41.82% for CVRMSE so no validation criteria was met. As previously discussed, even though the reconstruction had improved visually the effects of a lack of internal thermal mass may have become more prevalent leading to the observed results.

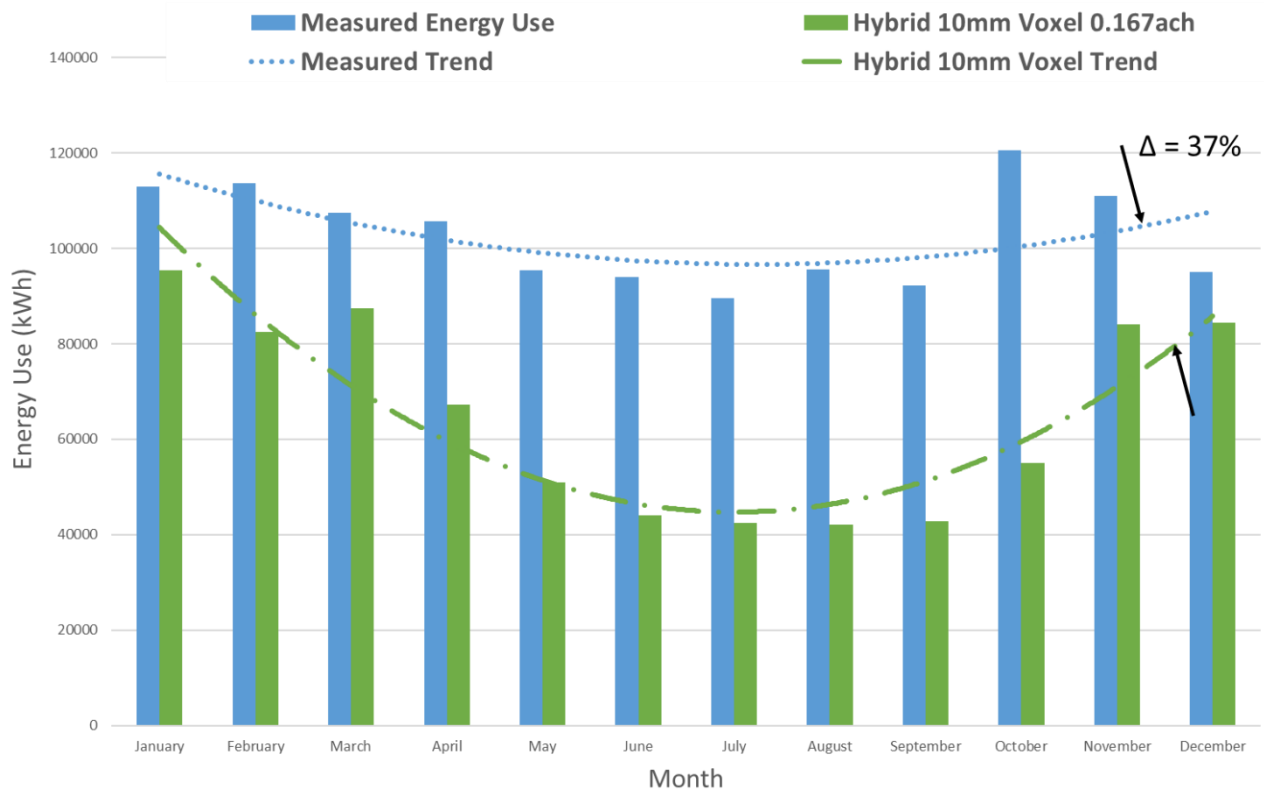


Figure 6.13. Hybrid BEMS results of 10mm voxel F2050 compared with measured data (0.167ach)

As with the non-hybrid reconstructions a calibration was performed on the maximum ach value used during the hybrid reconstruction simulations. The hybrid reconstruction, without a voxel filter, was simulated using a maximum air change rate of $0.263ach$ in an attempt to achieve a validated model. The results of the simulation predicted the annual energy use to within 3.8% of the measured data, see Figure 6.14. The ASHARE Guideline 14 [51] validation indices were calculated as 4.12% for NMBE and 9.98% for CVRMSE. Both of these calculations achieved values below the specified thresholds of 5% and 15% respectively and thus the Hybrid Reconstruction of F2050, without a voxel filter, was deemed valid when the maximum infiltration parameter was set to $0.263ach$.

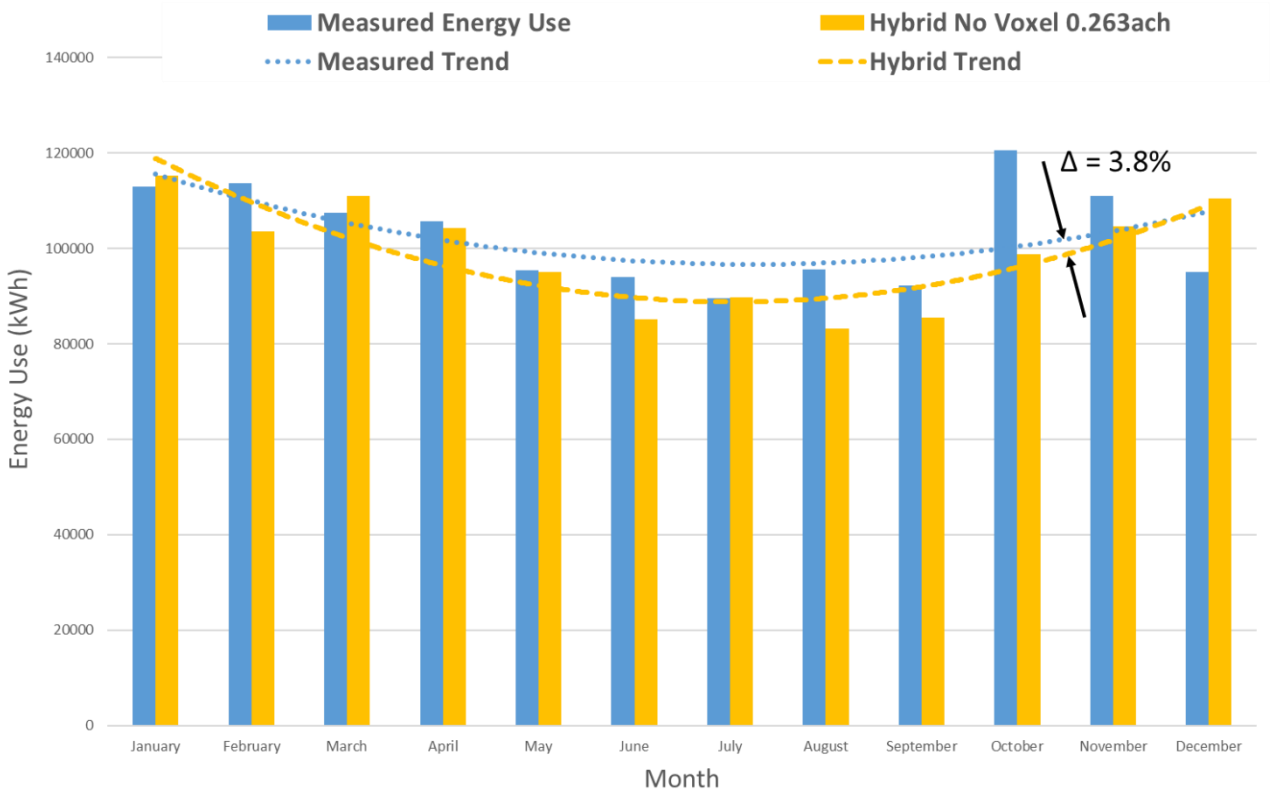


Figure 6.14. Hybrid BEMS results of no voxel F2050 compared with measured data (0.263ach)

This calibration was repeated for the hybrid reconstruction generated via the use of a $10mm$ voxel filter and while improvements were made over the pre-calibration results a validated set of results was not obtained. The results of the simulation predicted annual energy use that differed by 23% when compared with the measured data, see Figure 6.15. The ASHARE Guideline 14 [51] validation indices were calculated as 25.26% for NMBE and 34.18% for CVRMSE which did not meet validation criteria.

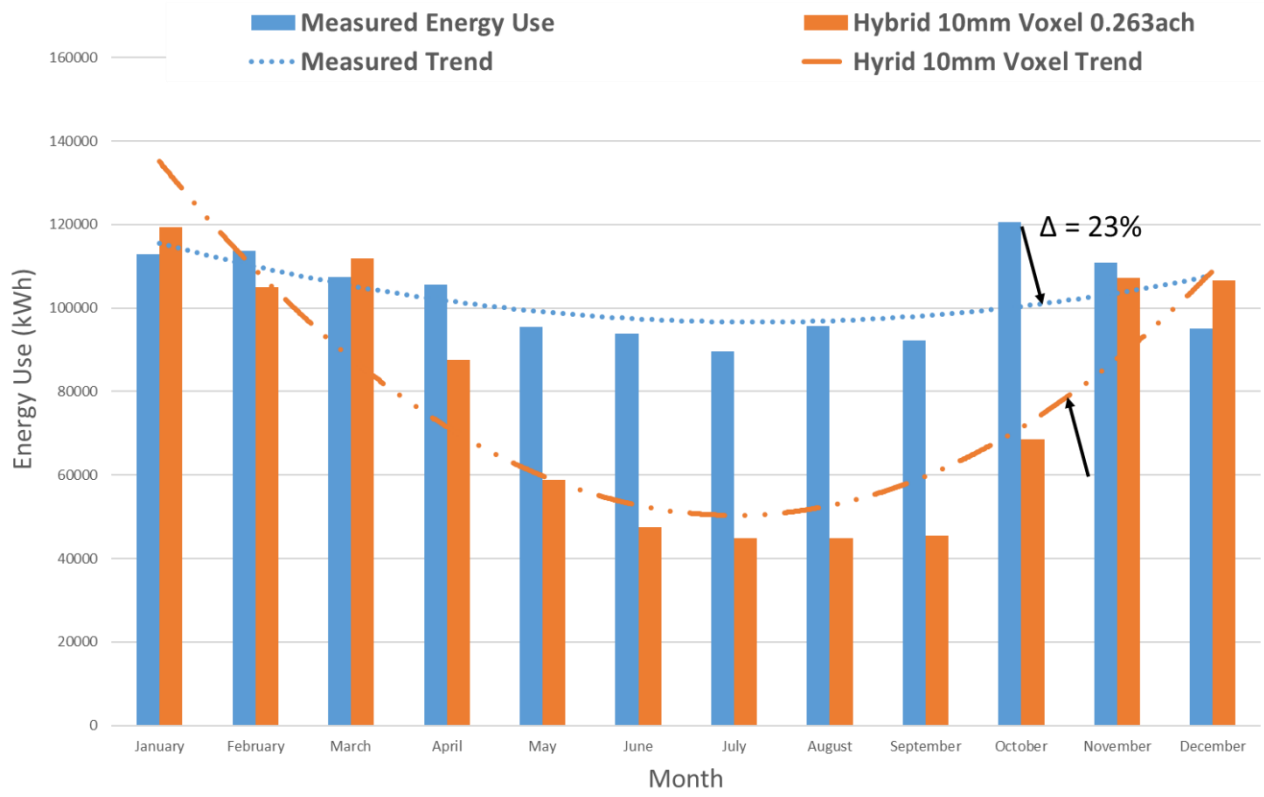


Figure 6.15. Hybrid BEMS results of 10mm voxel F2050 compared with measured data (0.263ach)

Being unable to produce a validated model in the hybrid reconstruction with a $10mm$ voxel filter at $0.263ach$ a sensitivity analysis was carried out to determine whether the results could meet the validation criteria by increasing the maximum air change rate in the simulations. Several more simulations were conducted that defined the maximum infiltration rate between $0.27ach$ and $0.4ach$ at $0.01ach$ increments, see Figure 6.16. This calibration highlighted that while the NMBE could have achieved an acceptable value at $0.385ach$ the lowest CVRMSE was only possible at $0.31ach$ and that increasing the ach parameter further would not have resulted in a validated set of BEMS results.

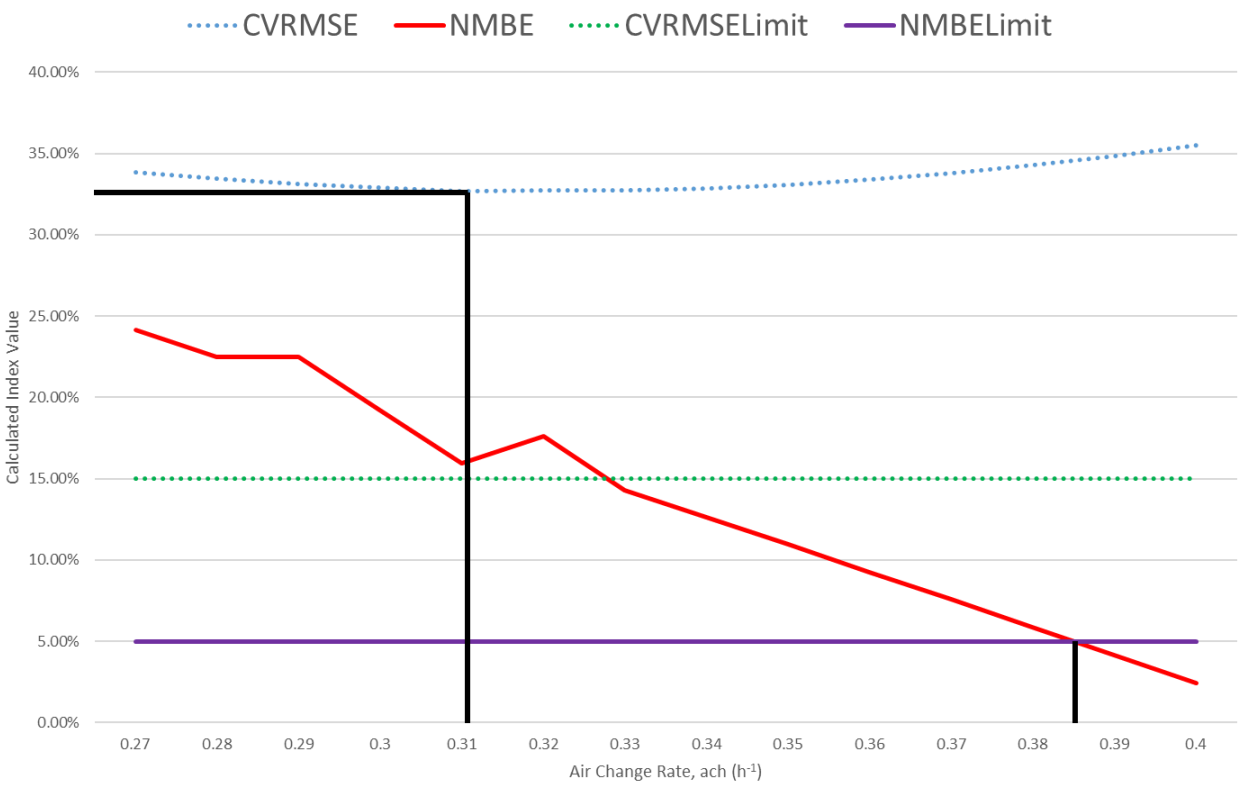


Figure 6.16. CVRMSE and NMBE calibration of hybrid F2050 10mm voxel reconstruction

Having achieved a validated BEMS for the Hybrid Reconstruction without a Voxel filter the distribution of simulation result data was plotted into a KDE, see Figure 6.17.

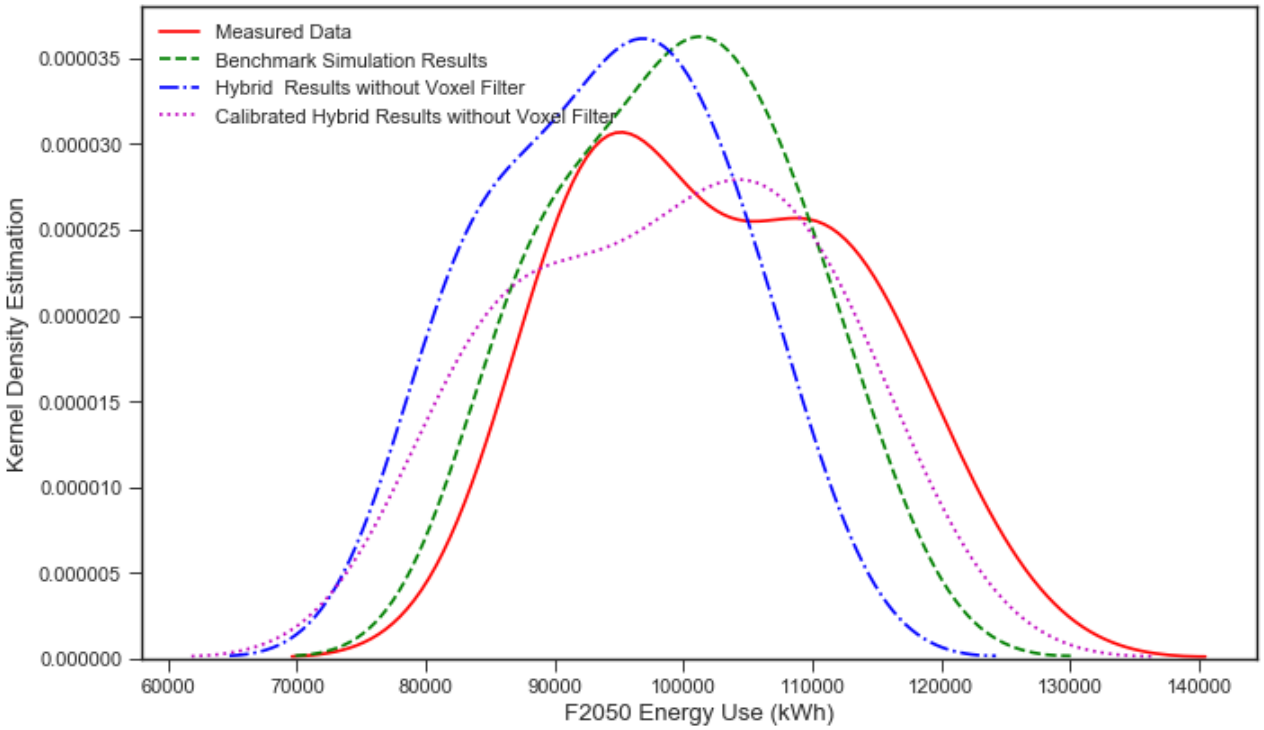


Figure 6.17. KDE plot of hybrid simulated results and measured data

The distribution of data in the validated hybrid model was observed to provide a much better match to both the measured and benchmark models. Prior to calibration the shape of the curve matched the benchmark curve very closely although offset to lower energy levels. The calibration process shifted the distribution to higher energy levels that enabled validated NMBE and CVRMSE indices to be calculated. However, the shape of the calibrated KDE curve has been warped consequently.

6.8. Discussion

The conducted research demonstrated that the BEMS workflow has been improved through the implementation of digital scanning survey and automated geometry reconstruction methods. An 85% improvement was observed in the time taken to produce the building geometry of F2050 for BEMS applications when compared with traditional workflows. The results from BEMS conducted on the reconstructed geometry underwent validation against industrial guidance using the NMBE and CVRMSE indices. The simulation results could not be validated on full reconstructions of F2050 however the hybrid reconstruction without a voxel filter achieved validation indices within acceptable limits following a calibration of

maximum air change rate for the building. As such the simulation results for this reconstruction were deemed a valid representation of the energy performance of F2050.

The two reconstructions on the whole of F2050 were able to predict the annual energy use of F2050 to within 1.8-11.2% of the measured annual energy use for 2016. Following a calibration of the maximum ach used in the simulation this prediction was reduced to within 1.4-5.2% of measured data. In consideration of the Performance Gap, discussed in academic literature, which has been frequently presented as ranging between 50-70% against measured data [27,53,54,57,65], the simulation results presented in this research have demonstrated a successful improvement on this range even if the models did not meet validation indices.

For the hybrid reconstruction without the use of a voxel filter, the annual energy use of F2050 was predicted to within 8.4% which was improved following a calibration to 3.8%. This simulation did achieve suitable values of the validation indices and demonstrated an improvement on the Performance Gap bolstered by the validated data set. Unfortunately, the simulation results of the hybrid reconstruction that used a $10mm$ voxel filter did not closely match that of the measured data over the year and acceptable validation indices were not achieved either. A calibration study was conducted which determined that a validated energy model couldn't be achieved through only modifying the maximum ach value.

A proof-of-concept solution has been developed to circumnavigate the problems associated with BIM interoperability [10,30,32,33] although it was identified that further development would be required to ensure the robustness of the method. Ideally the full building reconstructions would have achieved acceptable values for their validation indices without the need for additional calibration or hybrid models. However, the process followed in this research has identified limitations that must be overcome in future developments.

Throughout the reconstructed geometry defects were observed which introduced geometric elements that were different to the benchmark model. This resulted in different thermal volumes and heat transfer surfaces between the models. These discrepancies were expected to affect the performance of simulations on the reconstructed geometry when compared against measured data and this research has quantified that effect. In the

traditional retrofit BEMS workflow it was observed that a benchmark model is created first, as per this research, and then different retrofit options were created and simulated by changing the benchmark model. The potential changes that were assessed could include HVAC settings, shading strategies, building extensions and façade modification. While the current maturity of the automated reconstruction process was capable of producing geometry quickly, with a visual likeness to F2050, the defects in the reconstruction would make the assessment of retrofit changes challenging. It was identified that future research should refine the process such that these defects in reconstructed geometry were minimised and that validated energy models can be established consistently prior to assessing different retrofit options. The method requires maturing to the point where a benchmark model for comparison against reconstruction models is no longer required because of a robust confidence in the technique.

One key observation on both reconstructions of F2050 was made that the thermal mass included in each simulation differed. Figure 6.18 has presented the basic difference in main internal partitions between the benchmark and reconstructed geometry in the F2050 BEMS. The reconstructed BEMS only accounted for a single storey which was suitable for in the large workshops that had high vaulted ceilings but this meant that smaller offices and internal partitions located across different storeys were missed from the reconstructions. In the benchmark model the inclusion of this internal thermal mass had two primary effects. The increased floor space across multiple storeys was directly proportional to the internal gains applied to the model which were defined on a per m^2 basis. As such, less internal gains were provided to the reconstructed F2050 BEMS. The extra thermal mass included in these extra internal partitions, floors and ceilings also stored thermal energy more effectively than the air within the main thermal volumes that was replaced at regular intervals. Both of these effects resulted in a greater effective heat loss rate exhibited in the reconstructed BEMS when compared against the benchmark model. As a result, more energy was used to heat the building in the reconstructed BEMS while less energy was required to cool the building over the course of the year. This explained the overestimated energy use in winter months and underestimated energy use in summer months that was observed in the BEMS results on the F2050 reconstructions. As such it was identified that improving the method to include a more representative internal thermal mass would be beneficial to correct this

phenomenon. In addition, additional data acquisition methods could be combined with the captured LIDAR to overcome some of the limitations identified in the presented research method such as image recognition of building features.

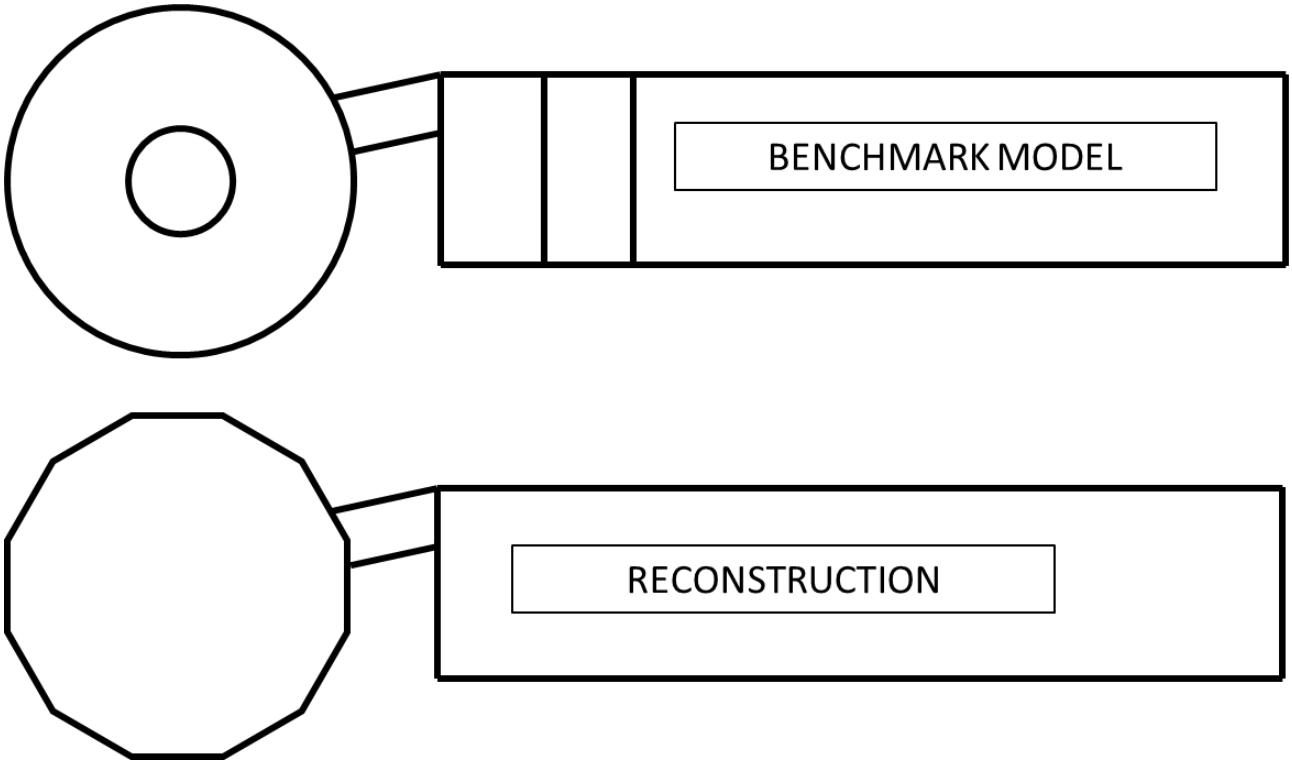


Figure 6.18. Comparison of main internal boundaries between benchmark and reconstructed BEMS

Two reconstructions were produced in this research. One used the raw point cloud data from a set of scans while the second applied a $10mm$ voxel filter to this point cloud data prior to performing the reconstruction process. Both reconstructions were visually identifiable as F2050 but a review of model parameters such as external surface area and thermal volume demonstrated considerable deviations from the benchmark model. While the $10mm$ voxel filtered reconstruction appeared to better capture the geometry of F2050 it was the reconstruction without a voxel filter that achieved better simulation results in predicting the annual energy use of F2050 and validated results when used in a hybrid simulation. Despite these results large volumes were observed to have been missed on this reconstruction. It is likely that although the $10mm$ reconstruction provided a visually better resemblance of the F2050 geometry its increased volume and lack of thermal mass started have a greater effect of the monthly energy predictions and subsequently the annual energy usage which prevented validation of its results against industry guidance. This phenomenon reinforced

the importance of further workflow development to simplify and improve robustness of adding thermal mass into a reconstructed model.

As discussed, the reconstruction method was only capable of detecting horizontal and vertical surfaces to within a defined tolerance. This meant that the roof shapes of F2050, where detected, were considerably simplified into horizontal surfaces thus reducing the thermal volume of the BEMS reconstruction when compared to the benchmark model. The height of these surfaces was also observed to vary across the circular workshop where a consistent height should have been achieved. The complex geometry in this area of the building was probably responsible for the difficulty in this reconstruction as a much greater consistency in ceiling height was observed in the rectangular portion of the F2050 reconstructions. The inability to process angled ceiling surfaces limited the applications of the presented workflow to very specific building designs and it was identified that this should be rectified in further work.

A novelty of this research was to perform a top-down assessment of the Point Cloud reconstruction process on a complex and real-world building. In doing so the process was benchmarked against traditional techniques, identifying its limitations as discussed below, and had to contend with the cluttered nature of a non-sterile and fully operational R&D environment during the building survey. This clutter included high noise generated from reflections off internal Perspex safety partitions and occluded building features due to the presence of equipment or storage areas. Figure 6.19 demonstrates a storage rack for materials that was taken as part of the internal survey of F2050. This was located directly in front of a wall at the end of the rectangular portion of F2050 and was identified as likely to be responsible for the reconstruction defects observed in the Southeast corner of F2050. Being unable to directly observe a large portion of a wall during the survey made planar detection more difficult during the reconstruction.

Figure 6.19. An example of environmental clutter in front of a wall in F2050

Another key difference between the reconstructions and benchmark model was the presence of solar shading around the perimeter of different sections of the F2050 roofs. While this was included in the benchmark model the method employed in the reconstructions, that utilised internal scans, were unable to directly observe the shading and thus could not have been reconstructed even if the software had the capability. While its effect would have been minimal in winter months its presence in summer months would have reduced solar gains into F2050 being more representative of the actual building. As discussed, regarding the modification of models for retrofit assessments, the maturity of the presented workflow was not sufficient to allow a robust addition of features such as the solar shading around F2050.

The research presented used a single case study building of F2050 which provided complex challenges for the automated reconstruction workflow. While a successful reconstruction was achieved for one additional facility no benchmark energy model or measured energy data was available to apply and compare the BEMS workflow. This facility had horizontal ceilings and a very limited number of glazed surfaces which the reconstruction process was more suited to than F2050. It was identified that applying the digital scan survey and automated reconstruction technique to additional case studies, ranging from simple to

complex, would provide greater information on the performance and robustness of the automated reconstruction workflow for a range of building constructions and geometries to identify further limitation for improvement.

A common theme in the limitations that were identified for the automated reconstruction process was the inability to accurately capture some internal surfaces. These included occluded walls, large glazed walls and non-horizontal ceiling surfaces. In addition, the geometry of a building roof located above a suspended ceiling was also not visible to the survey equipment and thus could not be reconstructed appropriately. These limitations were identified as causing defects in the reconstructed geometry which led to the capture of a smaller thermal volume than the benchmark model and geometry discontinuities. The discontinuities made the addition of robust internal partitions impractical too. It was hypothesised that an external building survey would be capable of capturing the maximum extent of a building's thermal volume. It was identified that such a survey would likely have less environmental clutter to contend with which could result in a smoother reconstruction with less defects. Such a development would have improved the ability to add in thermal mass in a simple and robust manner prior to applying BEMS to the reconstructed geometry.

The hardware used in the survey of F2050 was large, bulky and static during each scan. While it could have been applied to the external façade of F2050 it was incapable of accurately capturing the roof geometry of the building. This, in addition to the results of internal scanning in BEMS applications, supported the need to develop a UAV-based survey method to provide survey access to hard to reach building areas and enable capture of the full external thermal envelope. An external UAV-based scan would have had the potential to be conducted at much quicker speeds than the static scans conducted in this research and, through further development of the survey data processing software, enabled the creation of full F2050 reconstructions that had similar thermal volumes and external surface areas when compared with a benchmark model. This observation led to the proof of concept development of a low-cost LiDAR scanner as documented in Chapter 5.

6.9. Summary

This Chapter has presented the results of applying BEMS to automated geometry reconstructions of F2050. Subsection 6.2 reintroduction of the automated reconstruction

outputs presented in Chapter 4. Subsection 6.3 detailed the method employed to define BEMS from reconstructed geometry. Comparisons were made between the reconstructed and benchmark geometry which identified deviations in the thermal volumes and surface areas.

The time taken to produce reconstructed geometry with a visual likeness of F2050 using the automated process presented as part of this research was assessed, see subsection 6.4. BEMS results were obtained for the full F2050 reconstructions, using identical boundary conditions as the benchmark model, in subsection 6.5. Both reconstructions predicted the annual energy use of F2050 within a range of 1.8-11.2%. This demonstrated an improvement on the performance gap reported in the literature which has been found quoted to between 30-90%.

To compensate for defects in the F2050 reconstructions the BEMS were calibrated by increasing their maximum air change value, see subsection 6.6. The calibration reduced the annual energy use prediction range across both reconstructions to 1.4-5.2% of measured data, but acceptable validation index values could not be obtained. Two hybrid reconstructions were investigated by replacing the circular workshop of the reconstructions with the benchmark geometry and validated results were obtained, see subsection 6.7.

Subsection 6.8 concluded the chapter with a thorough discussion of the obtained BEMS results, the identification of key drivers behind the obtained results and how the results were used to recognise limitations of the automated geometry reconstruction method presented as part of the research. Priority areas for future development were identified as improving the inclusion of internal thermal mass of a reconstructed BEMS and the development of an even quicker and improved external scan method.

Chapter 7.

Conclusions & Future Work

7.1. Introduction

Conclusions, based upon the research presented in this thesis, are provided within this Chapter. The research aim of this thesis were introduced in Chapter 1 which was "*Develop a scientific method to remove barriers to urban scale Building Energy Modelling and Simulation (BEMS) using pattern recognition software to extract built forms from large data sets*". Achieving the aim would have demonstrated a more efficient application of BEMS in retrofit projects thus reducing disincentives to its use within industry. Ultimately, this would help contribute to a reduction in emissions from the Built Environment through an improved sustainability toolkit for practitioners.

7.2. Research Conclusions

Each conclusion is numbered against the matching research objectives listed in Chapter 1.

1. A critical review of existing literature was conducted for this research, see Chapter 2. The critical review covered a wide range of topics relevant to the research presented in this thesis including Building Energy Modelling and Simulation (BEMS), Point Cloud Reconstruction and Building Survey Hardware. The review highlighted that within established literature on BEMS the Performance Gap is prevalent leading to a low trust in results generated via BEMS. Regarding survey data capture, two primary means of digital building survey were identified as LiDAR and Photogrammetry; LiDAR is high cost, but direct measurement and Photogrammetry tends to be lower cost by in-direct measurement. The area of reconstruction from Point Clouds is active with most researchers focused on direct reconstruction for BIM. When reviewing hardware developments, no substantial efforts have been made in producing a low-cost LiDAR unit.
2. As described in Chapter 3, a case study facility, F2050, was manually modelled using IES VE [16]. Building geometry was provided as a Revit [34] BIM file which was unable to be exported directly into BEMS. The geometry of F2050 was manually remodelled in Sketchup and imported into IES VE. Through discussions with building owners and relevant industry guidance the boundary conditions for the building were defined and simulations were conducted. The results of the simulation were validated against

measured utility data using ASHRAE Guideline 14-2002 [51]. The validation indices were calculated as 3.66% for NMBE and 9.30% for CVRMSE with the annual energy use predicted to within 3.4% of measured data. Validation of this model provided confidence in the boundary conditions used on future research.

3. A new method was identified and developed to reconstruct internal Point Cloud data specifically for BEMS applications by parsing into a defined gbXML data structure [164], see Chapter 4. The method processed Point Cloud data automatically, detected building features and created a set of adjacent thermal volumes to be used in BEMS. Shortcomings of the technique included its ability to handle noise generated during the survey by occluded surfaces and high levels of reflections. Internal scan techniques are also intrusive for a building and lack of access to certain areas leaves holes in the survey data. This reconstruction technique resulted in an 85% improvement on the workflow for producing BEMS geometry.
4. The reconstructed BEMS geometry was imported in IES VE and boundary conditions verified for use on the benchmark model, in Chapter 3, were applied. Two BEMS reconstructions were used; full reconstruction and hybrid, both of which underwent an air change rate calibration. The results of the full reconstruction simulation highlighted that the ASHRAE Guideline 14-2002 [51] validation criteria could not be met and a large deviation away from measured results was observed. In comparison the hybrid result was able to meet the ASHRAE criteria on one reconstruction and the distribution of results was much closer to measured data. Through use of a Point Cloud hybrid reconstruction technique of building geometry, a set of validated annual BEMS results were obtained to within 3.8% of measured energy data, see Chapter 6. Where validation indices were not obtained of the full F2050 reconstructions the annual energy use was still predicted within a range of 1.4-5.2%. This demonstrated an improvement on the performance gap reported in the literature which has been found quoted to between 30-90%.
5. Open-source software was developed to convert raw Point Cloud data into discrete thermal volumes within a building envelope. This enabled each room to be made-up of thermal zones that could be easily parsed into the defined gbXML data structure, as per objective 3 in this Thesis. The software takes a set of registered Point Clouds which are pre-processed so that the normal direction of each point faces towards

the scanner. This set of Point Clouds are processed through shape recognition algorithms to identify planes such as walls, floors and ceilings. These planes are then grouped together to form individual volumes which can be parsed into a defined gbXML structure. The approach is currently limited to single storeys and flat ceiling planes.

6. A low-cost LiDAR unit prototype was developed to facilitate applying the Point Cloud reconstruction technique to large data sets of external scan data, see Chapter 5. During this research the unit was able to record a 2D slice of a room perimeter and steps were taken to develop this to 3D environment capture. The raw cost of materials used in the prototype at the end of this research totalled £336.50. This is an order of magnitude less than the LiDAR units currently available at the lower end of the market presenting an opportunity for further development. The LiDAR prototype was developed to combine LiDAR data with acceleration, gyroscopic and magnetic data to produce a mobile LiDAR unit capable of capturing a 3D scene. This data fusion was not achieved during the research but the hardware limitations preventing success were identified for improvements in future development.

Based on the conclusions set out above, the overarching aim of this research, "*Develop a scientific method to remove barriers to urban scale Building Energy Modelling and Simulation (BEMS) using pattern recognition software to extract built forms from large data sets*", as originally stated in Chapter 1, has been achieved.

7.3. Original Contribution to Knowledge

In conducting the research described in this thesis and arriving at the conclusions outlined in subsection 7.2 the following original contributions to knowledge are claimed;

1. BEMS has been conducted on reconstructed geometry of a real-world facility that was acquired using a laser scanner survey. Validated BEMS results were achieved using a hybrid approach.
2. Software has been developed such that point cloud data has been automatically processed into discrete finite volumes to represent thermal envelopes.

3. An 85% reduction in time taken to produce building geometry for BEMS was demonstrated on the pure reconstruction when compared with manual geometry creation methods.
4. A 1st generation prototype for a low-cost LiDAR scanner has been developed that demonstrated substantial cost savings over existing commercial offerings.

7.4. Future Work

In conducting the research, several areas were identified as worthy of further research to extend that presented within this thesis in the future. These include but are not limited to the following;

1. Performing the reconstruction technique on additional case study buildings would be of considerable benefit in further highlighting the strengths and weaknesses of the technique in a range of environments.
2. Modification of the technique to accept external scan data may improve the capture of the full thermal volume of any case study building. This should minimise the effects of occluded structures (walls, ceilings & floors) caused by environmental clutter and poor access to certain facility areas during the survey.
3. LiDAR was selected as the technology of choice for this research primarily for its direct measurement capability. Thus, a comparison of building reconstruction techniques using LiDAR and Photogrammetry should be conducted to see the relative merits of each approach of the BEMS workflow.
4. As noted in the research, opening detection on large Point Cloud datasets proved difficult; especially due to the noise generated in the Point Cloud near glazed surfaces. A logical step in improving on this short coming would be to apply 3D Artificial Intelligence image recognition techniques to a Point Cloud, or Photogrammetric, dataset to establish its efficacy.
5. Development of a plug-and-play low-cost LiDAR system, that builds upon the prototype presented in this research, for operation via UAV would be extremely beneficial. Such a development would make the survey technique more accessible to the wider industrial and academic community thus furthering the utility of the technology.

6. The use of multiple data capture streams to overcome limitations of a single data capture method such as fusing LIDAR reconstruction with image recognition so that a hybrid model is no longer required to achieved validated simulation results.

While the research outlined in this thesis has met its aims and objectives completion of the future work outlined above will progress this research area. Doing so will improve the BEMS retrofit workflow and thus remove disincentives to the application of BEMS for retrofit projects. Improving on these research areas will help to contribute towards a reduction in emissions from our built environment for a more sustainable society. Future plans are to pursue these areas using the research documented within this thesis as a sound foundation from which to extend the knowledge base within academia and industry.

References

[1] Tymkow P, Tassou S, Kolokotroni M, Jouhara H. Building services design for energy efficient buildings. 2013. doi:10.4324/9780203840733.

[2] Roberts S. Altering existing buildings in the UK. Energy Policy 2008;36:4482–6. doi:10.1016/j.enpol.2008.09.023.

[3] Centre for Low Carbon Futures. The Retrofit Challenge: Delivering Low Carbon Buildings. 2011.

[4] Department for Business Energy & Industrial Strategy. Energy Consumption in the UK. 2016.

[5] European Commission. Communiaction from the Commission: A Roadmap for moving to a competitive low carbon economy in 2050. 2011. doi:10.1002/jsc.572.

[6] HM Government of the United Kingdom. Climate Change Act 2008. HM Government of the United Kingdom; 2008.

[7] U.S. Energy Information Administration (U.S. EIA). International Energy Outlook 2016. Washington D.C.: 2016.

[8] Eastman M C, Fisher D, Lafue G, Lividini J, Stoker D, Yessios C. An Outline of the Building Description System. Carnegie-Mellon Univ 1974. doi:10.1017/CBO9781107415324.004.

[9] Garwood TL, Hughes BR, Oates MR, O'Connor D, Hughes R. A review of energy simulation tools for the manufacturing sector. Renew Sustain Energy Rev 2018;81:895–911. doi:10.1016/j.rser.2017.08.063.

[10] Bahar YN, Pere C, Landrieu J, Nicolle C. A Thermal Simulation Tool for Building and Its Interoperability through the Building Information Modeling (BIM) Platform. Buildings 2013;3:380–98. doi:10.3390/buildings3020380.

[11] DesignBuilder Software Ltd. DesignBuilder n.d. https://www.designbuilder.co.uk/ (accessed April 5, 2018).

[12] Department of Energy. EnergyPlus n.d. https://energyplus.net/ (accessed April 5, 2018).

[13] Department of Energy. eQUEST n.d. http://www.doe2.com/equest/ (accessed November 23, 2016).

[14] Autodesk. Green Building Studio n.d. https://gbs.autodesk.com/GBS/ (accessed November 2, 2016).

[15] International Building Physics Toolbox. International Building Physics Toolbox n.d. http://www.ibpt.org/index.html (accessed November 23, 2016).

[16] IES. Integrated Environmental Solutions Virtual Environment (IES VE) n.d. https://www.iesve.com/ (accessed June 26, 2018).

[17] Lloyd's Register. Building information modelling (BIM) Certification n.d. http://www.lr.org/en/utilities-building-assurance-schemes/building-information-modelling/ (accessed October 11, 2016).

[18] Novel C, Keriven R, Gerpho PG. Comparing Aerial Photogrammetry and 3D Laser Scanning Methods for Creating 3D Models of Complex Objects A Bentley Systems White Paper Comparing Aerial Photogrammetry and 3D Laser Scanning Methods for Creating 3D Models of Complex Objects. 2016.

[19] DJI - The World Leader in Camera Drones/Quadcopters for Aerial Photography n.d. https://www.dji.com/uk?from=store_top_nav (accessed February 26, 2019).

[20] Previtali M, Barazzetti L, Brumana R, Cuca B, Oreni D, Roncoroni F, et al. Automatic façade modelling using point cloud data for energy-efficient retrofitting. Appl Geomatics 2014;6:95–113. doi:10.1007/s12518-014-0129-9.

[21] Yang M-D, Su T-C, Lin H-Y. Fusion of Infrared Thermal Image and Visible Image for 3D Thermal Model Reconstruction Using Smartphone Sensors 2018. doi:10.3390/s18072003.

[22] Ocaña-Levario SJ, Carreño-Alvarado EP, Ayala-Cabrera D, Izquierdo J. GPR image analysis to locate water leaks from buried pipes by applying variance filters. J Appl Geophys 2018;152:236–47. doi:10.1016/j.jappgeo.2018.03.025.

[23] Atef A, Zayed T, Hawari A, Khader M, Moselhi O. Multi-tier method using infrared photography and GPR to detect and locate water leaks. Autom Constr 2016;61:162–70.

doi:10.1016/j.autcon.2015.10.006.

[24] Volk R, Stengel J, Schultmann F. Building Information Modeling (BIM) for existing buildings - Literature review and future needs. Autom Constr 2014;38:109–27. doi:10.1016/j.autcon.2013.10.023.

[25] Gao H, Koch C, Wu Y. Building information modelling based building energy modelling: A review. Appl Energy 2019;238:320–43. doi:10.1016/j.apenergy.2019.01.032.

[26] Farzaneh A, Monfet D, Forgues D. Review of using Building Information Modeling for building energy modeling during the design process. J Build Eng 2019;23:127–35. doi:10.1016/j.jobe.2019.01.029.

[27] De Wilde P. The gap between predicted and measured energy performance of buildings: A framework for investigation. Autom Construcrtion 2014;41:40–9. doi:10.1016/j.autcon.2014.02.009.

[28] Gourlis G, Kovacic I. Building Information Modelling for analysis of energy efficient industrial buildings - A case study. Renew Sustain Energy Rev 2017;68:953–63. doi:10.1016/j.rser.2016.02.009.

[29] Kamel E, Memari AM. Review of BIM's application in energy simulation: Tools, issues, and solutions. Autom Constr 2019;97:164–80. doi:10.1016/j.autcon.2018.11.008.

[30] O'Donnell JT, Maile T, Rose C, Mrazović N, Morrissey E, Regnier C, et al. Transforming BIM to BEM: Generation of Building Geometry for the NASA Ames Sustainability Base BIM. 2013.

[31] Graphisoft. About ARCHICAD — A 3D architectural BIM software for design & modeling n.d. http://www.graphisoft.com/archicad/ (accessed May 24, 2017).

[32] Reeves T, Olbina S, Issa RRA. Guidelines for Using Building Information Modeling for Energy Analysis of Buildings. Buildings 2015;5:1361–88. doi:10.3390/buildings5041361.

[33] Jeong W, Kim JB, Clayton MJ, Haberl JS, Yan W. Translating Building Information Modeling to Building Energy Modeling Using Model View Definition. Sci World J 2014:21. doi:10.1155/2014/638276.

[34] Autodesk. Revit 2017. http://www.autodesk.co.uk/products/revit-family/overview

(accessed April 5, 2017).

[35] Lawrence Berkeley National Laboratory. Modelica Buildings library n.d. http://simulationresearch.lbl.gov/modelica/ (accessed November 2, 2016).

[36] Kim JB, Jeong W, Clayton MJ, Haberl JS, Yan W. Developing a physical BIM library for building thermal energy simulation. Autom Constr 2015;50:16–28. doi:10.1016/j.autcon.2014.10.011.

[37] ASHRAE. ASHRAE 140-2001 Standard Method of Test for the Evaluation of Building Energy Analysis Computer Programs. ASHRAE 2001.

[38] Jeong W, Kim JB, Clayton MJ, Haberl JS, Yan W. A framework to integrate object-oriented physical modelling with building information modelling for building thermal simulation. J Build Perform Simul 2016;9:50–69. doi:10.1080/19401493.2014.993709.

[39] Jeong WS, Kim KH. A Performance Evaluation of the BIM-Based Object-Oriented Physical Modeling Technique for Building Thermal Simulations: A Comparative Case Study. Sustain 2016;8:1–27. doi:10.3390/su8070648.

[40] Guzmán Garcia E, Zhu Z. Interoperability from building design to building energy modeling. J Build Eng 2015;1:33–41. doi:10.1016/j.jobe.2015.03.001.

[41] Kim H, Shen Z, Kim I, Kim K, Stumpf A, Yu J. BIM IFC information mapping to building energy analysis (BEA) model with manually extended material information. Autom Constr 2016;68:183–93. doi:10.1016/j.autcon.2016.04.002.

[42] Abanda F, Vidalakis C, Oti A, Tah J. A critical analysis of Building Information Modelling systems used in construction projects. Adv Eng Softw 2015;90:183–201. doi:10.1016/j.advengsoft.2015.08.009.

[43] Ahn K-UU, Kim Y-JJ, Park C-SS, Kim I, Lee K. BIM interface for full vs. semi-automated building energy simulation. Energy Build 2014;68:671–8. doi:10.1016/j.enbuild.2013.08.063.

[44] Pezeshki Z, Soleimani A, Darabi A. Application of BEM and using BIM database for BEM: A review. J Build Eng 2019;23:1–17. doi:10.1016/j.jobe.2019.01.021.

[45] Gourlis G, Kovacic I. A study on building performance analysis for energy retrofit of

existing industrial facilities. Appl Energy 2016;184:1389–99. doi:10.1016/j.apenergy.2016.03.104.

[46] Katunsky D, Korjenic A, Katunska J, Lopusniak M, Korjenic S, Doroudiani S. Analysis of thermal energy demand and saving in industrial buildings: A case study in Slovakia. Build Environ 2013;67:138–46. doi:10.1016/j.buildenv.2013.05.014.

[47] Sun S, Kensek K, Noble D, Schiler M. A method of probabilistic risk assessment for energy performance and cost using building energy simulation. Energy Build 2016;110:1–12. doi:10.1016/j.enbuild.2015.09.070.

[48] Eguaras-Martínez M, Vidaurre-Arbizu M, Martín-Gómez C. Simulation and evaluation of Building Information Modeling in a real pilot site. Appl Energy 2014;114:475–84. doi:10.1016/j.apenergy.2013.09.047.

[49] Tian W, Heo Y, de Wilde P, Li Z, Yan D, Park CS, et al. A review of uncertainty analysis in building energy assessment. Renew Sustain Energy Rev 2018;93:285–301. doi:10.1016/j.rser.2018.05.029.

[50] Ascione F, Ceroni F, De Masi RF, de' Rossi F, Rosaria Pecce M. Historical buildings: Multidisciplinary approach to structural/energy diagnosis and performance assessment. Appl Energy 2017;185:1517–28. doi:10.1016/j.apenergy.2015.11.089.

[51] ASHRAE. Guideline 14-2002 Measurement of Energy and Demand Savings. 2002.

[52] Wang X, Kendrick C, Ogden R, Walliman N, Baiche B. A case study on energy consumption and overheating for a UK industrial building with rooflights. Appl Energy 2013;104:337–44. doi:10.1016/j.apenergy.2012.10.047.

[53] Khoury J, Alameddine Z, Hollmuller P. Understanding and bridging the energy performance gap in building retrofit. Energy Procedia 2017;122:217–22. doi:10.1016/j.egypro.2017.07.348.

[54] Burman E, Mumovic D, Kimpian J. Towards measurement and verification of energy performance under the framework of the European directive for energy performance of buildings. Energy 2014;77:153–63. doi:10.1016/j.energy.2014.05.102.

[55] Chaudhary G, New J, Sanyal J, Im P, O'neill Z, Garg V. Evaluation of "Autotune"

calibration against manual calibration of building energy models. Appl Energy 2016;182:115–34. doi:10.1016/j.apenergy.2016.08.073.

[56] Directive 2002/91/EC. Directive 2002/91/EC of the European Parliament and of the Council of 16 December 2002 on the energy performance of buildings. Off J Eur Union 2002. doi:10.1039/ap9842100196.

[57] Menezes AC, Cripps A, Bouchlaghem D, Buswell R. Predicted vs. actual energy performance of non-domestic buildings: Using post-occupancy evaluation data to reduce the performance gap. Appl Energy 2012;97. doi:10.1016/j.apenergy.2011.11.075.

[58] Janda KB. Buildings don't use energy: people do. Archit Sci Rev 2011;54:15–22. doi:10.3763/asre.2009.0050.

[59] Karlsson F, Rohdin PhD-student P, Persson M. Measured and predicted energy demand of a low energy building: important aspects when using Building Energy Simulation n.d. doi:10.1177/0143624407077393.

[60] Turner C, Analyst S, Frankel M. Energy Performance of LEED ® for New Construction Buildings. 2008. doi:10.1016/s0141-0229(99)00121-0.

[61] Newsham GR, Mancini S, Birt BJ. Do LEED-certified buildings save energy? Yes, but... Energy Build 2009. doi:10.1016/j.enbuild.2009.03.014.

[62] Scofield JH. Do LEED-certified buildings save energy? Not really... Energy Build 2009. doi:10.1016/j.enbuild.2009.08.006.

[63] Harish V, Kumar A. A review on modeling and simulation of building energy systems. Renew Sustain Energy Rev 2016;56:1272–92. doi:10.1016/j.rser.2015.12.040.

[64] Anil EB, Tang P, Akinci B, Huber D. Deviation analysis method for the assessment of the quality of the as-is Building Information Models generated from point cloud data. Autom Constr 2013;35:507–16. doi:10.1016/j.autcon.2013.06.003.

[65] Coakley D, Raftery P, Keane M. A review of methods to match building energy simulation models to measured data. Renew Sustain Energy Rev 2014;37:123–41. doi:10.1016/j.rser.2014.05.007.

[66] Thermal Energy System Specialists LLC. TRNSYS : Transient System Simulation Tool

n.d. http://www.trnsys.com/ (accessed November 23, 2016).

[67] University of Strathclyde. ESP-r n.d. http://www.esru.strath.ac.uk/Programs/ESP-r.htm (accessed November 18, 2016).

[68] Mustafaraj G, Marini D, Costa A, Keane M. Model calibration for building energy efficiency simulation. Appl Energy 2014;130:72–85. doi:10.1016/j.apenergy.2014.05.019.

[69] Bertagnolio S, Randaxhe F, Lemort V. Evidence-based calibration of a building energy simulation model: Application to an office building in Belgium. Int. Conf. Enhanc. Build. Oper., 2012.

[70] Bertagnolio S. Evidence-based calibration for efficient building energy services. 2012.

[71] Gerlich V, Sulovská K, Zálešák M. COMSOL Multiphysics validation as simulation software for heat transfer calculation in buildings: Building simulation software validation. Meas J Int Meas Confed 2013;46:2003–12. doi:10.1016/j.measurement.2013.02.020.

[72] COMSOL Multiphysics® Modeling Software n.d. https://uk.comsol.com/ (accessed February 18, 2019).

[73] New JR, Sanyal J, Bhandari M, Shrestha S. AUTOTUNE E + BUILDING ENERGY MODELS. Proc SimBuild 2012:270–8.

[74] Yuan J, Nian V, Su B, Meng Q. A simultaneous calibration and parameter ranking method for building energy models. Appl Energy 2017;206:657–66. doi:10.1016/j.apenergy.2017.08.220.

[75] Kim YS, Heidarinejad M, Dahlhausen M, Srebric J. Building energy model calibration with schedules derived from electricity use data. Appl Energy 2017;190:997–1007. doi:10.1016/j.apenergy.2016.12.167.

[76] Enríquez R, Jiménez MJ, Heras MR. Towards non-intrusive thermal load Monitoring of buildings: BES calibration. Appl Energy 2017;191:44–54. doi:10.1016/j.apenergy.2017.01.050.

[77] Harmer LC, Henze GP. Using calibrated energy models for building commissioning and load prediction. Energy Build 2015;92:204–15. doi:10.1016/j.enbuild.2014.10.078.

[78] Li Q, Augenbroe G, Brown J. Assessment of linear emulators in lightweight Bayesian calibration of dynamic building energy models for parameter estimation and performance prediction. Energy Build 2016;124:194–202. doi:10.1016/j.enbuild.2016.04.025.

[79] Roberti F, Oberegger UF, Gasparella A. Calibrating historic building energy models to hourly indoor air and surface temperatures: Methodology and case study. Energy Build 2015;108:236–43. doi:10.1016/j.enbuild.2015.09.010.

[80] Sun K, Hong T, Taylor-Lange SC, Piette MA. A pattern-based automated approach to building energy model calibration. Appl Energy 2016;165:214–24. doi:10.1016/j.apenergy.2015.12.026.

[81] Wang C, Cho YK. Application of As-built Data in Building Retrofit Decision Making Process. Procedia Eng 2015;118:902–8. doi:10.1016/j.proeng.2015.08.529.

[82] Cho YK, Ham Y, Golpavar-Fard M. 3D as-is building energy modeling and diagnostics: A review of the state-of-the-art. Adv Eng Informatics 2015;29:184–95. doi:10.1016/j.aei.2015.03.004.

[83] Tang P, Huber D, Akinci B, Lipman R, Lytle A. Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. Autom Constr 2010;19:829–43. doi:10.1016/j.autcon.2010.06.007.

[84] Poullis C. A Framework for Automatic Modeling from Point Cloud Data. IEEE Trans Pattern Anal Mach Intell 2013;35:2563–75. doi:10.1109/TPAMI.2013.64.

[85] Tooke TR, van der Laan M, Coops NC. Mapping demand for residential building thermal energy services using airborne LiDAR. Appl Energy 2014;127:125–34. doi:10.1016/j.apenergy.2014.03.035.

[86] Nageler P, Zahrer G, Heimrath R, Mach T, Mauthner F, Leusbrock I, et al. Novel validated method for GIS based automated dynamic urban building energy simulations. Energy 2017;139:142–54. doi:10.1016/j.energy.2017.07.151.

[87] OpenStreetMap. OpenStreetMap 2019. https://www.openstreetmap.org (accessed February 25, 2019).

[88] Chen Y, Hong T, Piette MA. Automatic generation and simulation of urban building energy models based on city datasets for city-scale building retrofit analysis. Appl Energy 2017;205:323–35. doi:10.1016/j.apenergy.2017.07.128.

[89] Chen Y, Hong T. Impacts of building geometry modeling methods on the simulation results of urban building energy models. Appl Energy 2018;215:717–35. doi:10.1016/j.apenergy.2018.02.073.

[90] Ma J, Cheng JCP. Estimation of the building energy use intensity in the urban scale by integrating GIS and big data technology. Appl Energy 2016;183:182–92. doi:10.1016/j.apenergy.2016.08.079.

[91] Buffat R, Froemelt A, Heeren N, Raubal M, Hellweg S. Big data GIS analysis for novel approaches in building stock modelling. Appl Energy 2017;208:277–90. doi:10.1016/j.apenergy.2017.10.041.

[92] Buyukdemircioglu M, Kocaman S, Isikdag U. Semi-Automatic 3D City Model Generation from Large-Format Aerial Images. ISPRS Int J Geo-Information 2018;7:339. doi:10.3390/ijgi7090339.

[93] Dorninger P, Pfeifer N. A Comprehensive Automated 3D Approach for Building Extraction, Reconstruction, and Regularization from Airborne Laser Scanning Point Clouds. Sensors 2008;8:7323–43. doi:10.3390/s8117323.

[94] Verma V, Kumar R, Hsu S. 3D building detection and modeling from aerial LIDAR data. Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2, 2006, p. 2213–20. doi:10.1109/CVPR.2006.12.

[95] De Angelis E, Ciribini ALC, Tagliabue LC, Paneroni M. The Brescia Smart Campus demonstrator. Renovation toward a zero energy classroom building. Procedia Eng 2015;118:735–43. doi:10.1016/j.proeng.2015.08.508.

[96] Department of Energy. OpenStudio n.d. https://www.openstudio.net/ (accessed November 21, 2016).

[97] Trimble Inc. SketchUp n.d. http://www.sketchup.com/ (accessed April 5, 2018).

[98] Wang C, Cho YK. Automated gbXML-Based Building Model Creation for Thermal

Building Simulation. Proc - 2014 Int Conf 3D Vis Work 3DV 2014 2015:111–7. doi:10.1109/3DV.2014.109.

[99] Green Building XML (gbXML) Schema Inc. Green Building XML (gbXML) Schema n.d. http://www.gbxml.org/Schema_Current_GreenBuildingXML_gbXML (accessed April 5, 2018).

[100] Wang C, Cho YK, Kim C. Automatic BIM component extraction from point clouds of existing buildings for sustainability applications. Autom Constr 2015;56:1–13. doi:10.1016/j.autcon.2015.04.001.

[101] Thomson C, Boehm J. Automatic Geometry Generation from Point Clouds for BIM. Remote Sens 2015;7:11753–75. doi:10.3390/rs70911753.

[102] Armesto J, Sánchez-Villanueva C, Patiño-Cambeiro F, Patiño-Barbeito F. Indoor Multi-Sensor Acquisition System for Projects on Energy Renovation of Buildings. Sensors 2016;16:785–99. doi:10.3390/s16060785.

[103] Rottensteiner F, Sohn G, Gerke M, Wegner JD, Breitkopf U, Jung J. Results of the ISPRS benchmark on urban object detection and 3D building reconstruction 2014. doi:10.1016/j.isprsjprs.2013.10.004.

[104] Bosché F, Ahmed M, Turkan Y, Haas CT, Haas R. The value of integrating Scan-to-BIM and Scan-vs-BIM techniques for construction monitoring using laser scanning and BIM: The case of cylindrical MEP components. Autom Constr 2015;49:201–13. doi:10.1016/j.autcon.2014.05.014.

[105] Lagüela S, Martínez J, Armesto J, Arias P. Energy efficiency studies through 3D laser scanning and thermographic technologies. Energy Build 2011;43:1216–21. doi:10.1016/j.enbuild.2010.12.031.

[106] Lagüela S, Díaz-Vilariño L, Martínez J, Armesto J. Automatic thermographic and RGB texture of as-built BIM for energy rehabilitation purposes. Autom Constr 2013;31:230–40. doi:10.1016/j.autcon.2012.12.013.

[107] Elberink SO, Vosselman G. ISPRS Journal of Photogrammetry and Remote Sensing Quality analysis on 3D building models reconstructed from airborne laser scanning data. ISPRS J Photogramm Remote Sens 2011;66:157–65.

doi:10.1016/j.isprsjprs.2010.09.009.

[108] Sinha A, Papadakis P, Elara MR. A staircase detection method for 3D point clouds. 2014 13th Int Conf Control Autom Robot Vis 2014:652–6. doi:10.1109/ICARCV.2014.7064381.

[109] Xiong X, Adan A, Akinci B, Huber D. Automatic creation of semantically rich 3D building models from laser scanner data. Autom Constr 2013;31:325–37. doi:10.1016/j.autcon.2012.10.006.

[110] Zhou Z, Gong J. Automated residential building detection from airborne LiDAR data with deep neural networks. Adv Eng Informatics 2018;36:229–41. doi:10.1016/j.aei.2018.04.002.

[111] Ochmann S, Vock R, Wessel R, Klein R. Automatic reconstruction of parametric building models from indoor point clouds. Comput Graph 2016;54:94–103. doi:10.1016/j.cag.2015.07.008.

[112] DURAARK. DURAARK – Durable Architectural Knowledge n.d. http://duraark.eu/ (accessed April 5, 2018).

[113] Sony S, Laventure S, Sadhu A. A literature review of next-generation smart sensing technology in structural health monitoring. Struct Control Heal Monit 2019;26:1–22. doi:10.1002/stc.2321.

[114] Jordan S, Moore J, Hovet S, Box J, Perry J, Kirsche K, et al. State-of-the-art technologies for UAV inspections. IET Radar, Sonar Navig 2017;12:151–64. doi:10.1049/iet-rsn.2017.0251.

[115] Roca D, Lagüela S, Díaz-Vilariño L, Armesto J, Arias P. Low-cost aerial unit for outdoor inspection of building façades. Autom Constr 2013;36:128–35. doi:10.1016/j.autcon.2013.08.020.

[116] Roca D, Martínez-Sánchez J, Lagüela S, Arias P. Novel Aerial 3D Mapping System Based on UAV Platforms and 2D Laser Scanners. J Sensors 2016;2016:1–8. doi:10.1155/2016/4158370.

[117] Kaldén P, Sternå E. Development of a low-cost laser range-finder (LIDAR) n.d.

[118] Madgwick SOH. An efficient orientation filter for inertial and inertial/magnetic sensor

arrays. 2010.

[119] Madgwick SOH, Harrison AJL, Vaidyanathan R. Estimation of IMU and MARG orientation using a gradient descent algorithm. IEEE Int. Conf. Rehabil. Robot., Zurich, Switzerland: 2011. doi:10.1109/ICORR.2011.5975346.

[120] Kalman RE. A New Approach to Linear Filtering and Prediction Problems. 1960.

[121] Mahony R, Hamel T, Pflimlin J-M, Pflimlin Nonlinear J-M, Mahony R. Complementary Filters on the Special Orthogonal Group. IEEE Trans Autom Control Inst Electr Electron Eng 2008;53:1203–17. doi:10.1109/TAC.2008.923738.

[122] CIBSE. Guide A - Environmental Design. 2016.

[123] Çengel YA, Boles MA. Thermodynamics: an Engineering Approach, 6th Edition. 2007.

[124] Oates MR. A new approach to modelling process and building energy flows in manufacturing industry. De Montfort University, Leicester, 2013.

[125] CIBSE, Day T. TM41: Degree-days: theory and application. 2006.

[126] Fabrizio E, Monetti V. Methodologies and advancements in the calibration of building energy models. Energies 2015;8. doi:10.3390/en8042548.

[127] Booth AT, Choudhary R, Spiegelhalter DJ. Handling uncertainty in housing stock models. Build Environ 2012;48:35–47. doi:10.1016/j.buildenv.2011.08.016.

[128] Ramos Ruiz G, Fernández Bandera C. Validation of Calibrated Energy Models: Common Errors. Energies 2017;10:1587–605. doi:10.3390/en10101587.

[129] Wang L, Mathew P, Pang X. Uncertainties in energy consumption introduced by building operations and weather for a medium-size office building. Energy Build 2012;53:152–8. doi:10.1016/j.enbuild.2012.06.017.

[130] Macdonald IA. Quantifying the Effects of Uncertainty in Building Simulation. University of Strathclyde, 2002.

[131] Huang P, Huang G, Wang Y. HVAC system design under peak load prediction uncertainty using multiple-criterion decision making technique. Energy Build 2015;91:26–36. doi:10.1016/j.enbuild.2015.01.026.

[132] Breesch H, Janssens A. Performance evaluation of passive cooling in office buildings based on uncertainty and sensitivity analysis. Sol Energy 2010;84:1453–67. doi:10.1016/j.solener.2010.05.008.

[133] Corrado V, Mechri HE. Uncertainty and Sensitivity Analysis for Building Energy Rating. J Build Phys 2009;33. doi:10.1177/1744259109104884.

[134] Granderson J, Touzani S, Custodio C, Sohn M, Fernandes S. Assessment of Automated Measurement and Verification MEthods. Berkeley: 2015.

[135] Reddy TA, Maor I. Procedures for Reconciling Computer-Calculated Results With Measured Energy Data ASHRAE Research Project 1051- RP. ASHRAE Res Proj 1051- RP 2006.

[136] CIBSE, Field J. TM46: Energy benchmarks. London: 2008.

[137] The University of Sheffield. Land deal supports advanced manufacturing ambitions - Latest - News - The University of Sheffield n.d. https://www.sheffield.ac.uk/news/nr/sheffield-business-park-advanced-manufacturing-1.357465 (accessed June 20, 2019).

[138] Fitzgerald V. Raising the roof on Factory 2050. Manuf 2015. https://www.themanufacturer.com/articles/raising-the-roof-on-factory-2050/ (accessed November 16, 2019).

[139] MET Office. Weather and climate change n.d. http://www.metoffice.gov.uk/ (accessed April 5, 2018).

[140] IES VE. IES VE Plugin for Sketchup n.d. http://www.iesve.com/software/interoperability/sketchup (accessed April 5, 2018).

[141] Department of Energy. Blower Door Tests n.d. https://www.energy.gov/energysaver/blower-door-tests (accessed April 3, 2019).

[142] HM Government. The Building Regulations 2010: Part L1A Conservation of fuel and power in new dwellings, 2013 Edition. n.d.

[143] Johnston D, Stafford A. Estimating the background ventilation rates in new-build UK dwellings-Is n 50 /20 appropriate? Indoor Built Environ 2016;26:502–13.

doi:10.1177/1420326X15626234.

[144] Ridley I, Fox J, Oreszczyn T, Hong SH. The Impact of Replacement Windows on Air Infiltration and Indoor Air Quality in Dwellings. Int J Vent 2016;1:209–18. doi:10.1080/14733315.2003.11683636.

[145] Liddament M. A Guide to Energy Efficient Ventilation. Warwick; Coventry; Internation Energy Agency: Air Infiltration and Vantilation Centre; 1996.

[146] BS 5925. Ventilation principles and designing for 1991.

[147] Goyer GG, Watson R. The Laser and its Application to Meteorology. Bull Am Meteorol Soc 1963:564–70. doi:https://doi.org/10.1175/1520-0477-44.9.564.

[148] Leica Geosystems. Leica Geosystems HomePage n.d. https://leica-geosystems.com/en-GB (accessed February 22, 2019).

[149] FARO Technologies UK Ltd. 3D Measurement & Realization Technology n.d. https://www.faro.com/en-gb/ (accessed February 22, 2019).

[150] Leica Geosystems. Leica ScanStation P20 Industry's Best Performing Ultra-High Speed Scanner. n.d.

[151] Leica Geosystems. Leica Cyclone - 3D Point Cloud Processing Software - Leica Geosystems - HDS n.d. http://hds.leica-geosystems.com/en/leica-cyclone_6515.htm (accessed March 7, 2017).

[152] ASTM. ASTM E2807-11 Standard Specification for 3D Imaging Data Exchange, Version 1.0. 2011.

[153] Huber D. The ASTM E57 File Format for 3D Imaging Data Exchange. n.d.

[154] Hilbert P. E57 Processor. Github, Inc 2016. https://github.com/paulhilbert/e57_processor/tree/master/src.

[155] ASTM E57.04 Data Interoperability Committee. libE57: Software Tools for Managing E57 files (ASTM E2807 standard) n.d. http://www.libe57.org/index.html (accessed March 13, 2019).

[156] Dawes B, Abrahams D. Boost C++ Libraries n.d. https://www.boost.org/ (accessed

March 13, 2019).

[157] Jacob B, Guennebaud G. Eigen C++ Library n.d. http://eigen.tuxfamily.org/index.php?title=Main_Page (accessed March 13, 2019).

[158] Rusu RB, Cousins S. 3D is here: Point Cloud Library (PCL). n.d.

[159] Rusu RB. Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments. KI - Künstliche Intelligenz 2010. doi:10.1007/s13218-010-0059-6.

[160] The Computational Geometry Algorithms Library (CGAL) n.d. https://www.cgal.org/ (accessed March 14, 2019).

[161] Hilbert P. e57_pcl. Github, Inc 2015. https://github.com/paulhilbert/e57_pcl.

[162] CodeSynthesis. XSD - XML Data Binding for C++ n.d. https://www.codesynthesis.com/products/xsd/ (accessed March 14, 2019).

[163] Garwood TL, Hughes BR, O'Connor D, Calautit JK, Oates MR, Hodgson T. Geometry Extraction for High Resolution Building Energy Modelling Applications from Point Cloud Data: A Case Study of a Factory Facility. In: Yan J, Wu J, Li H, editors. Energy Procedia, Cardiff: Elsevier Ltd.; 2017, p. 1805–10. doi:10.1016/j.egypro.2017.12.567.

[164] Garwood TL, Hughes BR, O'Connor D, Calautit JK, Oates MR, Hodgson T. A framework for producing gbXML building geometry from Point Clouds for accurate and efficient Building Energy Modelling. Appl Energy 2018;224:527–37. doi:10.1016/j.apenergy.2018.04.046.

[165] DDS-CAD Viewer n.d. http://www.dds-cad.net/downloads/dds-cad-viewer/ (accessed April 5, 2018).

[166] Arduino. No Title 2019. https://www.arduino.cc/.

[167] Arduino Uno Rev3 n.d. https://store.arduino.cc/arduino-uno-rev3 (accessed June 20, 2019).

[168] LIDAR-Lite v3HP | Garmin n.d. https://buy.garmin.com/en-US/US/p/578152 (accessed March 26, 2019).

[169] Garmin. LIDAR-LITE V3HP OPERATION MANUAL AND TECHNICAL SPECIFICATIONS.

2018.

[170] Grau A. Don’t worry be creative | A blog on projects with robotics, computer vision, 3D printing, microcontrollers, car diagnostics, localization & mapping, digital filters, LiDAR and more n.d. http://grauonline.de/wordpress/ (accessed March 26, 2019).

[171] Omron. EE-SX1320 Ultra-Compact Slot / SMD Type (Slot width: 2 mm) Specifications. n.d.

[172] Zhi R. A Drift Eliminated Attitude & Position Estimation Algorithm In 3D. n.d.

[173] GPS.gov: GPS Accuracy 2018. https://www.gps.gov/systems/gps/performance/accuracy/ (accessed August 21, 2018).

[174] Taylor BR. Bolder Flight Systems - MPU-9250. GitHub, Inc 2018. https://github.com/bolderflight/MPU9250.

[175] Madgwick SOH. Gait-Tracking-With-x-IMU. GitHub, Inc 2017. https://github.com/xioTechnologies/Gait-Tracking-With-x-IMU (accessed April 11, 2019).

[176] Garwood TL, Hughes BR, O’Connor D, Oates MR. Automatic Building Geometry Creation From Point Clouds For Simulating Industrial Building Energy Use. Proc. BSO 2018 4th Build. Simul. Optim. Conf. Cambridge, UK 11-12 Sept. 2018, Cambridge, UK: 2018, p. 396–401.

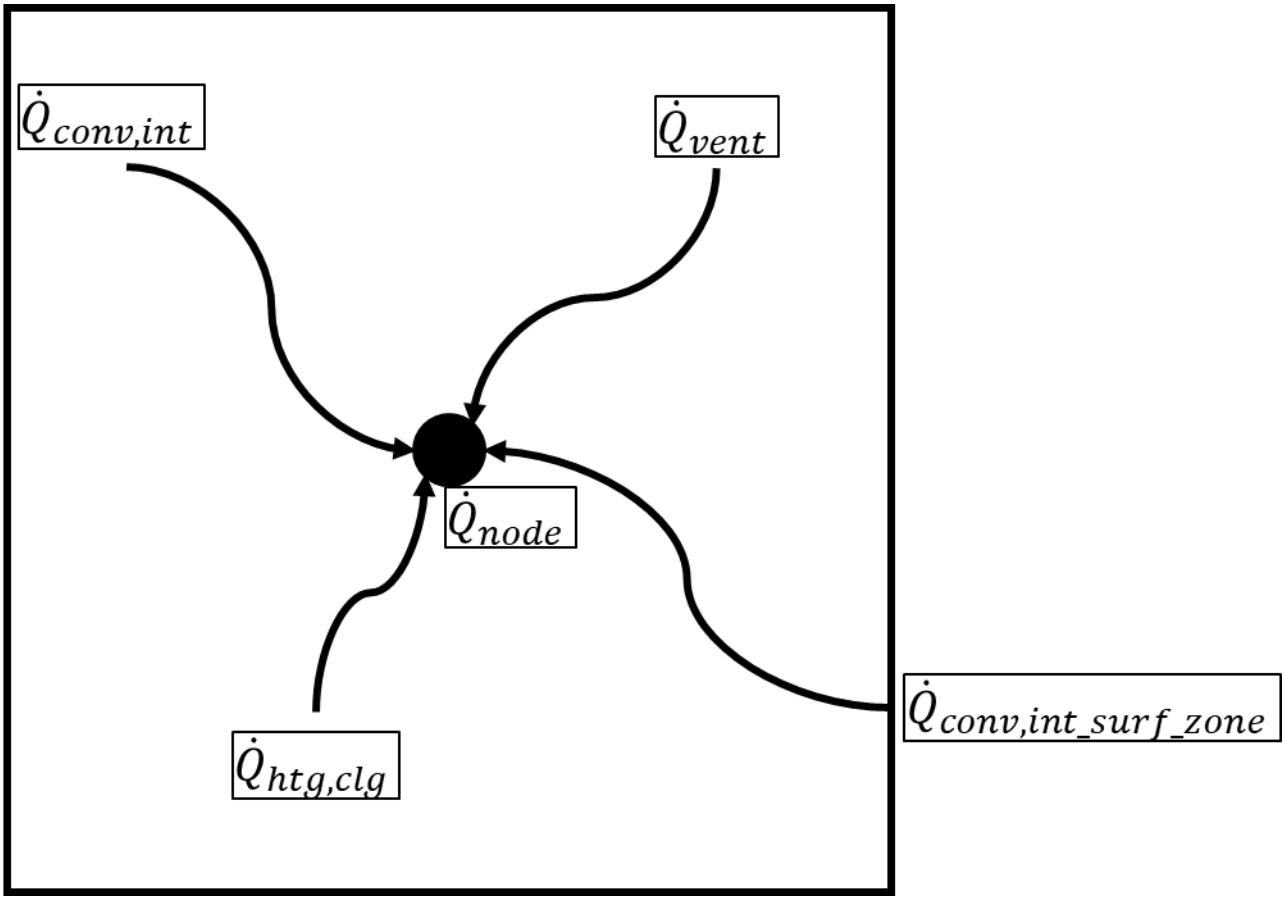
[177] Schnabel R, Wahl R, Klein R. Efficient RANSAC for Point-Cloud Shape Detection. Comput Graph Forum 2007;26:214–26. doi:10.1111/j.1467-8659.2007.01016.x.

[178] Mahony R, Euston M, Kim J, Coote P, Hamel T. A non-linear observer for attitude estimation of a fixed-wing unmanned aerial vehicle without GPS measurements. Trans Inst Meas Control 2011;33:699–717. doi:10.1177/0142331209343660.

Appendix A

Derivation of Heat Balance on Air Mass Node

BEMS has been used to discretise each thermal volume in a building model into a set of central nodes to represent the thermal conditions in each space and boundary for every time step of a simulation. The rate of energy transfer $\dot{Q}_{node}$ (W) to and from each central air node has been determined to comprise of four main components; 1) heat gains due to ventilation $\dot{Q}_{vent}$ (W), 2) heat gains due to space conditioning $\dot{Q}_{htg,clg}$ (W), 3) internal convective gains within the discrete thermal volume $\dot{Q}_{conv,int}$ (W) and 4) internal convective gains from the surrounding building fabric $\dot{Q}_{conv,int\_surf\_zone}$ (W) [124], see Appendix Figure A.1.



Appendix Figure A.1. Heat gain contributions to thermal air mass node

Thus the heat balance at the air mass node has been described by Appendix Equation A.1 as highlighted in Chapter 3. Each component of this equation has been used to describe a different heat transfer phenomenon which this Appendix has expanded for clarity.

$$\dot{Q}_{node} = \dot{Q}_{vent} + \dot{Q}_{htg,clg} + \dot{Q}_{conv,source} + \dot{Q}_{conv,int\_surf\_zone}$$

Appendix Equation A.1. Contributing gains to the heat balance of interior air mass node

Starting with the thermal air mass node itself, the rate of heat exchange at this point has been defined by Appendix Equation A.2 where ρ_{air} ($kg\ m^{-3}$) was the air density, c_p ($J\ kg^{-1}\ K^{-1}$) was the specific heat capacity of air, V (m^3) was the thermal zone volume, T (K) was the room dry-bulb air temperature and t (s) was time. This equation has described how the mass of air in a fixed thermal volume effects heat gains and losses at the node when temperature gradients between adjacent spaces over time are considered.

$$\dot{Q}_{node} = \rho_{air}\ c_p\ V\ \frac{dT}{dt}$$

Appendix Equation A.2. Heat balance at interior air mass node

The first component of this heat balance has been contributed too by ventilation into and out of the thermal volume under consideration $\dot{Q}_{vent}$ (W). These contributions have been determined to occur through passive means such as infiltration and active means such as HVAC systems. When the volume flow rate of air into the thermal volume $\dot{V}$ ($m\ s^{-1}$) has been defined in addition to the air dry-bulb temperatures inside $T_{air,in}$ (K) and outside $T_{air,out}$ (K) a volume, the contribution of ventilation to the thermal mass air node has been defined using Appendix Equation A.3.

$$\dot{Q}_{vent} = \rho_{air}\ c_p\ \dot{V}\ \left(T_{air,out} - T_{air,in}\right)$$

Appendix Equation A.3. Contributing ventilation gains to the heat balance of interior air mass node

The second contribution to the thermal balance of an air mass node has been identified as gains from space conditioning $\dot{Q}_{htg,clg}$ (W). This contribution has been determined by the HVAC set-point temperature inside a thermal zone which the HVAC system aims to achieve through heating or cooling the incoming air supply. This contribution has been defined by Appendix Equation A.4 where T_{HVAC} (K) was the defined HVAC set-point temperature.

$$\dot{Q}_{htg,clg} = \rho_{air}\ c_p\ V\ \left(\frac{T_{HVAC} - T_{air,in}}{dt}\right)$$

Appendix Equation A.4. Contributing heating & cooling gains to the heat balance of interior air mass node

The third heat balance contribution $\dot{Q}_{conv,source}\ (W)$ has been used to define the convective internal gains of a thermal volume from additional sources not already covered by the ventilation and heating and cooling contributions. Typically, this has included gains from occupants $\dot{Q}_{occu}\ (W)$, equipment $\dot{Q}_{equip}\ (W)$, lighting $\dot{Q}_{light}\ (W)$ and any number of additional sources within a single thermal zone $\dot{Q}_{n}\ (W)$. Appendix Equation A.5 has been used to describe this contribution which is usually defined by equipment specifications or BEMS industrial guidance such as CIBSE Guide A [122]. It has been noted that these heat gains into a thermal space also generate a radiative component however this only effects the thermal balance of nodes contained within the building fabric such as walls and not the air mass node within an air thermal volume.

$$\dot{Q}_{conv,sources} = \dot{Q}_{occu} + \dot{Q}_{equip} + \dot{Q}_{light} + \cdots + \dot{Q}_{n}$$

Appendix Equation A.5. Contributing convective source gains to the heat balance of interior air mass node

The final contribution to the thermal air mass node has been defined as the convective gains into a thermal volume from the surrounding surfaces (e.g. walls, floors and ceilings) $\dot{Q}_{conv,int\_surf\_zone}\ (W)$. This contribution has been calculated using Appendix Equation A.6 which has provided a summation of the convective heat transfers from all surfaces n surrounding a thermal volume where $\hat{h}_i$ was the convection heat transfer coefficient, $A_i\ (m^2)$ was the area and $T_{surf,i}\ (K)$ was the temperature of each surface i.

$$\dot{Q}_{conv,int\_surf\_zones} = \sum_{i=1}^{n} \hat{h}_i\, A_i \left(T_{surf,i} - T_{air,in}\right)$$

Appendix Equation A.6. Contributing convective surface gains to the heat balance of interior air mass node

The equations that have been included in this Appendix allow for calculation of the heat balance at a thermal mass node at each point within a time-series simulation. As discussed for the convective internal gains contribution the thermal air mass node does not receive heating or cooling from radiation phenomenon. Instead radiation gains created within a thermal volume have been imparted into thermal nodes that represent the surface surrounding a thermal volume.

Appendix B

F2050 Energy Bills

This appendix has presented redacted copies of the energy bills used in F2050. A number of elements from each utility bill have been redacted to protect identities and commercially sensitive information. The energy bills have been included for January 2016 (Appendix Figure B.1), February 2016 (Appendix Figure B.2), March 2016 (Appendix Figure B.3), April 2016 (Appendix Figure B.4), May 2016 (Appendix Figure B.5), June 2016 (Appendix Figure B.6), July 2016 (Appendix Figure B.7), August 2016 (Appendix Figure B.8), September 2016 (Appendix Figure B.9), October 2016 (Appendix Figure B.10), November 2016 (Appendix Figure B.11) and December 2016 (Appendix Figure B.12).

BREAKDOWN OF CHARGES FOR THE PERIOD: **01/01/2016 to 31/01/2016**

Supply Address: FACTORY 2050, EUROPA LINK, SHEFFIELD, S9 1XZ

Supply Number: S[redacted]

Meter Serial Number: [redacted]

Energy Charges			
Day 07:00 - 24:00	84274.9 kWh	at [redacted]p per kWh	[redacted]
Night 00:00 - 07:00	28674.7 kWh	at [redacted]p per kWh	[redacted]
Subtotal Energy Charges			[redacted]
Other Charges and Fees			
Availability	3341 kVA	at [redacted]£ per kVA/mth	[redacted]
Standing Charge			[redacted]
Data Collector & Data Aggregator			[redacted]
Subtotal Other Charges and Fees			[redacted]
Taxes, Levies and Other Statutory Obligations			
Climate Change Levy	for 100% of energy used	at [redacted]per kWh	[redacted]
Subtotal Taxes, Levies and Other Statutory Obligations			[redacted]
Amount (excluding VAT)			[redacted]
VAT payable	[redacted]	at 20%	[redacted]
Charges for the period			[redacted]

Appendix Figure B.1. F2050 energy bill for January 2016

BREAKDOWN OF CHARGES FOR THE PERIOD: **01/02/2016 to 29/02/2016**

Supply Address: FACTORY 2050, EUROPA LINK, SHEFFIELD, S9 1XZ

Supply Number: S [redacted]

Meter Serial Number: [redacted]

Energy Charges			
Day 07:00 - 24:00	84285.7 kWh	at [redacted] p per kWh	[redacted]
Night 00:00 - 07:00	29432 kWh	at [redacted] p per kWh	[redacted]
Subtotal Energy Charges			[redacted]
Other Charges and Fees			
Availability	3341 kVA	at [redacted] £ per kVA/mth	[redacted]
Standing Charge			[redacted]
Data Collector & Data Aggregator			[redacted]
Subtotal Other Charges and Fees			[redacted]
Taxes, Levies and Other Statutory Obligations			
Climate Change Levy	for 100% of energy used	at [redacted] p per kWh	[redacted]
Subtotal Taxes, Levies and Other Statutory Obligations			[redacted]
Amount (excluding VAT)			[redacted]
VAT payable	[redacted]	at 20%	[redacted]
Charges for the period			[redacted]

Appendix Figure B.2. F2050 energy bill for February 2016

BREAKDOWN OF CHARGES FOR THE PERIOD: **01/03/2016 to 31/03/2016**

Supply Address: FACTORY 2050, EUROPA LINK, SHEFFIELD, S9 1XZ

Supply Number: S [redacted]

Meter Serial Number: [redacted]

Energy Charges			
Day 07:00 - 24:00	79068.3 kWh	at [redacted]p per kWh	[redacted]
Night 00:00 - 07:00	28401.4 kWh	at [redacted]p per kWh	[redacted]
Subtotal Energy Charges			[redacted]
Other Charges and Fees			
Availability	3341 kVA	at [redacted]£ per kVA/mth	[redacted]
Standing Charge			[redacted]
Data Collector & Data Aggregator			[redacted]
Subtotal Other Charges and Fees			[redacted]
Taxes, Levies and Other Statutory Obligations			
Climate Change Levy	for 100% of energy used	at [redacted]p per kWh	[redacted]
Subtotal Taxes, Levies and Other Statutory Obligations			[redacted]
Amount (excluding VAT)			[redacted]
VAT payable	[redacted]	at 20%	[redacted]
Charges for the period			[redacted]

Appendix Figure B.3. F2050 energy bill for March 2016

BREAKDOWN OF CHARGES FOR THE PERIOD: **01/04/2016 to 30/04/2016**

Supply Address: FACTORY 2050, EUROPA LINK, SHEFFIELD, S9 1XZ

Supply Number: S [redacted]

Meter Serial Number: [redacted]

Energy Charges			
Day 07:00 - 24:00	76685 kWh	[redacted] p per kWh	[redacted]
Night 00:00 - 07:00	28956.2 kWh	[redacted] p per kWh	[redacted]
Subtotal Energy Charges			[redacted]
Other Charges and Fees			
Availability	3341 kVA	at [redacted] £ per kVA/mth	[redacted]
Standing Charge			[redacted]
Data Collector & Data Aggregator			[redacted]
Subtotal Other Charges and Fees			[redacted]
Taxes, Levies and Other Statutory Obligations			
Climate Change Levy	for 100% of energy used	at [redacted] p per kWh	[redacted]
Subtotal Taxes, Levies and Other Statutory Obligations			[redacted]
Amount (excluding VAT)			[redacted]
VAT payable	[redacted]	at 20%	[redacted]
Charges for the period			[redacted]

Appendix Figure B.4. F2050 energy bill for April 2016

BREAKDOWN OF CHARGES FOR THE PERIOD: **01/05/2016 to 31/05/2016**

Supply Address: FACTORY 2050, EUROPA LINK, SHEFFIELD, S9 1XZ

Supply Number: S [redacted]

Meter Serial Number: [redacted]

Energy Charges			
Day 07:00 - 24:00	69319.4 kWh	a[redacted]p per kWh	[redacted]
Night 00:00 - 07:00	26141.7 kWh	a[redacted]p per kWh	[redacted]
Subtotal Energy Charges			[redacted]
Other Charges and Fees			
Availability	3341 kVA	at[redacted]£ per kVA/mth	[redacted]
Standing Charge			[redacted]
Data Collector & Data Aggregator			[redacted]
Subtotal Other Charges and Fees			[redacted]
Taxes, Levies and Other Statutory Obligations			
Climate Change Levy	for 100% of energy used	at[redacted]per kWh	[redacted]
Subtotal Taxes, Levies and Other Statutory Obligations			[redacted]
Amount (excluding VAT)			[redacted]
VAT payable	[redacted]	at 20%	[redacted]
Charges for the period			[redacted]

Appendix Figure B.5. F2050 energy bill for May 2016

BREAKDOWN OF CHARGES FOR THE PERIOD: **01/06/2016 to 30/06/2016**

Supply Address: FACTORY 2050, EUROPA LINK, SHEFFIELD, S9 1XZ

Supply Number: S [redacted]

Meter Serial Number: [redacted]

Energy Charges			
Day 07:00 - 24:00	68603.7 kWh	[redacted] p per kWh	[redacted]
Night 00:00 - 07:00	25309.6 kWh	[redacted] p per kWh	[redacted]
Subtotal Energy Charges			[redacted]
Other Charges and Fees			
Availability	3341 kVA	at [redacted] € per kVA/mth	[redacted]
Standing Charge			[redacted]
Data Collector & Data Aggregator			[redacted]
Subtotal Other Charges and Fees			[redacted]
Taxes, Levies and Other Statutory Obligations			
Climate Change Levy	for 100% of energy used	at [redacted] p per kWh	[redacted]
Subtotal Taxes, Levies and Other Statutory Obligations			[redacted]
Amount (excluding VAT)			[redacted]
VAT payable	[redacted]	at 20%	[redacted]
Charges for the period			[redacted]

Appendix Figure B.6. F2050 energy bill for June 2016

BREAKDOWN OF CHARGES FOR THE PERIOD: **01/07/2016 to 31/07/2016**

Supply Address: FACTORY 2050, EUROPA LINK, SHEFFIELD, S9 1XZ

Supply Number: S [redacted]

Meter Serial Number: [redacted]

Energy Charges			
Day 07:00 - 24:00	65772.9 kWh	[redacted]p per kWh	[redacted]
Night 00:00 - 07:00	23767.9 kWh	[redacted] per kWh	[redacted]
Subtotal Energy Charges			[redacted]
Other Charges and Fees			
Availability	3341 kVA	at [redacted]£ per kVA/mth	[redacted]
Standing Charge			[redacted]
Data Collector & Data Aggregator			[redacted]
Subtotal Other Charges and Fees			[redacted]
Taxes, Levies and Other Statutory Obligations			
Climate Change Levy	for 100% of energy used	at [redacted] per kWh	[redacted]
Subtotal Taxes, Levies and Other Statutory Obligations			[redacted]
Amount (excluding VAT)			[redacted]
VAT payable	[redacted]	at 20%	[redacted]
Charges for the period			[redacted]

Appendix Figure B.7. F2050 energy bill for July 2016

BREAKDOWN OF CHARGES FOR THE PERIOD: **01/08/2016 to 31/08/2016**

Supply Address: FACTORY 2050, EUROPA LINK, SHEFFIELD, S9 1XZ

Supply Number: S [redacted]

Meter Serial Number: [redacted]

Energy Charges			
Day 07:00 - 24:00	70918.4 kWh	[redacted] per kWh	[redacted]
Night 00:00 - 07:00	24664 kWh	[redacted] per kWh	[redacted]
Subtotal Energy Charges			[redacted]
Other Charges and Fees			
Availability	3341 kVA	at [redacted] £ per kVA/mth	[redacted]
Standing Charge			[redacted]
Data Collector & Data Aggregator			[redacted]
Subtotal Other Charges and Fees			[redacted]
Taxes, Levies and Other Statutory Obligations			
Climate Change Levy	for 100% of energy used	[redacted] per kWh	[redacted]
Subtotal Taxes, Levies and Other Statutory Obligations			[redacted]
Amount (excluding VAT)			[redacted]
VAT payable	[redacted]	at 20%	[redacted]
Charges for the period			[redacted]

Appendix Figure B.8. F2050 energy bill for August 2016

BREAKDOWN OF CHARGES FOR THE PERIOD: **01/09/2016 to 30/09/2016**

Supply Address: FACTORY 2050, EUROPA LINK, SHEFFIELD, S9 1XZ

Supply Number: S [redacted]

Meter Serial Number: [redacted]

Energy Charges			
Day 07:00 - 24:00	68133.7 kWh	a[redacted] per kWh	[redacted]
Night 00:00 - 07:00	24072.9 kWh	at[redacted] per kWh	[redacted]
Subtotal Energy Charges			[redacted]
Other Charges and Fees			
Availability	3341 kVA	at [redacted] £ per kVA/mth	[redacted]
Standing Charge			[redacted]
Data Collector & Data Aggregator			[redacted]
Subtotal Other Charges and Fees			[redacted]
Reconciliation Adjustments			
FIT Levelisation Reconciliation Charge			[redacted]
Subtotal Reconciliation Adjustments			[redacted]
Taxes, Levies and Other Statutory Obligations			
Climate Change Levy	for 100% of energy used	at [redacted] p per kWh	[redacted]
Subtotal Taxes, Levies and Other Statutory Obligations			[redacted]
Amount (excluding VAT)			[redacted]
VAT payable	[redacted]	at 20%	[redacted]
Charges for the period			[redacted]

Appendix Figure B.9. F2050 energy bill for September 2016

BREAKDOWN OF CHARGES FOR THE PERIOD: **01/10/2016 to 31/10/2016**

Supply Address: FACTORY 2050, EUROPA LINK, SHEFFIELD, S9 1XZ

Supply Number: S [redacted]

Meter Serial Number: [redacted]

Energy Charges			
Day 07:00 - 24:00	86504.3 kWh	[redacted] per kWh	[redacted]
Night 00:00 - 07:00	34075.1 kWh	[redacted] per kWh	[redacted]
Subtotal Energy Charges			[redacted]
Other Charges and Fees			
Availability	3341 kVA	[redacted] per kVA/mth	[redacted]
Standing Charge			[redacted]
Data Collector & Data Aggregator			[redacted]
Subtotal Other Charges and Fees			[redacted]
Taxes, Levies and Other Statutory Obligations			
Climate Change Levy	for 100% of energy used	[redacted] p per kWh	[redacted]
Subtotal Taxes, Levies and Other Statutory Obligations			[redacted]
Amount (excluding VAT)			[redacted]
VAT payable	[redacted]	at 20%	[redacted]
Charges for the period			[redacted]

Appendix Figure B.10. F2050 energy bill for October 2016

BREAKDOWN OF CHARGES FOR THE PERIOD: **01/11/2016 to 30/11/2016**

Supply Address: FACTORY 2050, EUROPA LINK, SHEFFIELD, S9 1XZ

Supply Number: S [redacted]

Meter Serial Number: [redacted]

Energy Charges			
Day 07:00 - 24:00	81385.2 kWh	a[redacted]per kWh	[redacted]
Night 00:00 - 07:00	29566.8 kWh	a[redacted]per kWh	[redacted]
Subtotal Energy Charges			[redacted]
Other Charges and Fees			
Availability	3341 kVA	at [redacted] per kVA/mth	[redacted]
Standing Charge			[redacted]
Data Collector & Data Aggregator			[redacted]
Subtotal Other Charges and Fees			[redacted]
Taxes, Levies and Other Statutory Obligations			
Climate Change Levy	for 100% of energy used	[redacted]p per kWh	[redacted]
Subtotal Taxes, Levies and Other Statutory Obligations			[redacted]
Amount (excluding VAT)			[redacted]
VAT payable	[redacted]	at 20%	[redacted]
Charges for the period			[redacted]

Appendix Figure B.11. F2050 energy bill for November 2016

BREAKDOWN OF CHARGES FOR THE PERIOD: **01/12/2016 to 31/12/2016**

Supply Address: FACTORY 2050, EUROPA LINK, SHEFFIELD, S9 1XZ

Supply Number: S [redacted]

Meter Serial Number: [redacted]

Energy Charges			
Day 07:00 - 24:00	68853.3 kWh	[redacted] per kWh	[redacted]
Night 00:00 - 07:00	26125.6 kWh	[redacted] per kWh	[redacted]
Subtotal Energy Charges			[redacted]
Other Charges and Fees			
Availability	3341 kVA	[redacted] per kVA/mth	[redacted]
Standing Charge			[redacted]
Data Collector & Data Aggregator			[redacted]
Reactive Power	31 kVArh	[redacted] per kVArh	[redacted]
Subtotal Other Charges and Fees			[redacted]
Taxes, Levies and Other Statutory Obligations			
Climate Change Levy	for 100% of energy used	[redacted] per kWh	[redacted]
Subtotal Taxes, Levies and Other Statutory Obligations			[redacted]
Amount (excluding VAT)			[redacted]
VAT payable	[redacted]	at 20%	[redacted]
Charges for the period			[redacted]

Appendix Figure B.12. F2050 energy bill for December 2016

Appendix C

Air Infiltration Development Algorithm

This appendix has included a copy of the C++ code used to reproduce the Air Infiltration Development Algorithm as proposed by Liddament [145]. It has been developed further to accept a large amount of input data via a text file instead of requiring manual entries. The software contained seven separate files written in C++ and was developed using Microsoft Visual Studio 2015 as presented below;

1. AIDAv2.h

```
#pragma once

#include "Common.h"
#include "PressureCalcs.h"
#include "InfiltrationCalcs.h"
```

2. Common.h

```
#pragma once

#include <iostream>
#include <string>
#include <vector>
#include <fstream>
#include <iomanip>
```

3. InfiltrationCalcs.h

```
#pragma once
#include "Common.h"

void InfiltrationCalculation(int *L, std::vector<float> *T, std::vector<float> *F,
std::vector<float> *C, std::vector<float> *N, float *Q, float *A, float *V);
```

4. PressureCalcs.h

```
#pragma once
#include "Common.h"

float windPresCalc(float density, float pressure, float windspeed);
float stackPresCalc(float height, float extTemp, float intTemp);
float totalPresCalc(float windPres, float stackPres);
```

5. AIDAv2.cpp

```
// AIDAv2.cpp : Defines the entry point for the console application.
//The following Air Infiltration Development Algorithm (AIDA) has been written under the
guidance of Appendix 4.A3 of CIBSE Guide A.

//The names of variables are as follows :
```

```
//            A             Air change rate(h-1)
//            B             Flow balance
//            C(J)   Flow coefficient for path J
//            D             Air density(kg·m-3)
//            E             External temperature(°C)
//            F(J)   Calculated flow rate for flow path J(m3·s-1)
//            H(J)   Height of flow path J(m)
//            I             Internal temperature(°C)
//            J             Flow path number
//            L             Total number of flow paths
//            N(J)   Flow exponent for flow path J
//            O             Pressure difference across flow path(Pa)
//            P(J)   Wind pressure coefficient for flow path J
//            Q             Infiltration rate(m3·s-1)
//            R             Internal pressure(Pa)
//            S(J)   Stack - induced pressure for flow path J(Pa)
//            T(J)   Total external pressure on flow path J(Pa)
//            U             Wind speed at building height(m·s-1)
//            V             Volume of building enclosure(m3)
//            W(J)   Wind - induced pressure(Pa)
//            X             Iteration pressure step(Pa)
//            Y             Iteration counter

#include "AIDAv2.h"


int main()
{
	/// Welcome/Input string messages
	std::string _welcome, _instruct1, _volume, _flowPaths, _height, _flowCoef, _flowExp,
_presCoef, _path, _checkInputs, _validDataMsg, _validData, _climate, _extTemp, _intTemp,
_windSpeed;
	_welcome = "Welcome to  Air Infiltration Development Algorithm (AIDA).\nThe programme
has been written in C++ under the guidance of Appendix 4.A3 of CIBSE Guide A by Tom
Garwood";
	_instruct1 = "Reading Building Data";
	_volume = "Building volume (m3) = ";
	_flowPaths = "Number of flow paths = ";
	_height = "Height (m) ";
	_flowCoef = "Flow coeffiecient ";
	_flowExp = "Flow Exponent ";
	_presCoef = "Pressure Coefficient ";
	_path = "for path: ";
	_checkInputs = "Please double check inputs before continuing:";
	_validDataMsg = "Is Data input correct? (Y/N): ";
	_climate = "Enter climatic data:";
	_extTemp = "External Temperature (degC) = ";
	_intTemp = "Internal Temperature (degC) = ";
	_windSpeed = "Wind speed at building height(m*s^-1)";

	//Calculation String Messages
	std::string _presCalc, _windPresCalc, _stackPresCalc, _totFlowPathPres, _infilCalc;
	_presCalc = "Executing pressure calculation on each flow path...";
	_windPresCalc = "Executing wind pressure calculation on flow path ";
	_stackPresCalc = "Executiong stack pressure calculation on flow path ";
	_totFlowPathPres = "Executing total pressure calculation on flow path ";
	_infilCalc = "Executing infiltration calculation on building...";

	//Result String Messages
	std::string _infilRate, _ach, _NewClimateCalcQ, _NewClimateCalc, _flowRate;
	_infilRate = "Infiltration rate (m3.s^-1) = ";
```

```
	_ach = "Air changes per hour (ach) (h^-1) = ";
	_flowRate = "Flow Rate (m3.s^-1)";
	_NewClimateCalcQ = "Do you wish to repeat the calculation for different climatic
conditions? (Y/N)";


	std::cout << _welcome << std::endl;

//Initialise variable vectors

	std::vector<float> H;			//	Height of flow path J(m)

	std::vector<float> C_med;	//	Median Flow coefficient for path J
	std::vector<float> C_UQ;	//	Upper Quartile Flow coefficient for path J
	std::vector<float> C_LQ;	//	Lower Quartile Flow coefficient for path J

	std::vector<float> N_med;	//	Median Flow exponent for flow path J
	std::vector<float> N_UQ;	//	Upper Quartile Flow exponent for flow path J
	std::vector<float> N_LQ;	//	Lower Quartile Flow exponent for flow path J

	std::vector<float> P;			//	Wind pressure coefficient for flow path J
	std::vector<float> T;			//	Total external pressure on flow path J(Pa)
	std::vector<float> W;			//	Wind - induced pressure(Pa)
	std::vector<float> S;			//	Stack - induced pressure for flow path J(Pa)

	std::vector<float> F_med;	//	Median Calculated flow rate for flow path J(m3*s^-
1)
	std::vector<float> F_UQ;	//	Upper Quartile Calculated flow rate for flow path
J(m3*s^-1)
	std::vector<float> F_LQ;	//	Lower Quartile Calculated flow rate for flow path
J(m3*s^-1)

	float D = 1.29;				// Air density(kg*m^-3) at 0°C
	float V;					// Volume of building enclosure(m3)
	int L;						// Total number of flow paths
	float E;					// External temperature(°C)
	float I;					// Internal temperature(°C)
	float U;					// Wind speed at building height(m*s^-1)
	float Q_med = 0;			// Median Infiltration rate (m3·s-1)
	float Q_UQ = 0;				// Upper Quartile Infiltration rate (m3·s-1)
	float Q_LQ = 0;				// Lower Quartile Infiltration rate (m3·s-1)
	float A_med = 0;			// Median Air change rate (h-1)
	float A_UQ = 0;				// Upper Quartile Air change rate (h-1)
	float A_LQ = 0;				// Lower Quartile Air change rate (h-1)

// Open Text File

	std::fstream inputFile;
	std::string inputFileValue;
	inputFile.open("C:/Users/Tom Garwood/Desktop/AIDA Input.txt");
	if (!inputFile)  //check file has been opened
	{
		std::cerr << "Unable to open input file";
		std::exit(1); // call system to stop
	}
	else
	{
		std::cout << "Input data file found.  Importing Data.";
	}
	// Find out how many flowpaths innput file
	int flowPaths = -1;
```

```
std::vector<std::vector<float>> inputData(8);
int index = 0;
bool height = true;
bool C_lq = false;
bool N_lq = false;
bool C_m = false;
bool N_m = false;
bool C_uq = false;
bool N_uq = false;

while (inputFile >> inputFileValue)
{
	if (flowPaths == -1)
	{
		V = std::stof(inputFileValue);
		flowPaths++;
	}
	else
	{
		while (index == 7)
		{
			inputData[7].push_back(std::stof(inputFileValue));
			index = 8;
		}
		while (index == 6)
		{
			inputData[6].push_back(std::stof(inputFileValue));
			index = 7;
		}
		while (index == 5)
		{
			inputData[5].push_back(std::stof(inputFileValue));
			index = 6;
		}
		while (index == 4)
		{
			inputData[4].push_back(std::stof(inputFileValue));
			index = 5;
		}
		while (index == 3)
		{
			inputData[3].push_back(std::stof(inputFileValue));
			index = 4;
		}
		while (index == 2)
		{
			inputData[2].push_back(std::stof(inputFileValue));
			index = 3;
		}
		while (index == 1)
		{
			inputData[1].push_back(std::stof(inputFileValue));
			index = 2;
		}
		while (index == 0)
		{
			inputData[0].push_back(std::stof(inputFileValue));
			flowPaths++;
			index = 1;
		}
		while (index == 8)
```

```
                {
                        index = 0;
                }
        }


    }

    L = flowPaths;

    // Data inputs
    std::cout << _instruct1 << std::endl << _volume << std::setprecision(1) <<
std::to_string(V);
    std::cout << std::endl;

    std::cout <<  std::endl << _flowPaths + std::to_string(L);
    std::cout << std::endl;
    //std::setprecision(6);

    //Height, Flow Coefficient, Flow Exponent & Pressure Coefficient Inputs
    int j = 0;
    for (int i = 0; i < L; i++)
    {
        //Height
        H.push_back(inputData[j][i]);
        j++;

        //LQ Flow Coeffieicnt
        C_LQ.push_back(inputData[j][i]);
        j++;

        //LQ Flow Exponent
        N_LQ.push_back(inputData[j][i]);
        j++;

        //Median  Flow Coeffieicnt
        C_med.push_back(inputData[j][i]);
        j++;

        //Median Flow Exponent
        N_med.push_back(inputData[j][i]);
        j++;

        //UQ Flow Coeffieicnt
        C_UQ.push_back(inputData[j][i]);
        j++;

        //UQ Flow Exponent
        N_UQ.push_back(inputData[j][i]);
        j++;

        // Pressure Coefficient
        P.push_back(inputData[j][i]);
        j++;
        if (j == 8) j = 0;

    }

    bool finalClimateData = false;

    while (!finalClimateData)
```

```
	{

		std::cout << std::endl << _climate << std::endl;
		std::cout << _extTemp;
		std::cin >> E;
		std::cout << _intTemp;
		std::cin >> I;
		std::cout << _windSpeed;
		std::cin >> U;

		std::cout << std::endl << _checkInputs << std::endl;

		std::cout << _extTemp << std::to_string(E) << std::endl;
		std::cout << _intTemp << std::to_string(I) << std::endl;
		std::cout << _windSpeed << std::to_string(U) << std::endl << std::endl;


		std::cout << _validDataMsg;
		std::cin >> _validData;
		bool validData = false;
		if (_validData == "N" || _validData == "n") validData = false;
		if (_validData == "Y" || _validData == "y") validData = true;

		while (!validData)
		{
			std::string _flowPathError, _climateError;
			bool flowPathDataError, climateDataError;
			std::cout << "Is there an error with Flow Path Entries? (Y/N)" <<
std::endl;
			std::cin >> _flowPathError;
			if (_flowPathError == "N" || _flowPathError == "n") flowPathDataError =
false;
			if (_flowPathError == "Y" || _flowPathError == "y") flowPathDataError =
true;
			std::cout << "Is there an error with Climatic Conditions? (Y/N)" <<
std::endl;
			std::cin >> _climateError;
			if (_climateError == "N" || _climateError == "n") climateDataError =
false;
			if (_climateError == "Y" || _climateError == "y") climateDataError =
true;


			while (flowPathDataError)
			{
				int flowPathNo;
				std::string _flowPathNo;
				std::cout << "Which flow path has incorrect data?" << std::endl;
				std::cin >> _flowPathNo;
				flowPathNo = stoi(_flowPathNo);
				std::cout << "Re-enter correct flow path data for flow path " +
_flowPathNo << std::endl;

				std::string _input;
				//Height
				std::cout << _height << _path << _flowPathNo + " = ";
				std::cin >> _input;
				H[flowPathNo - 1] = std::stof(_input);
				std::cout << std::endl;

				//UQ Flow Coeffieicnt
```

```
                        std::cout << "Upper Quartile " + _flowCoef << _path <<
_flowPathNo + " = ";
                        std::cin >> _input;
                        C_UQ[flowPathNo - 1] = std::stof(_input);
                        //Median Flow Coeffieicnt
                        std::cout << "Median " + _flowCoef << _path << _flowPathNo + " =
";
                        std::cin >> _input;
                        C_med[flowPathNo - 1] = std::stof(_input);
                        //LQ Flow Coeffieicnt
                        std::cout << "Lower Quartile " + _flowCoef << _path <<
_flowPathNo + " = ";
                        std::cin >> _input;
                        C_LQ[flowPathNo - 1] = std::stof(_input);
                        std::cout << std::endl;

                        //UQ Flow Exponent
                        std::cout << "Upper Quartile " + _flowExp << _path <<
_flowPathNo + " = ";
                        std::cin >> _input;
                        N_UQ[flowPathNo - 1] = std::stof(_input);
                        //Median Flow Exponent
                        std::cout << "Median " + _flowExp << _path << _flowPathNo + " =
";
                        std::cin >> _input;
                        N_med[flowPathNo - 1] = std::stof(_input);
                        //LQ Flow Exponent
                        std::cout << "Lower Quartile " + _flowExp << _path <<
_flowPathNo + " = ";
                        std::cin >> _input;
                        N_LQ[flowPathNo - 1] = std::stof(_input);
                        std::cout << std::endl;

                        // Pressure Coefficient
                        std::cout << _presCoef << _path << _flowPathNo + " = ";
                        std::cin >> _input;
                        P[flowPathNo - 1] = std::stof(_input);
                        std::cout << std::endl;

                        std::cout << std::endl << _checkInputs << std::endl;

                        std::cout << "Is there still an error with Flow Path Entries?
                    (Y/N)" << std::endl;
                        std::cin >> _flowPathError;
                        if (_flowPathError == "N" || _flowPathError == "n")
flowPathDataError = false;
                        if (_flowPathError == "Y" || _flowPathError == "y")
flowPathDataError = true;
                    }
                    while (climateDataError)
                    {
                        std::cout << std::endl << _climate << std::endl;
                        std::cout << _extTemp;
                        std::cin >> E;
                        std::cout << _intTemp;
                        std::cin >> I;
                        std::cout << _windSpeed;
                        std::cin >> U;

                        std::cout << _extTemp << std::to_string(E) << std::endl;
                        std::cout << _intTemp << std::to_string(I) << std::endl;
```

```
                        std::cout << _windSpeed << std::to_string(U) << std::endl <<
std::endl;

                        std::cout << "Is there still an error with Climatic Conditions?
(Y/N)" << std::endl;
                        std::cin >> _climateError;
                        if (_climateError == "N" || _climateError == "n")
climateDataError = false;
                        if (_climateError == "Y" || _climateError == "y")
climateDataError = true;
                    }
                    validData = true;
            }

// Calculations
            //Pressure Calcs
            std::cout << std::endl << _presCalc << std::endl << std::endl;
            for (int i = 0; i < L; i++)
            {
                  //Wind Pressure Calculation
                  std::cout << _windPresCalc + std::to_string(i + 1) << std::endl;
                  W.push_back(windPresCalc(D, P[i], U));

                  //Stack Pressure Calculation
                  std::cout << _stackPresCalc + std::to_string(i + 1) << std::endl;
                  S.push_back(stackPresCalc(H[i], E, I));

                  //Total Flow Path Pressure Calculation
                  std::cout << _totFlowPathPres + std::to_string(i + 1) << std::endl;
                  T.push_back(totalPresCalc(W[i], S[i]));
            }

            //Infiltration Calcs

            std::cout << std::endl << _infilCalc << std::endl << std::endl;

            //Median
            InfiltrationCalculation(&L, &T, &F_med, &C_med, &N_med, &Q_med, &A_med, &V);
            //UQ
            InfiltrationCalculation(&L, &T, &F_UQ, &C_UQ, &N_UQ, &Q_UQ, &A_UQ, &V);
            //LQ
            InfiltrationCalculation(&L, &T, &F_LQ, &C_LQ, &N_LQ, &Q_LQ, &A_LQ, &V);

            // Results

            std::cout << std::endl << "Upper Quartile " + _infilRate <<
std::to_string(Q_UQ) << std::endl;
            std::cout << "Median " + _infilRate << std::to_string(Q_med) << std::endl;
            std::cout << "Lower Quartile " + _infilRate << std::to_string(Q_LQ) <<
std::endl << std::endl;

            std::cout << std::endl << "Upper Quartile " + _ach << std::to_string(A_UQ) <<
std::endl;
            std::cout << "Median " + _ach << std::to_string(A_med) << std::endl;
            std::cout << "Lower Quartile " + _ach << std::to_string(A_LQ) << std::endl <<
std::endl;

            std::cout << _NewClimateCalcQ;
            std::cin >> _NewClimateCalc;

            if (_NewClimateCalc == "N" || _NewClimateCalc == "n")
```

```
		{
			finalClimateData = true;
		}

		W.clear();
		S.clear();
		T.clear();
		F_med.clear();
		F_UQ.clear();
		F_LQ.clear();
		Q_med = 0;
		Q_UQ = 0;
		Q_LQ = 0;
		A_med = 0;
		A_UQ = 0;
		A_LQ = 0;

	}

	std::cout << "CALCULATIONS COMPLETE!";


    return 0;
}
```

6. InfiltrationCalcs.cpp

```
#include "InfiltrationCalcs.h"


void InfiltrationCalculation(int *L, std::vector<float> *T, std::vector<float> *F,
std::vector<float> *C, std::vector<float> *N, float *Q, float *A, float *V)
{
	float R = -100;				// Internal Pressure (Pa)
	float X = 50;			// Iteration Pressure Step (Pa)
	int Y = 0;				// Iteration Counter

	bool iterationFinished = false;

	while (!iterationFinished)
	{
		float B = 0;			// Flow balance
		R = R + X;
		F->clear();

		for (int i = 0; i < *L; i++)
		{
			Y = Y + 1;
			float O = T->at(i) - R;  // Pressure difference across flow path
			if (O == 0)
			{
				F->push_back(0);
			}
			else
			{
				F->push_back(C->at(i) * pow(abs(O), N->at(i)) * (O / abs(O)));
				B = B + F->at(i);
			}
		}
		iterationFinished = true;
```

```
		if (B < 0)
		{
			std::cout << "Flow balance is negative, adjusting seed points and
restarting iteration." << std::endl;
			R = R - X;
			X = X / 2;
			iterationFinished = false;
		}
		if (B >= 0.0001)
		{
			std::cout << "Flow balance is greater than tolerance, restarting
iteration." << std::endl;
			iterationFinished = false;
		}
	}

	for (int i = 0; i < *L; i++)
	{
		if (F->at(i) > 0)
		{
			*Q = *Q + F->at(i);
		}
	}

	*A = *Q * 3600 / *V;

}
```

7. PressureCalcs.cpp

```
#include "PressureCalcs.h"

float windPresCalc(float density, float pressure, float windspeed)
{
	float windPressure;

	windPressure = 0.5 * density * pressure * windspeed * windspeed;

	return windPressure;
}


float stackPresCalc(float height, float extTemp, float intTemp)
{
	float stackPressure;

	stackPressure = -3455 * height * ((1 / (extTemp + 273)) - (1 / (intTemp + 273)));

	return stackPressure;
}

float totalPresCalc(float windPres, float stackPres)
{
	float totalPresCalc;

	totalPresCalc = windPres + stackPres;

	return totalPresCalc;
}
```

Appendix D

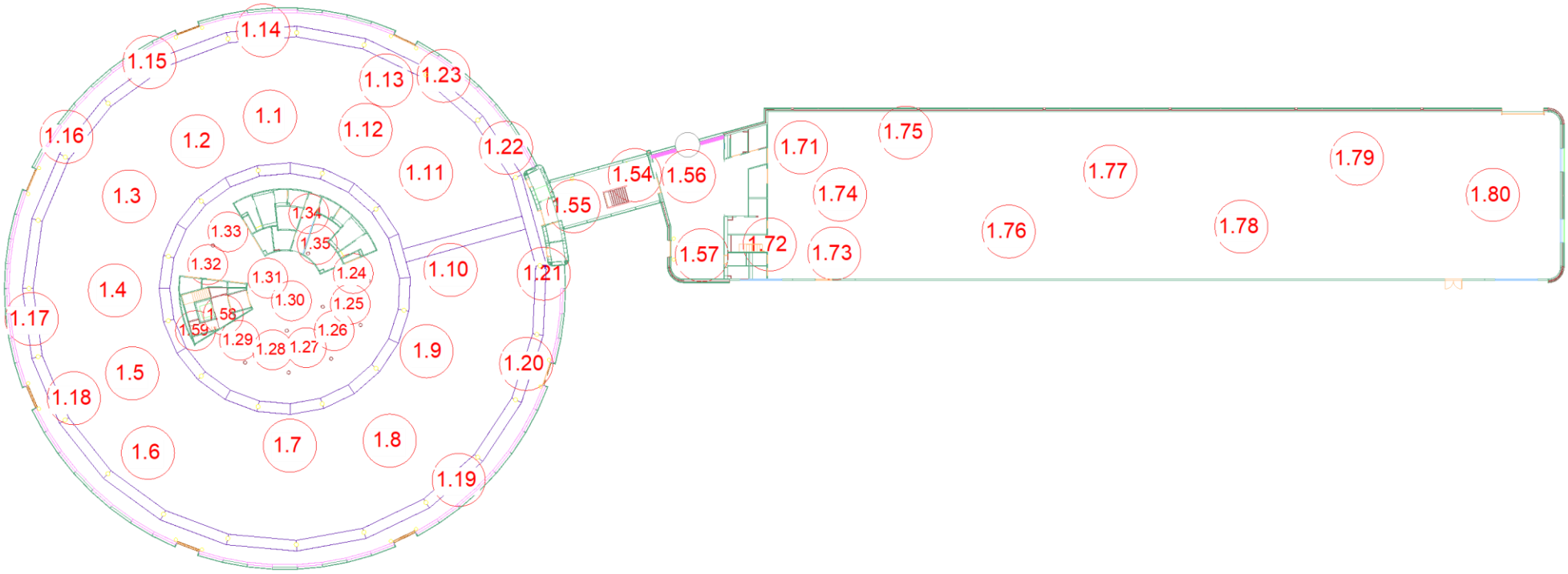
F2050 Scan Survey Maps

This Appendix has presented the scanning maps produced following the laser scan survey of F2050. For clarity each map has been presented in a landscape manner.

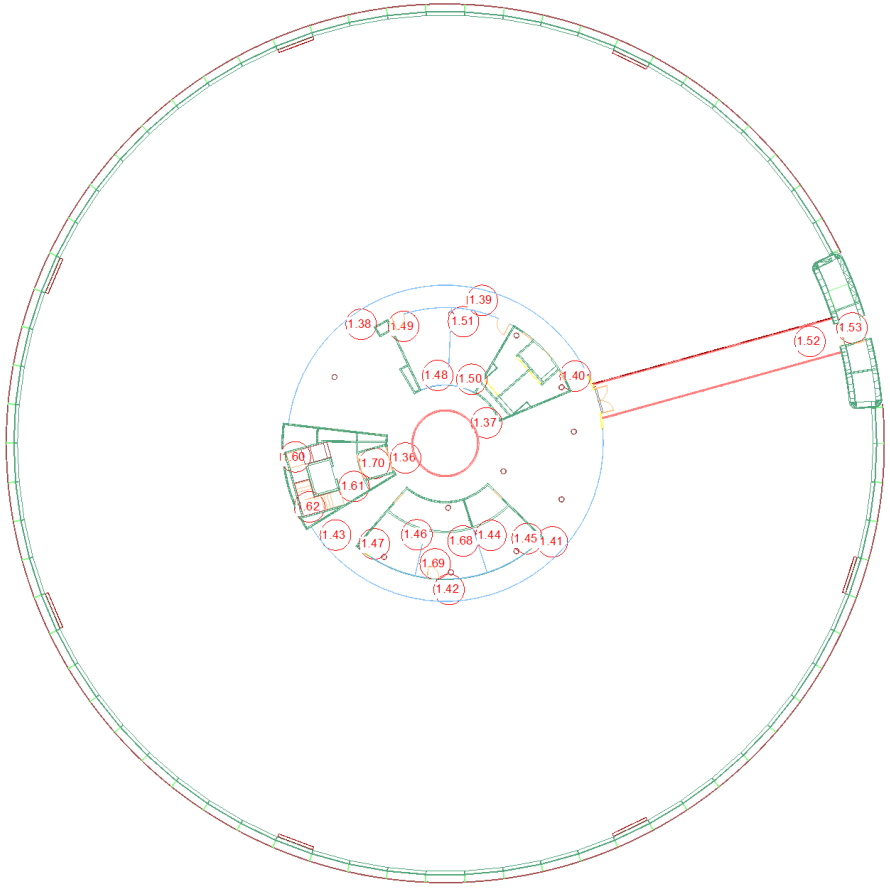
The first survey of F2050 acquired survey data from the Ground Floor with blinds open (Appendix Figure D.1), the 1st Floor of the Circular Workshop (Appendix Figure D.2), the Mezzanine Floor (Plant Room) of the Circular Workshop (Appendix Figure D.3) and the 2nd Floor of the Circular Workshop (Appendix Figure D.4).

When reconstruction results were achieved on these scans that highlighted the challenge of processing large data set noise, a second survey was conducted of a portion of the F2050 ground floor with the blinds closed (Appendix Figure D.5).

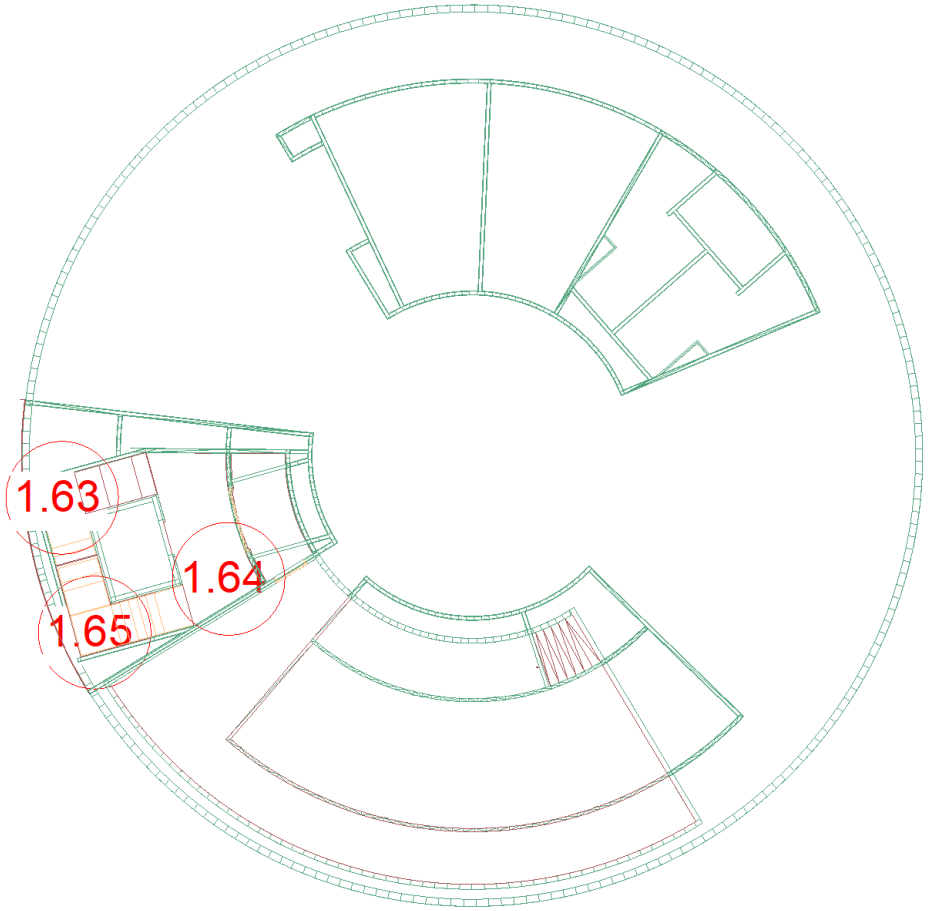
Following the second survey a set of scans from the original ground floor survey were combined with a selection of scans taken with blinds down. These were found to produce promising reconstruction results that were used further in BEMS research (Appendix Figure D.6).



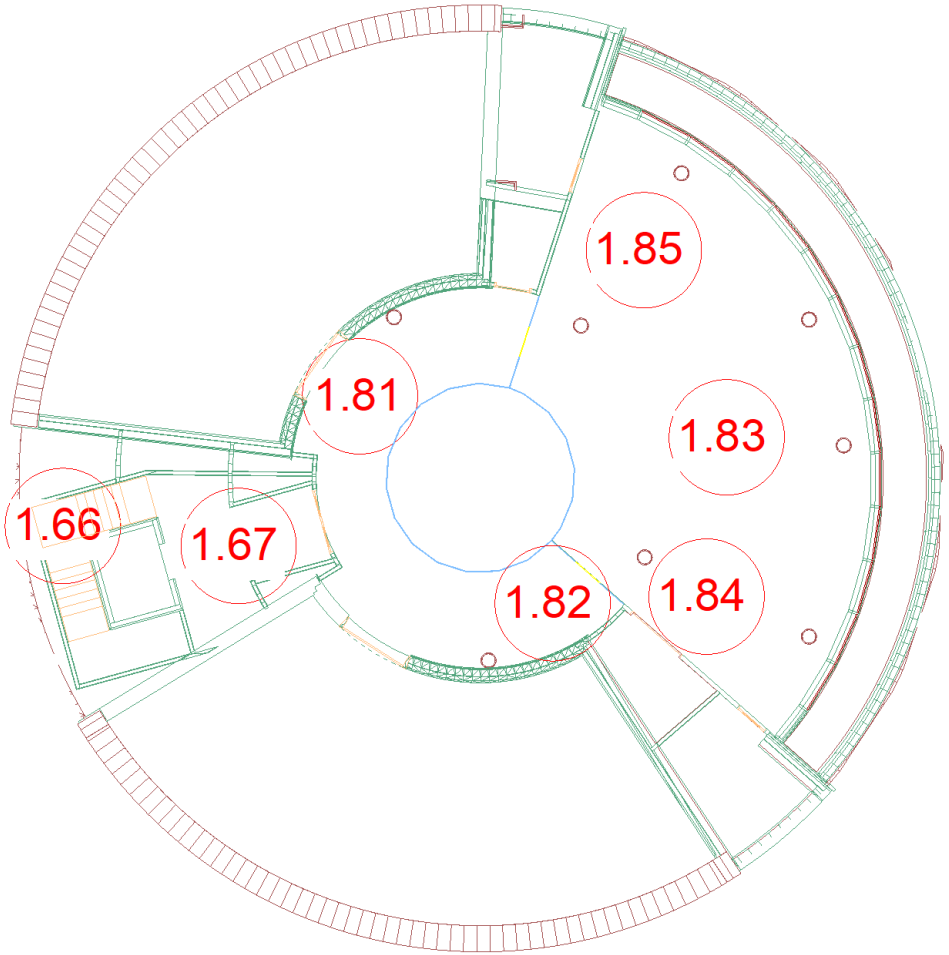
Appendix Figure D.1. F2050 ground floor map of scanning locations (blinds open)



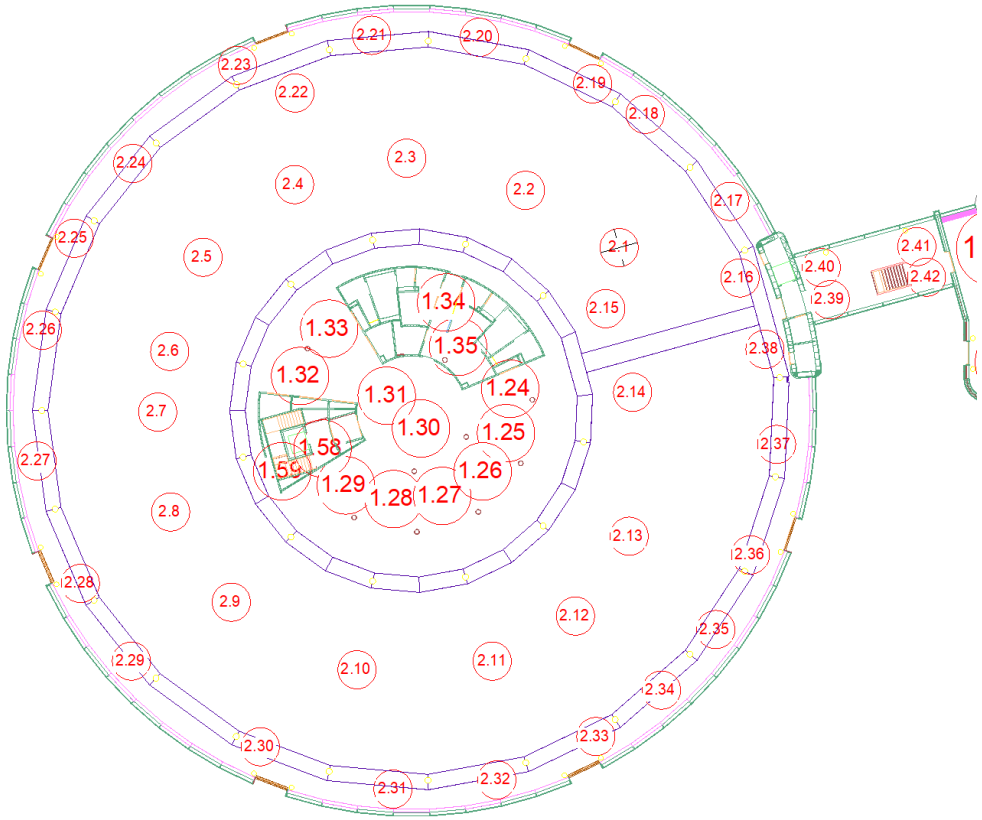
Appendix Figure D.2. F2050 1st floor of circular workshop map of scanning locations



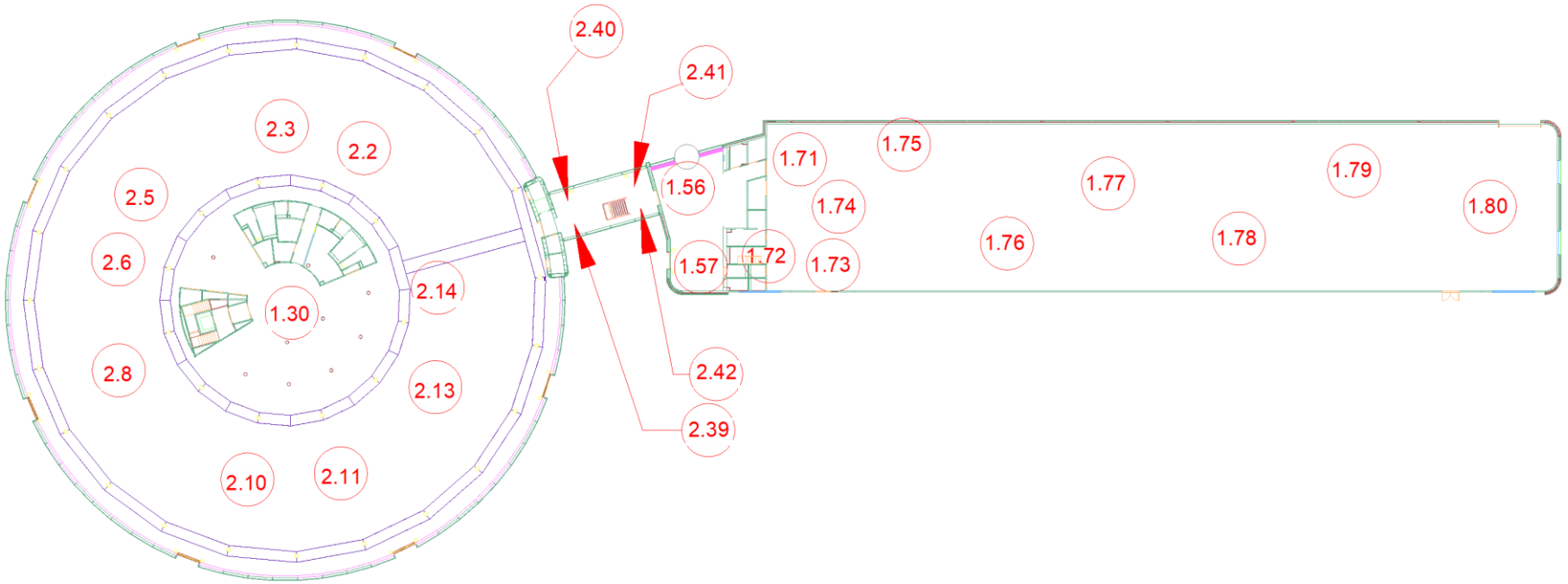
Appendix Figure D.3. F2050 mezzanine floor of circular workshop map of scanning locations



Appendix Figure D.4. F2050 2nd floor of circular workshop map of scanning locations



Appendix Figure D.5. F2050 ground floor map of scanning locations (blinds closed)



Appendix Figure D.6. Selection of scans from F2050 survey used in geometry reconstruction

Appendix E

Automated Reconstruction Process

The automated reconstruction process consisted of twelve separate steps which have been listed below. A description of each step was provided and where further development was undertaken from the original source code this has been identified;

1) Data input,

To start the Point Cloud reconstruction process a pre-processed E57 Point Cloud file that included normal direction estimations, as described in subsection 4.5.1, was required. Using the command prompt in Windows 10 the executable file was run and the location of the input E57 file and output gbXML file, which was to be created in the process, was specified, see Appendix Figure E.1.

$$pcReconstruction.exe\ \text{--}input\ \text{"F2050.e57"}\ \text{--}outputgbXML\ \text{"}F2050.xml\text{"}$$

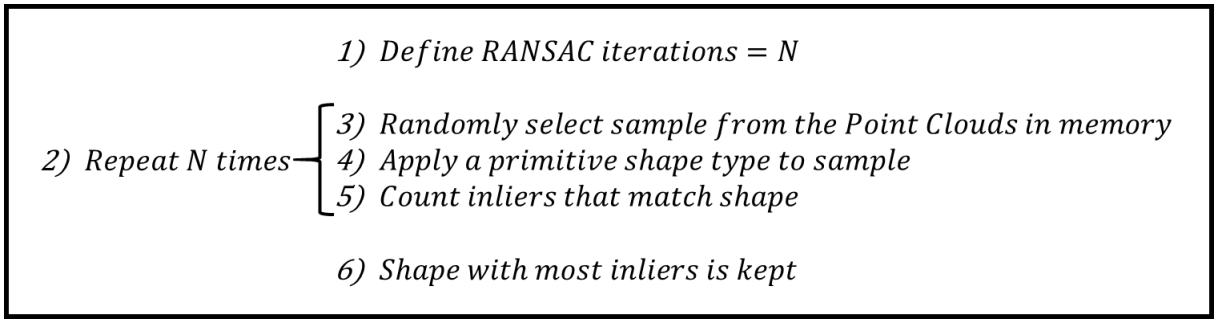
Appendix Figure E.1. User input to run the Point Cloud reconstruction process

2) Define Point Clouds,

The second step involved loading the Point Clouds into the programme's memory. The E57 file was read and the number of individual scans in the file were identified. The programme extracted the location of the scanner from each scan and the data associated with each point in all of the scans; coordinates and estimated normal directions.

3) Primitive Shape Identification,

With the set of Point Cloud scans loaded into memory the CGAL [160] library was used to detect primitive shapes within the Point Cloud. The library used a Random Sample Consensus (RANSAC) method to detect global shapes within the Point Cloud. The library supported the detection of planes, spheres, cylinders, cones and tori however only the plane detection was used in the Point Cloud Reconstruction. The algorithms developed for the library were produced by Schnabel *et al.*[177]. The basic RANSAC algorithm is outlined in Appendix Figure E.2.



Appendix Figure E.2. RANSAC algorithm

This simplistic version of the RANSAC algorithm has been observed to be impractical for large datasets as a large amount of computational time and power was required to test different shape primitives against the full input data set. To overcome this limitation, the CGAL library employed an *efficient* RANSAC algorithm which compared candidate shapes against a subset of the input data. The number of RANSAC iterations were also limited through a user defined probability of not identifying the best shape candidate to fit against the data and exploiting the normal direction of the data.

The error between a point and the candidate primitive shape was expressed by calculation of the Euclidean distance and the deviation in normal direction between the point and the candidate shape. User defined variables were used to define whether an error was within acceptable tolerances. The *efficient* RANSAC relied on five user defined parameters;

1. $epsilon$ was the maximum allowable tolerance of Euclidean distance between a point and a shape. This was set at 10mm for Point Cloud Reconstruction.
2. $normal\_threshold$ defined the maximum tolerance in deviation between the normal direction of a point to the normal direction of candidate shape. It was calculated by the dot product of the two vectors at the closest projection of the point onto the shape. This maximum tolerance was set to 5% deviation (i.e. $normal\_threshold = 0.95$).
3. $cluster\_epsilon$ was used to define the adjacency limit of points such that when inside the defined limit they were considered connected for a detected shape. This parameter was set at $500mm$. In this case the algorithm then considered two points separated by a maximum distance of $2\sqrt{2} \times 500mm$ as being adjacent

which provided a suitable cluster size when dealing with forms in the built environment.

4. min_points defined the minimum number of points required for shape detection to apply a candidate shape. This was set to 500 points for the reconstruction process.
5. $probability$ defined the probability of missing the largest candidate shape to fit against a dataset. This avoided lengthy programme runtime in the case where the best fit shape was randomly detected early during the iterations. This tolerance was set to 1%.

Once the *efficient* RANSAC had completed processing the Point Clouds stored in memory it had defined a set of primitive plane shapes, that were used to represent building forms such as floors, walls and ceilings, as well as associated data points from the input data.

4) Surface Identification,

Using the primitive plane dataset that had been created in the efficient RANSAC process each plane was assessed on its orientation. The current Point Cloud reconstruction method was only capable of handling vertical or horizontal surfaces. While this was identified as limitation it was still capable of capturing floor surfaces and most walls. Non-horizontal ceilings or curved features could not be processed with the current process or were simplified into horizontal planes.

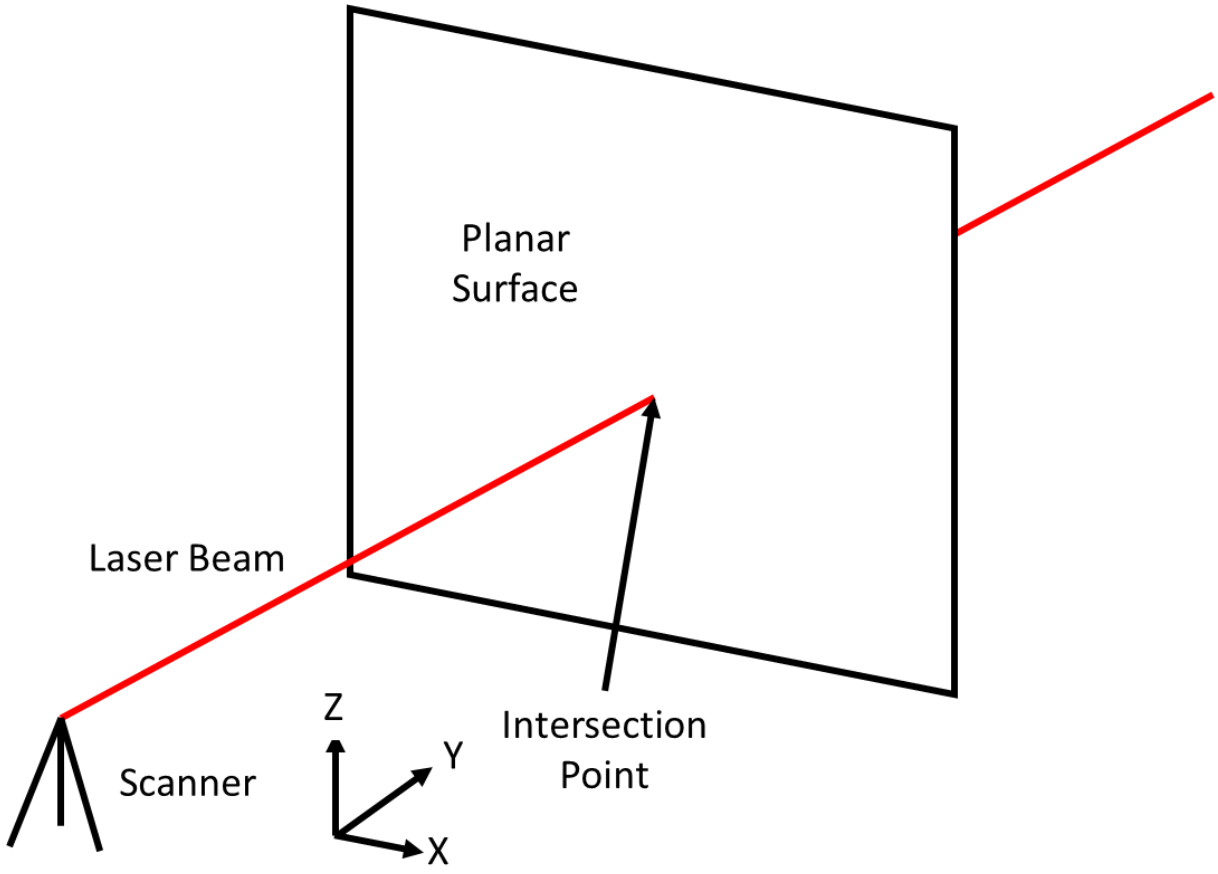
For Point Cloud reconstruction of F2050 a tolerance on detected primitives of 5° out of the horizontal or vertical plane was selected. Any primitive plane shapes that fell outside of this tolerance were discarded leaving only established horizontal and vertical planes.

5) Ray Casting,

Having identified surfaces within the Point Cloud data the next step involved determining whether a surface was visible to the laser scanner. This was based on the knowledge that when a laser scanner was positioned inside a room it had direct visibility of the walls of that room. In consideration of the surfaces that had been detected and the underlying

Point Cloud of the surface, the normal directions had been directed towards the scanner location.

Line of sight between a laser scanner origin and a surface associated with the scan captured by the same laser scanner was performed by using a technique known as Ray Casting. Ray Casting has been used as a technique extensively in the field of computer vision and graphics. The problem solved by ray casting was in identifying where a virtual ray, that was being directed from an origin point, intersected a plane, see Appendix Figure E.3. To solve this the mathematical definitions of a line and a plane in three dimensions needed consideration.



Appendix Figure E.3. Virtual ray intersecting a plane

In this application the virtual ray that emanated from a scanner location was considered as a three-dimensional line. The scanner location O was defined by the coordinates x_O, y_O and z_O in Appendix Equation E.1.

$$O = \begin{bmatrix} x_O \\ y_O \\ z_O \end{bmatrix}$$

Appendix Equation E.1. Coordinates of scanner location

In a similar manner the position vector $\overrightarrow{r_O}$ of the point O from the origin of the coordinate system was similarly defined by Appendix Equation E.2.

$$\overrightarrow{r_O} = \begin{bmatrix} \overrightarrow{x_O} \\ \overrightarrow{y_O} \\ \overrightarrow{z_O} \end{bmatrix}$$

Appendix Equation E.2. Position vector of scanner location

When a ray was sent out from the scanner location O its direction was defined by $\overrightarrow{vr}$ as per Appendix Equation E.3.

$$\overrightarrow{vr} = \begin{bmatrix} \overrightarrow{vr_x} \\ \overrightarrow{vr_y} \\ \overrightarrow{vr_z} \end{bmatrix}$$

Appendix Equation E.3. Direction vector for ray

The three-dimensional line that the ray travelled along was then described by Appendix Equation E.4 where E was a constant.

$$\vec{r} = \overrightarrow{r_O} + E\overrightarrow{vr} = \begin{bmatrix} \overrightarrow{x_O} \\ \overrightarrow{y_O} \\ \overrightarrow{z_O} \end{bmatrix} + E \begin{bmatrix} \overrightarrow{vr_x} \\ \overrightarrow{vr_y} \\ \overrightarrow{vr_z} \end{bmatrix}$$

Appendix Equation E.4. General equation for three-dimensional line

Given two points $p_{pc,1}$ and $p_{pc,2}$ that both lay on the same plane that were described by position vectors $\overrightarrow{r_{O1}}$ and $\overrightarrow{r_{O2}}$ respectively, and also given the normal direction vector of the plane $\vec{n}$, the equation of the plane was determined using Appendix Equation E.5 as all vectors on the plane were always orthogonal to the normal direction on the plane.

$$\vec{n} \cdot (\overrightarrow{r_{O2}} - \overrightarrow{r_{O1}}) = 0$$

Appendix Equation E.5. Dot product between plane normal vector and vector parallel to plane

From Appendix Equation E.5, some simplifications were made to yield the general equation of a plane in three-dimensional space as per Appendix Equation E.6 where A, B,

C and D were constants and x, y and z represented the coordinates of any point that lay on the plane. The constants are determined by the normal direction of the plane.

$$Ax + By + Cz + D = 0$$

Appendix Equation E.6. General equation of a 3D plane

With Appendix Equation E.4 and Appendix Equation E.6 the point of intersection between a line and plane in three dimensional space was determined using linear algebra. If an intersection occurred, then a solution was found for Appendix Equation E.7 that determined the coordinate of the point of intersection.

$$A(\overrightarrow{x_O} + E\overrightarrow{vr_x}) + B\left(\overrightarrow{y_O} + E\overrightarrow{vr_y}\right) + C\ (\overrightarrow{z_O} + E\overrightarrow{vr_z}) + D = 0$$

Appendix Equation E.7. Equation to determine intersection between a line and plane

After all Point Cloud scans had undergone a ray casting process three steps were taken to establish whether shapes surrounding a scanner represented the internal walls of the room;

1. Any shapes detected in the Point Cloud that were not hit were classified as external surfaces.
2. The normal direction of the shape being hit and the ray was expected to generally oppose one another if the shape under consideration was directly visible by the scanner. If this was not the case the ray was considered to hit the back of a surface.
3. The intersection point was compared with the bounding limits of shapes detected in the point cloud to determine if it was enclosed by the shape.

All candidate shapes that passed the above criteria were labelled as representing the internal walls visible to each laser scanner. Once the primitive shape detections had been labelled accordingly, the ray casting and labelling process was repeated a user defined amount to refine the results. For the Point Cloud reconstruction this was conducted twice.

Once the ray casting and categorisation of surfaces, that were either internal or external objects was completed, the surfaces were passed through a Gaussian filter. The primary purpose of the filter was to smooth out the underlying data points associated with each detected shape such that noise in the scan was reduced.

6) Grid Population,

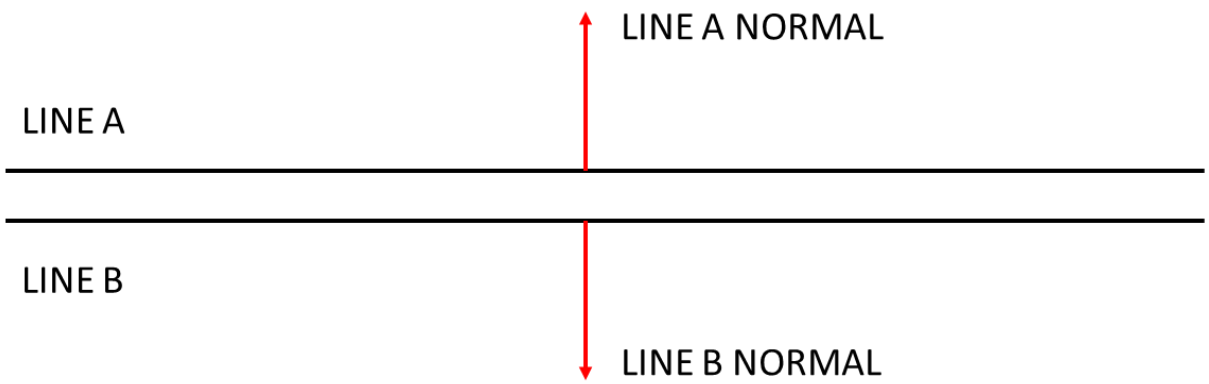
Once the Point Clouds had been categorised into points that were associated with internal and external surfaces, a floor plan was developed. Ignoring all points associated with external surface data, a bounding box was created and filled with all points that represented the detected internal surfaces.

A subsequent Gaussian point smoothing process was undertaken on the points within the bounding box grid to remove any potential outliers and noise. This resulted in a data set of internal surfaces, visible at each scanner location, that had a set of data points associated with it.

7) Line Generation,

From the internal surface and Point Cloud dataset the programme considered each vertical surface (i.e. walls) and generated a line vector to represent the direction of the wall in a horizontal plane. The line vector extended to the limits of the grid bounding box. This line was offset from the measured surface to represent a theoretical centreline of the wall using an assumed thickness of $200mm$. This resulted in a dataset of one-sided wall candidate lines.

Subsequently each of the one-sided wall candidates was compared against all other one-sided wall candidates. A check was made to see if the wall candidates were parallel using the dot product of the surface normal directions. As illustrated in Appendix Figure E.4, two parallel walls that represented both sides of a single partition were expected to have equal but opposing normal directions. The dot product of two such normal directions would have equalled -1 and the programme allowed for a ±1° tolerance in wall parallelism.



Appendix Figure E.4. Comparison of one-sided wall candidates

If the walls were considered to be parallel and opposing surfaces, a check was made to identify if the walls had at least $1m$ of overlap in length. A final check was made to establish if the walls were separated by more than $600mm$. Wall candidate pairs with an overlap greater than $1m$ and separation distance less than $600mm$ were considered as two-sided wall candidates representing either side of the same internal partition. On completion of this assessment any one-sided wall candidate that was not also a two-sided candidate was considered to represent the internal surface of an external wall at an assumed wall thickness of $200mm$.

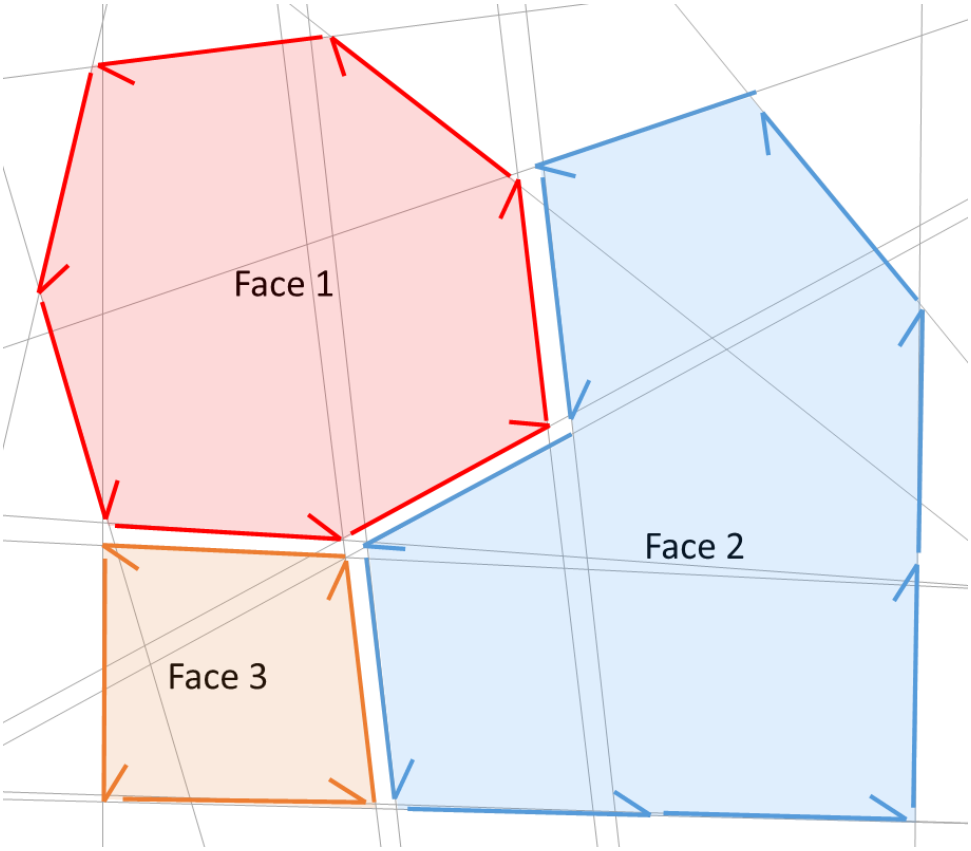
8) Wall Elevation,

Wall elevations were determined through consideration of horizontal surfaces and their underlying data points in the Point Clouds. As previously mentioned the programme was only capable of dealing with horizontal ceiling surfaces. Through consideration of the normal direction in the z-axis of a horizontal surface it was identified as either a floor or ceiling surface. Then if two opposing surfaces were found to overlap sufficiently they were used to define the elevation height of a single wall.

9) Layout Optimisation,

With a dataset that consisted of a grid of overlapping straight lines that represented the wall directions, each room was processed so that line intersection points were identified. Line segments, or edges, that represented the room boundaries were identified and used to define adjacent walls around the perimeter of a room. Any wall line segments not used were discarded which left the remaining edges that represented the floor plan of the

building. Each edge was represented by a direction, a length, start node, end node and next edge around the end node. The areas enclosed by the walls at both floor and ceiling height were labelled as faces in the floor plan as provided as an example in Appendix Figure E.5.



Appendix Figure E.5. Output of layout optimisation

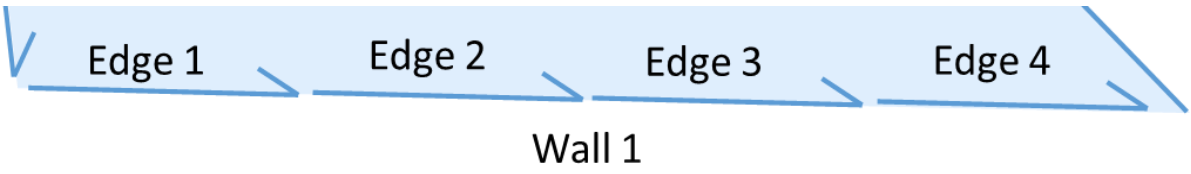
An additional relation between the edges included the identification of a twinned edge. Those edges that were twinned with a second edge represented an internal partition between rooms. Those without a twinned edge represented external walls.

10) Geometry Reconstruction (Developed further),

Having extracted edge and face data that represented the 2D floor plans of a building and the elevations of individual walls in the arrangement, the data was processed for reconstruction purposes. Individual wall elevations were previously calculated and these were now considered at a room basis by finding a common mean height among all of the walls that surrounded a single room. This was then used to define the distance between the opposing floor and ceiling faces to determine a room height.

The next step defined each room and the room definition was developed further than previous research. Originally a room was defined by a list of node coordinates where the walls of the room intersected, the floor and ceiling heights and the floor area of the room. In addition to these features the room definition was enhanced to include a list of edges, in order, around a room perimeter and the face index used for the floor and ceiling. Doing so provided a more complete picture of the elements that comprised a room and how they related to one another.

Following this enhancement in room definition, the individual walls of the reconstruction were defined, the process of which remained unchanged from previous research. It was possible that a single wall consisted of multiple edges as shown in Appendix Figure E.6. The wall definition included a list of these edges, the room index associated with the wall, the node coordinates at the start and end of the wall, vertical coordinates of the wall base and top, wall thickness, wall direction and wall length. In addition, an empty data store was created to reference any openings (i.e. doors, windows or holes) detected within the wall planar surfaces during a later stage of geometry reconstruction.

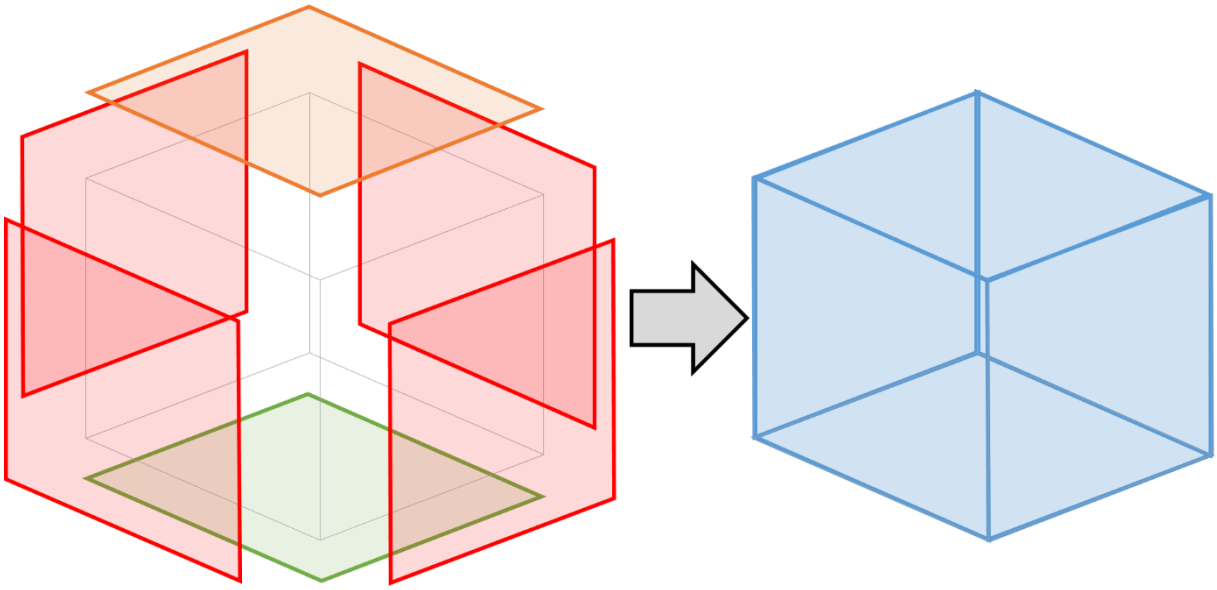


Appendix Figure E.6. Single wall that consisted of several edges

After wall definition the room floor and ceiling were fully defined. These included the nodes around the edge of the face, the assumed thickness of the surface and whether the surface was a floor or ceiling surface. As a further development to previous research the floor and ceiling definition was enhanced to reference the edges that surround the room perimeter. This simplified relating individual walls to the floor and ceiling surfaces.

Up to this point in the reconstruction step the individual elements had been defined for BIM applications only. This meant prior to conducting this research each reconstruction element was not well related to other components by keeping horizontal and vertical building elements completely separate (e.g. the walls were not easily related to the floor and ceiling). A key area of new development conducted for this research involved the

definition of thermal surfaces specifically for BEMS application. The thermal surfaces were defined such that they represented the centreline of any wall, floor and ceiling that were previously defined but, crucially, they were used to create watertight thermal volumes by relating all surfaces surrounding a volume to each other, see Appendix Figure E.7.



Appendix Figure E.7. Reconstructing thermal surfaces (right) instead of individual construction elements (left)

This thermal surface arrangement provided a fuller definition of the geometry of each thermal volume that allowed consideration of finite volumes for energy assessment in BEMS.

11) Opening Detection

The final step in geometry reconstruction was the detection of openings in surfaces. These included windows, doors and holes. To perform opening detection another ray casting process was conducted. The difference in the opening detection ray casting was that instead of identifying if a set of rays in all directions intersected candidate shapes, the rays were focused on each of the walls, that had been identified in previous steps, from a scanner origin position considered to have a view of that wall surface.

During wall generation, planes were detected and expanded to encompass the entire wall height and width, this had the effect of assuming a smooth surface where there may

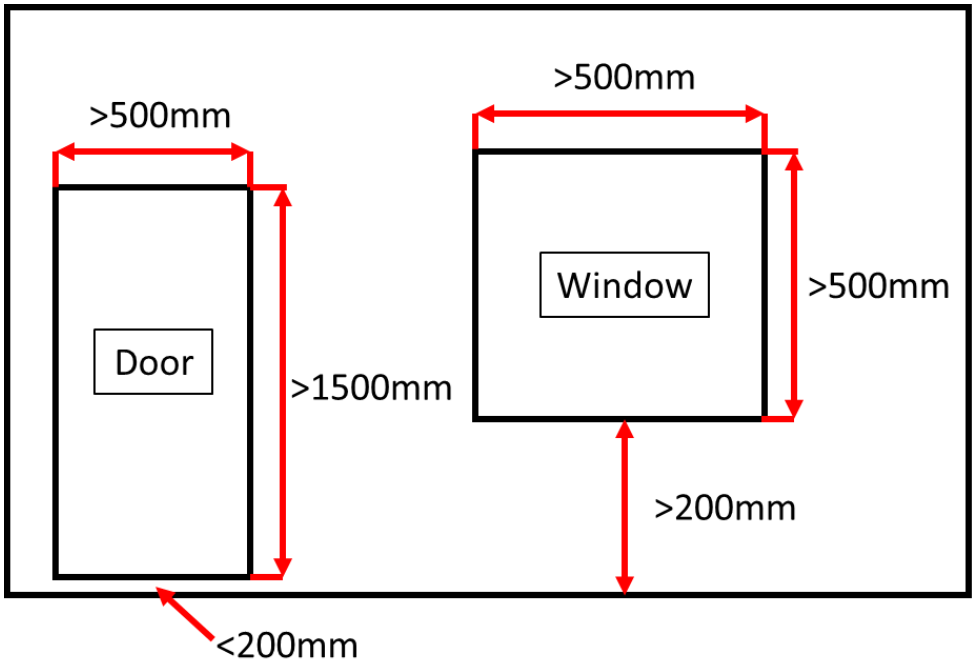
be discontinuities in the underlying Point Cloud due to the presence of an open door or a window.

During the ray cast a single ray was directed at each of the data points from the Point Cloud that were associated with an identified wall. The distance between scanner and data points were then compared with the distance between scanner and identified wall. If the Point Cloud data point was within $200mm$ of the identified wall surface, then no opening was established. Inversely if the data point was found to be greater than $200mm$ from the wall surface it was identified as a potential opening in the wall.

Once all data points associated with a wall had been assessed then clustering effects of data points associated with potential openings were considered. During this process step a resolution radius of $50mm$ between opening data points was assumed. This enabled the area of the wall associated with openings to grow from a set of discrete points into a physical bounding shape on a wall.

The final step in opening detection was classification. This was a crude but effective process where the opening type (e.g. door or window) was determined based on its dimensions. While it could not be applied to the full range of potential door and window shapes that occur within the built environment it was likely to cover most simplistic rectangular opening shapes.

Openings were classified as doors when the opening base was less than $200mm$ from the floor level, opening width greater than $500mm$ and opening height greater than $1{,}500mm$. Windows were similarly defined as having an opening base greater than $200mm$ above floor level and greater than $500mm$ in height and width, see Appendix Figure E.8.



Appendix Figure E.8. Opening classifications

Any detected openings that did not meet the door or window criteria described above were considered invalid openings. They were retained in the reconstruction data but not used further and did not affect the wall surface.

At the end of this step a Point Cloud had been automatically processed, room surfaces such as walls, floors and ceiling had been identified and their extents defined to create a set of thermal volumes, and the size and classification of openings within the wall surfaces had also been defined.

12) Export BEMS geometry (New Development).

To maximise the utility of the reconstruction process in this research that had been described thus far the results needed exporting in a format suitable for BEMS. This was a new development in this research and involved the exportation of the reconstruction information in a gbXML format.

gbXML was defined as an open-source file format that was written using XML and the file structure was defined using the approved XML schema definition [99]. It was used widely in sustainability research, as discussed in Chapter 2, and compared to other file formats, such as IFC, it was considered to be better suited to sustainability applications [10]. The

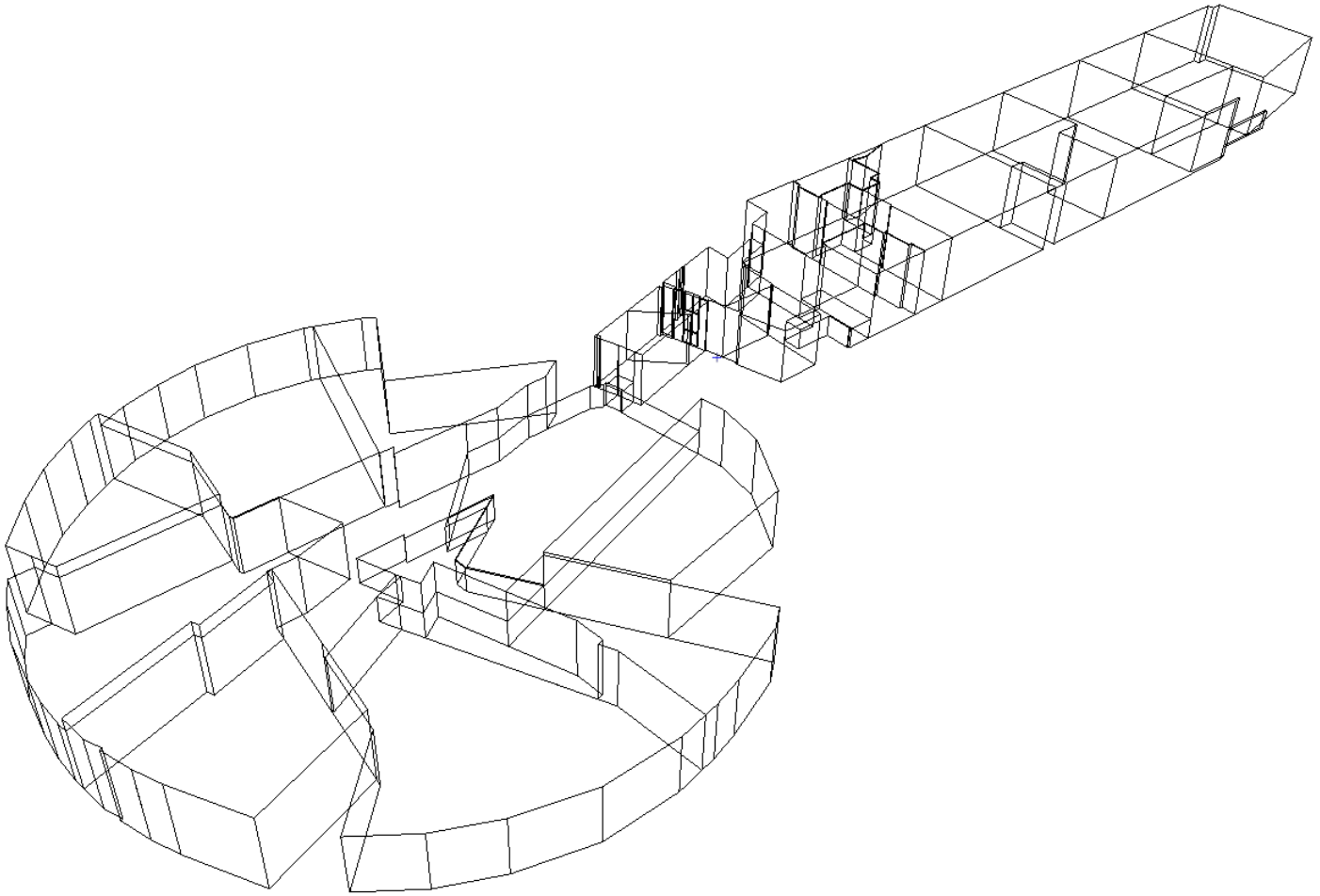
schema covered a wide range of information that could be defined for a building. This ranged from geometry and construction materials right through to HVAC systems and occupancy schedules. The schema version that was used in this research and was current at the time of writing is v6.01. One advantage of gbXML is that it was portable and widely recognised by leading BEMS software such as IES VE [16].

Appendix F

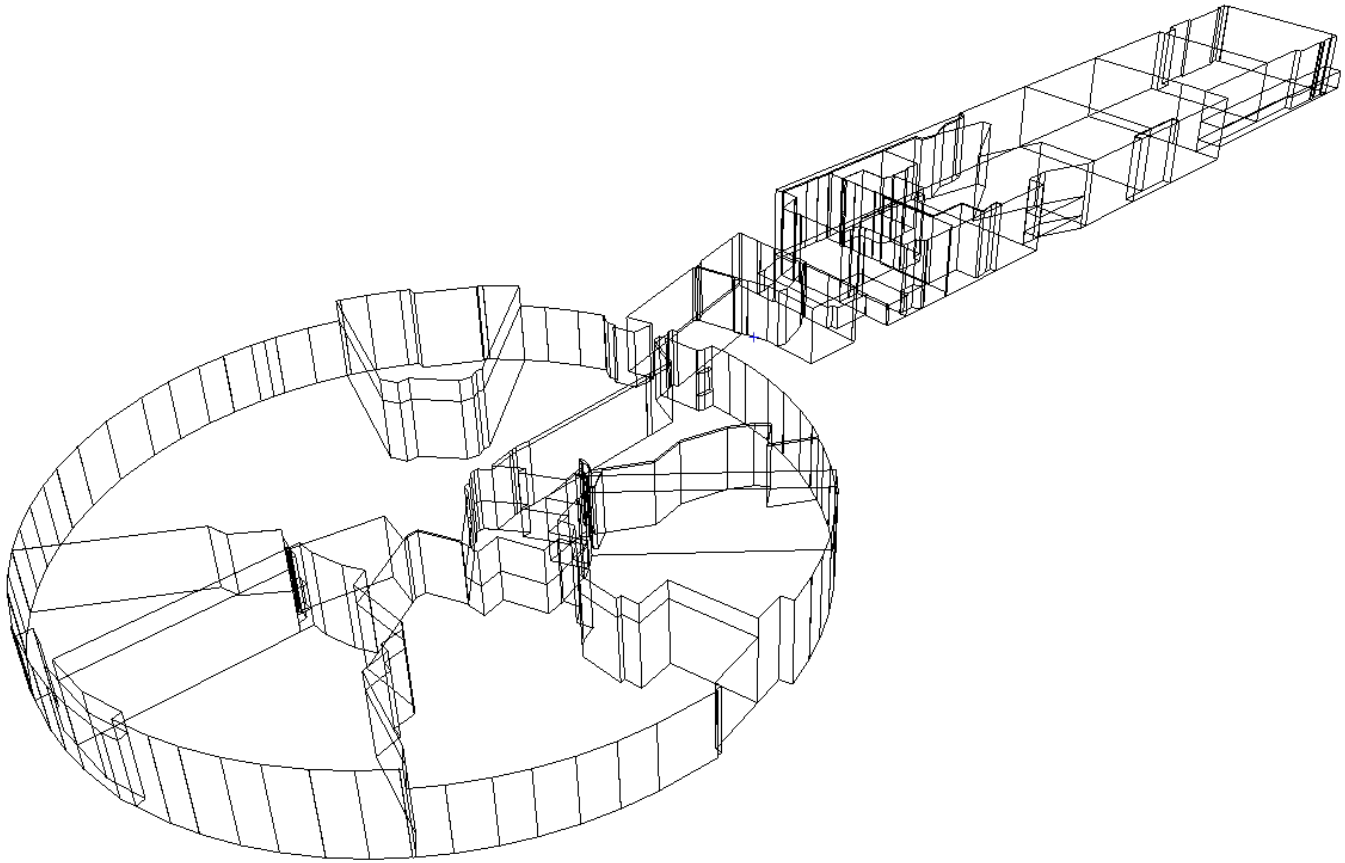
Voxel Filtered Reconstruction Results

This Appendix has presented the reconstructions of F2050 when differing levels of voxel filter had been applied to the original Point Cloud survey data. When attempted the reconstructions that obtained results used voxel filters of $0mm$ (i.e. no filter) (Appendix Figure F.1), $10\,mm$ (Appendix Figure F.2), $15\,mm$ (Appendix Figure F.3), $20\,mm$ (Appendix Figure F.4), $25mm$ (Appendix Figure F.5), $30mm$ (Appendix Figure F.6), $40mm$ (Appendix Figure F.7), $50mm$ (Appendix Figure F.8), $60mm$ (Appendix Figure F.9), $65mm$ (Appendix Figure F.10), $75mm$ (Appendix Figure F.11), $80mm$ (Appendix Figure F.12), $85mm$ (Appendix Figure F.13), $90\,mm$ (Appendix Figure F.14), $95\,mm$ (Appendix Figure F.15) and $100\,mm$ (Appendix Figure F.16).

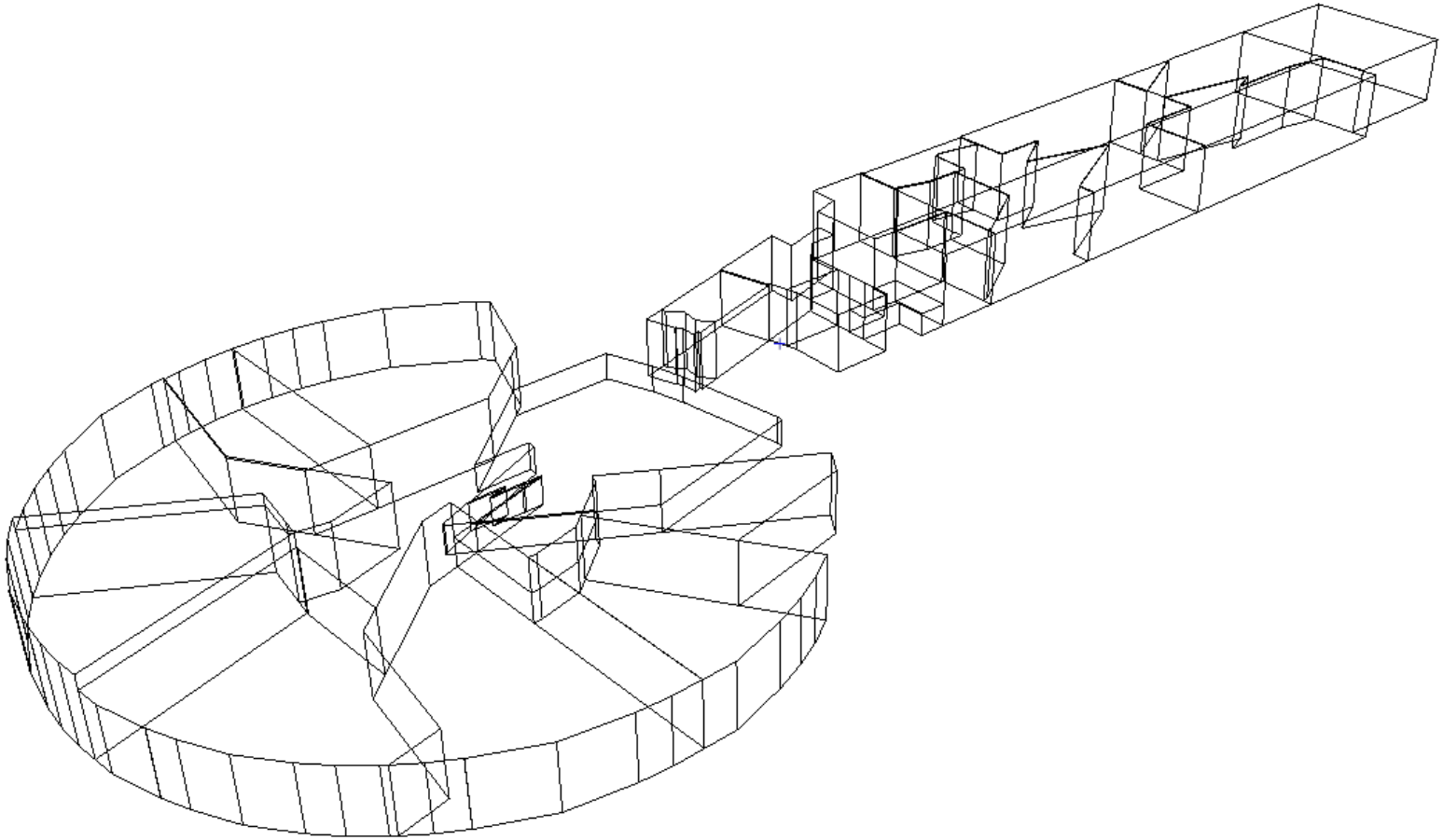
Four other voxel filters were attempted at $35mm$, $45mm$, $55mm$, $70mm$ however the reconstruction programme was either not able to complete or did not detect any geometry during processing of the Voxel filtered data.



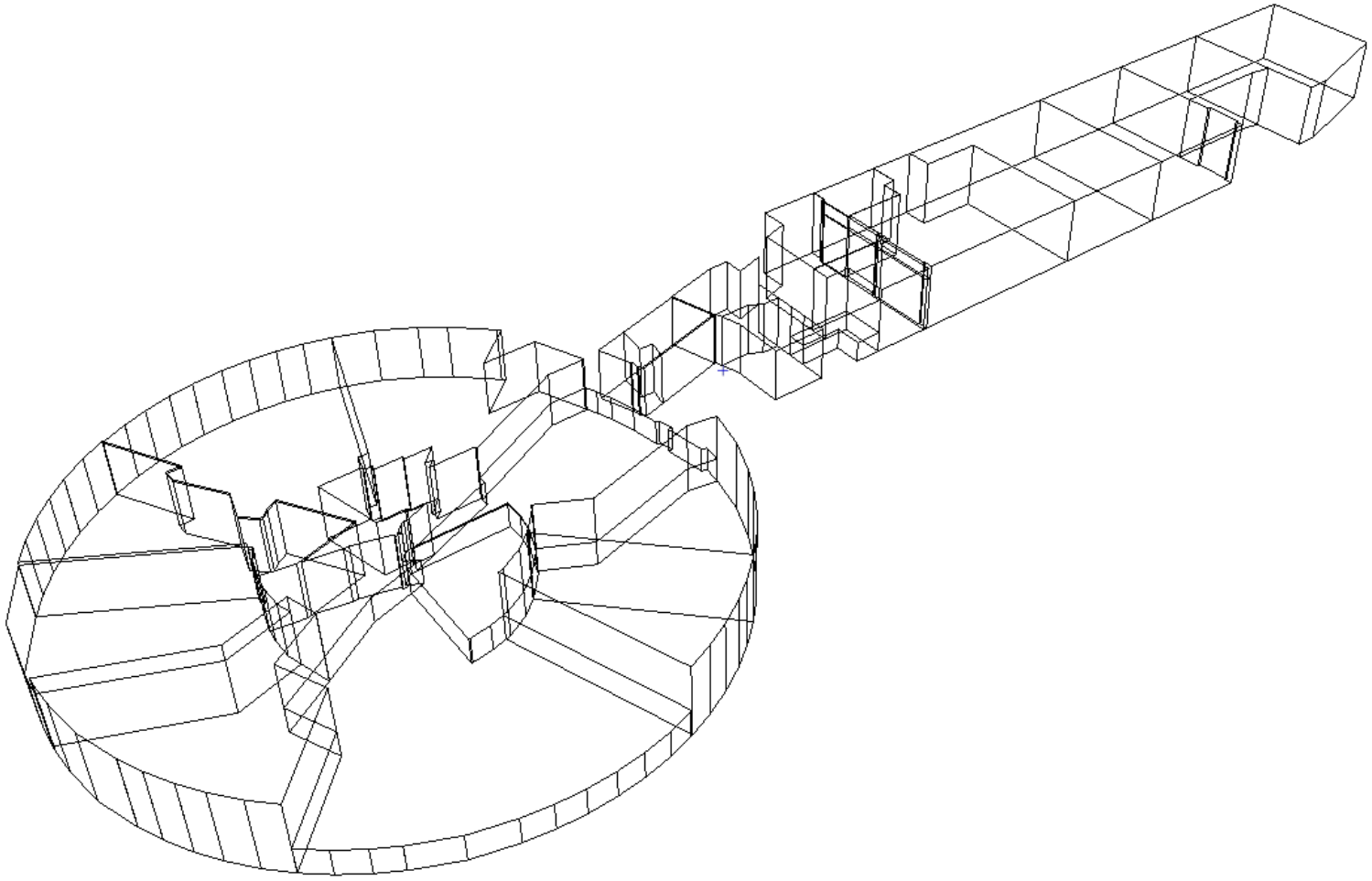
Appendix Figure F.1. F2050 ground floor reconstruction with no voxel filter



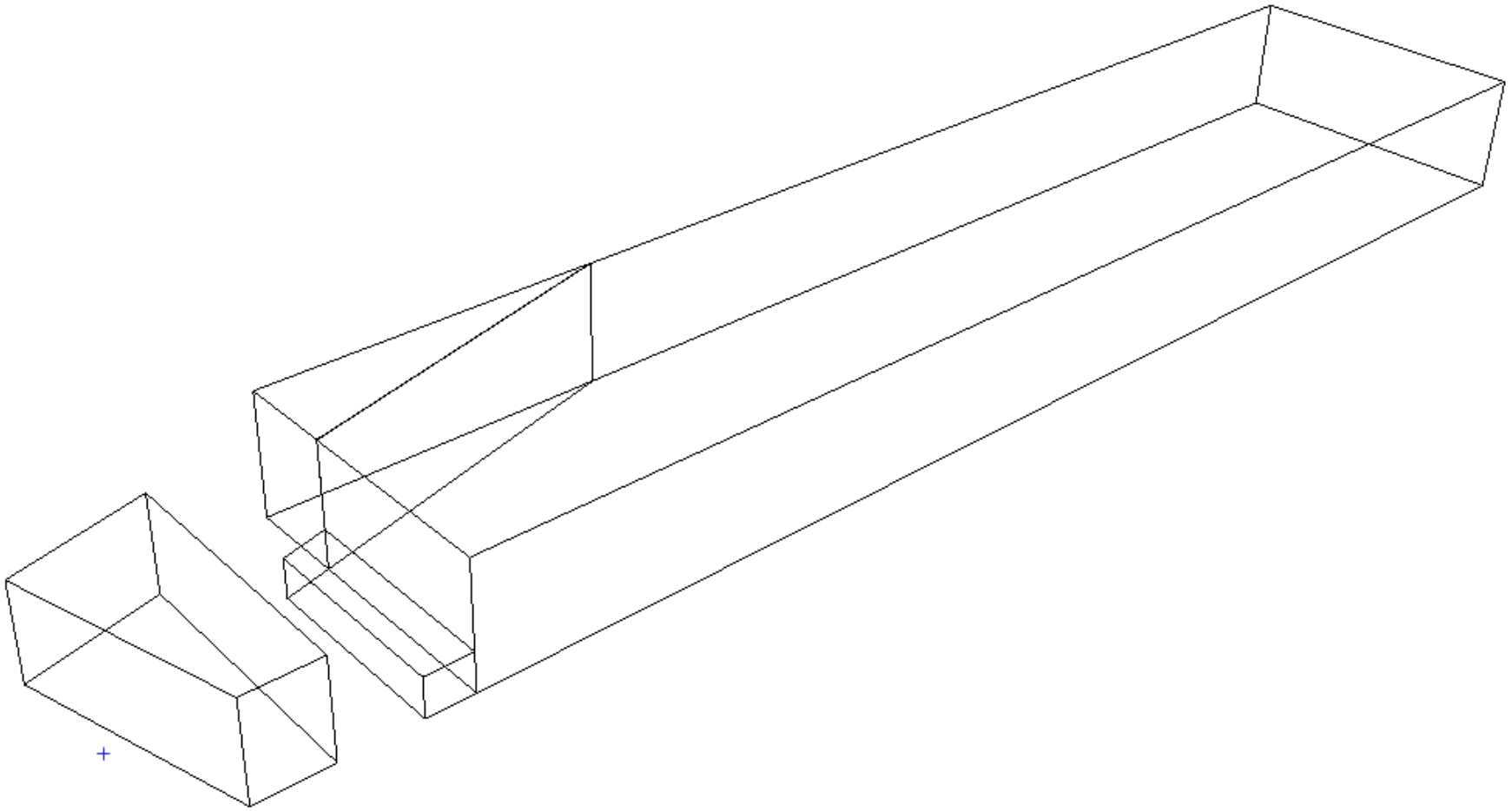
Appendix Figure F.2. F2050 ground floor reconstruction with 10mm voxel filter



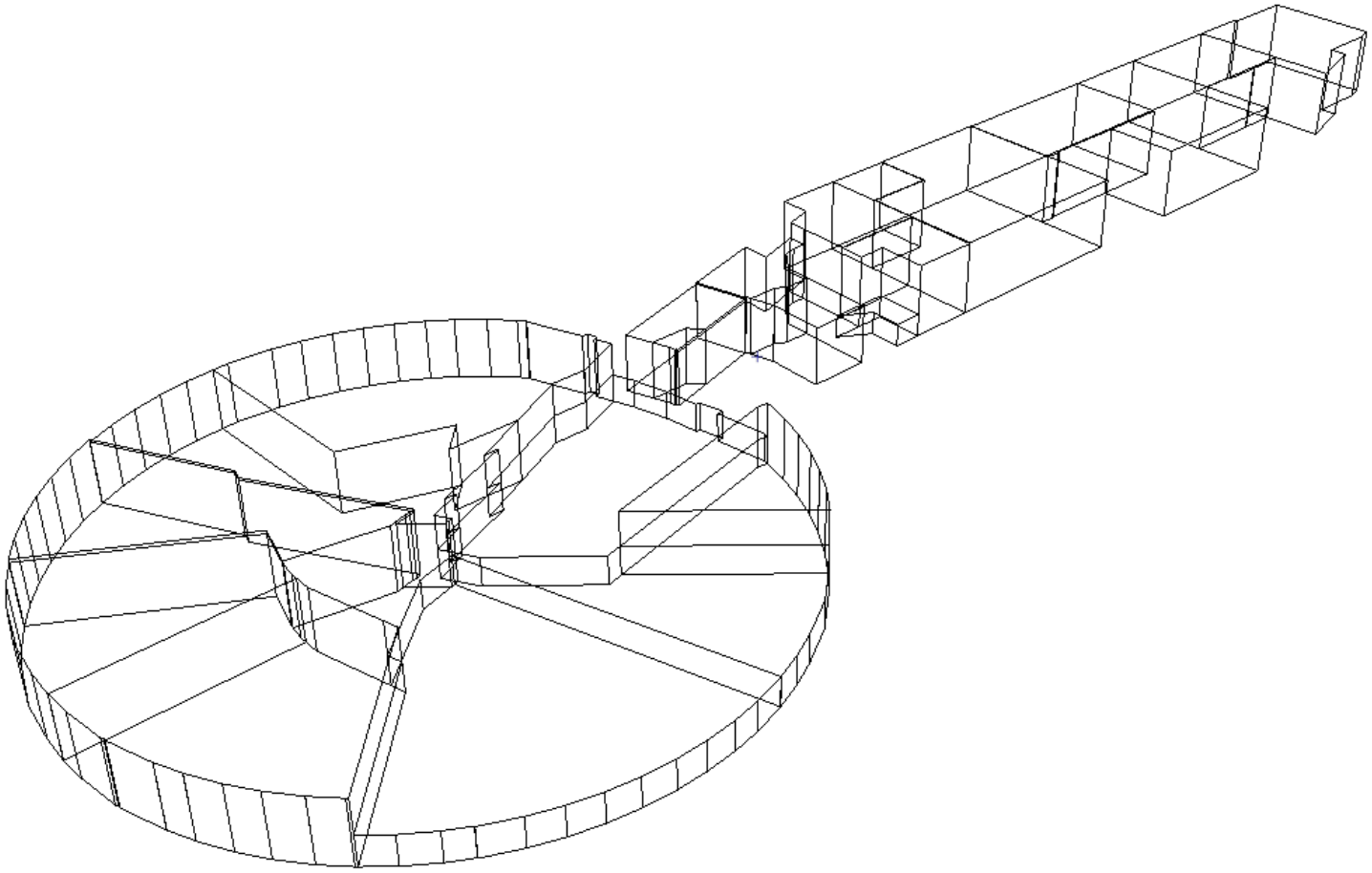
Appendix Figure F.3. F2050 ground floor reconstruction with 15mm voxel filter



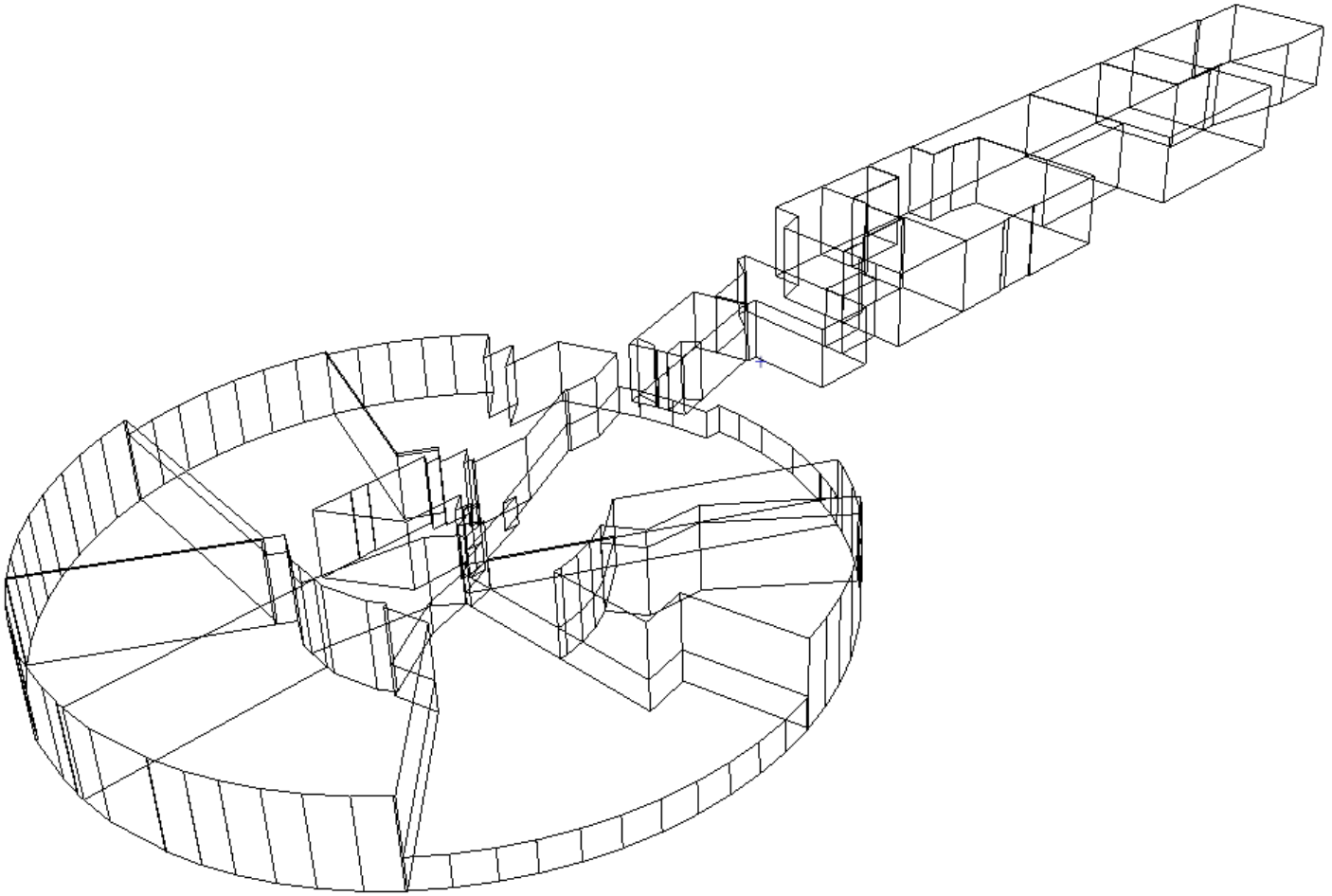
Appendix Figure F.4. F2050 ground floor reconstruction with 20mm voxel filter



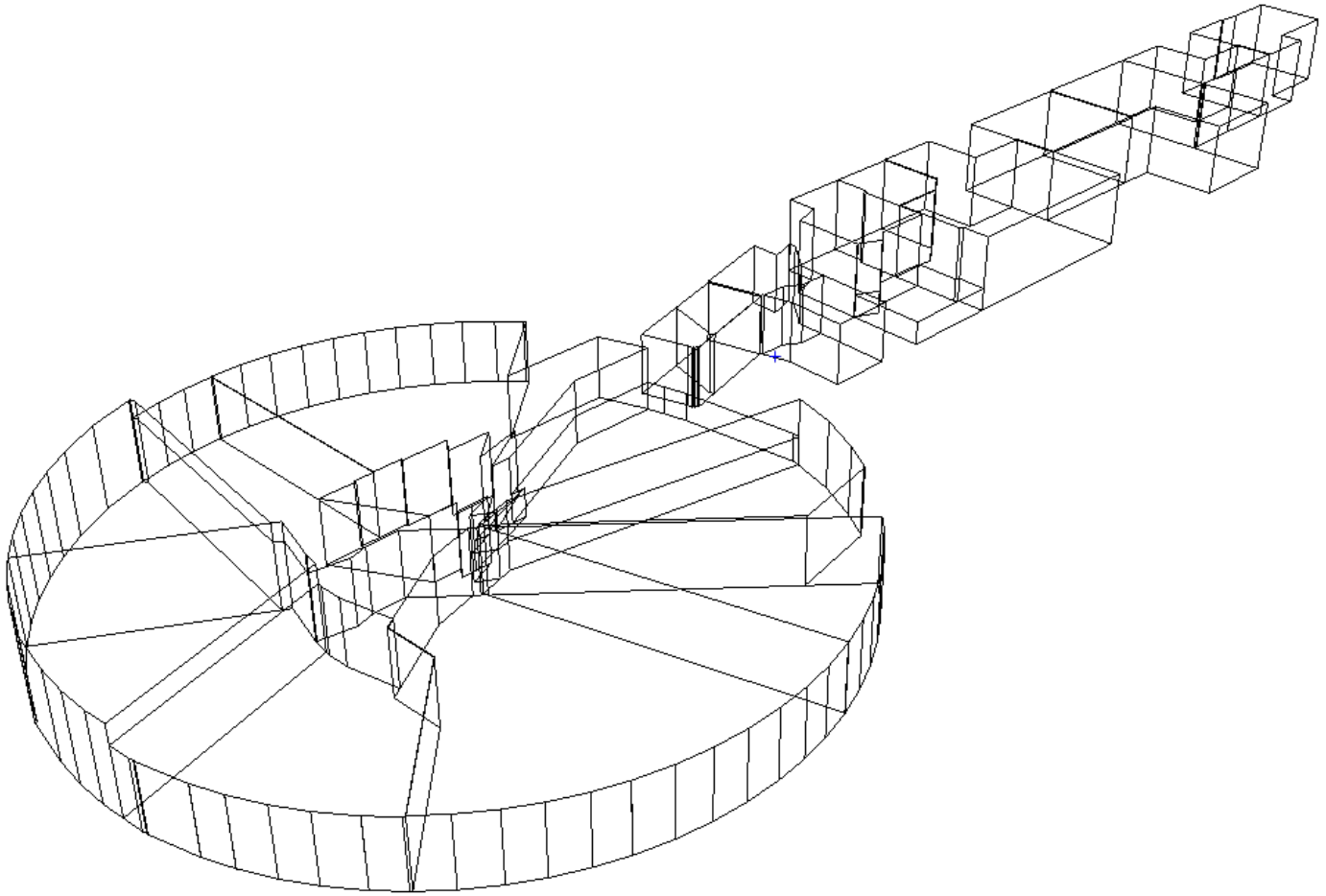
Appendix Figure F.5. F2050 ground floor reconstruction with 25mm voxel filter



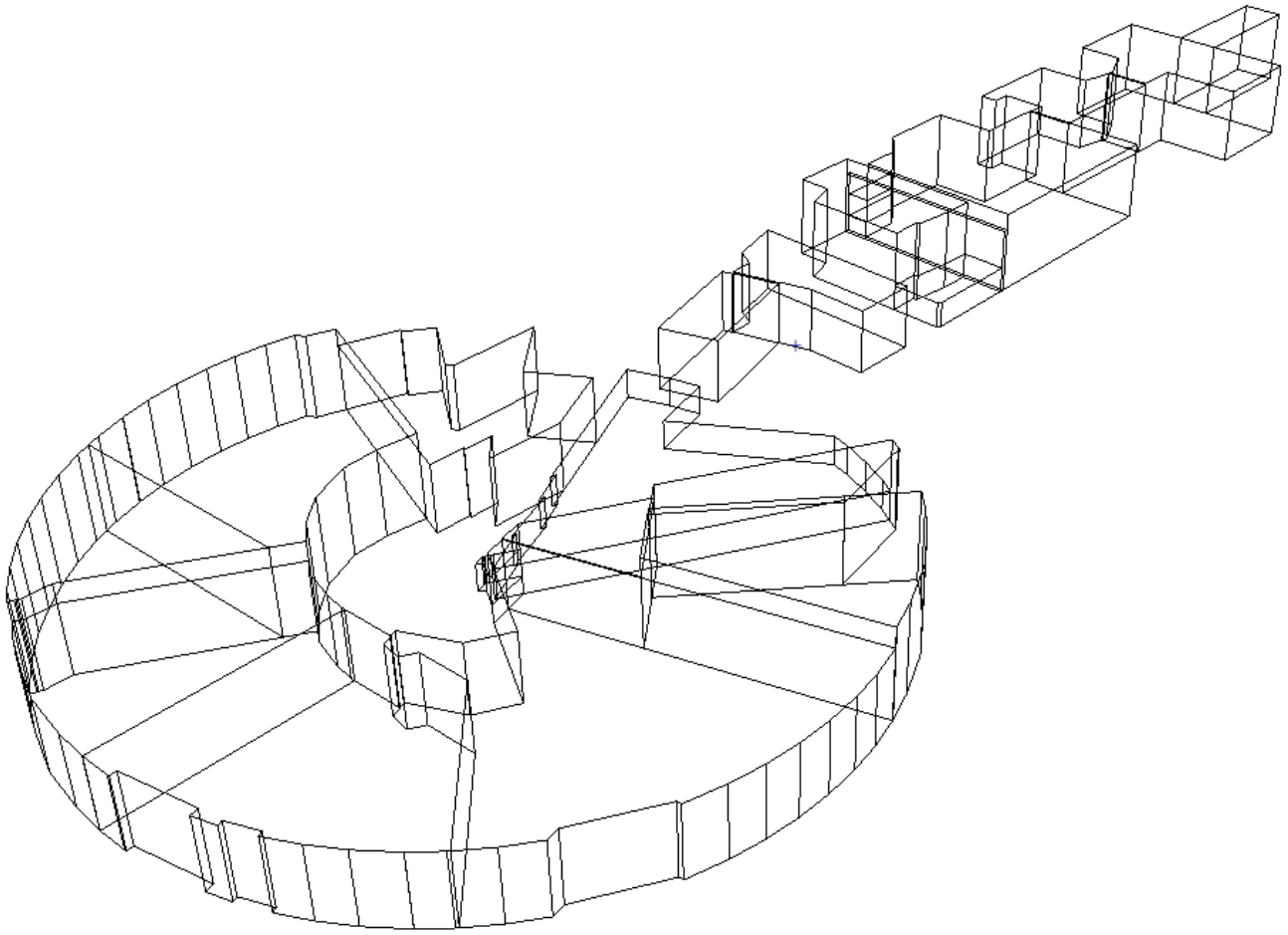
Appendix Figure F.6. F2050 ground floor reconstruction with 30mm voxel filter



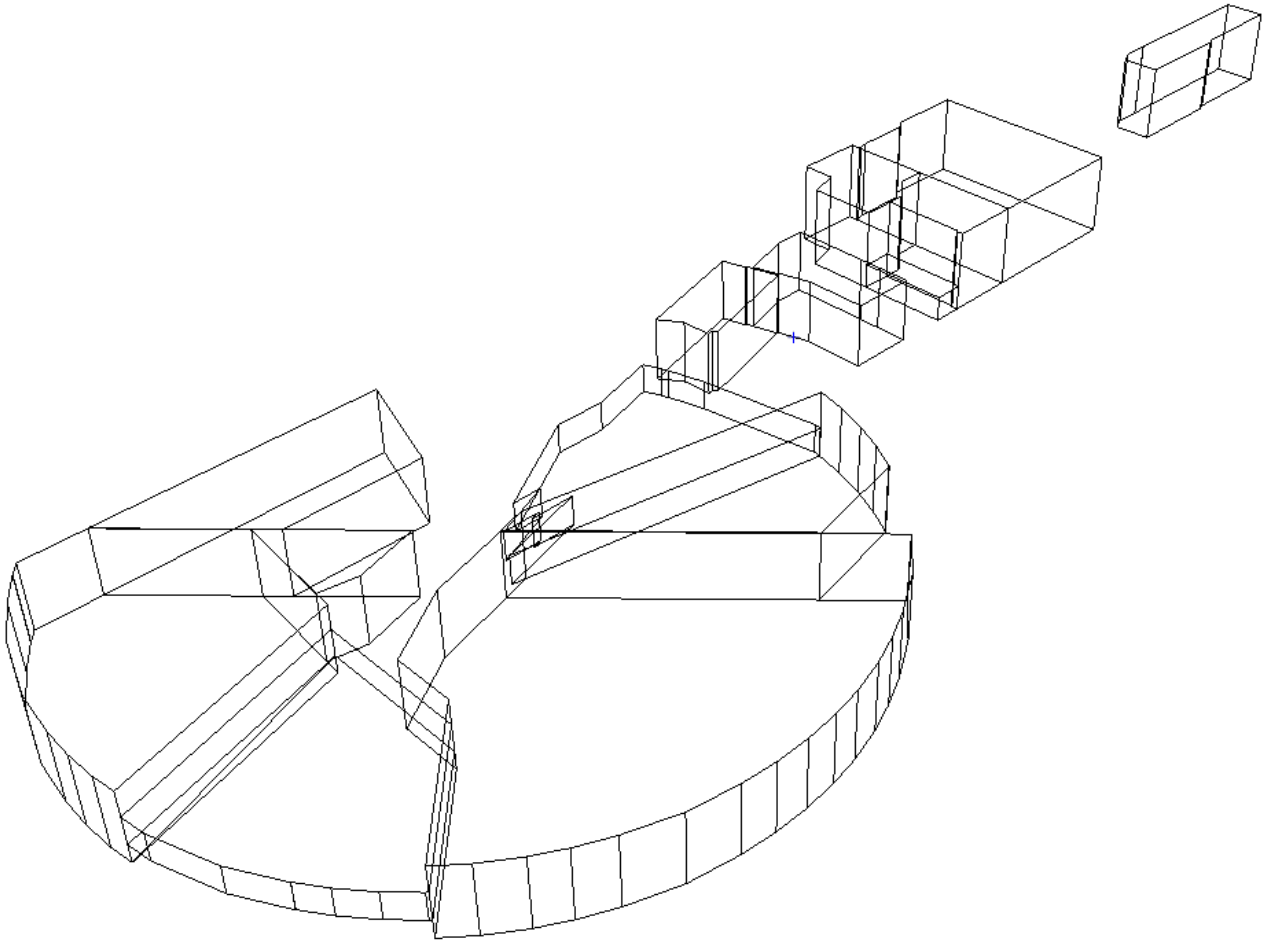
Appendix Figure F.7. F2050 ground floor reconstruction with 40mm voxel filter



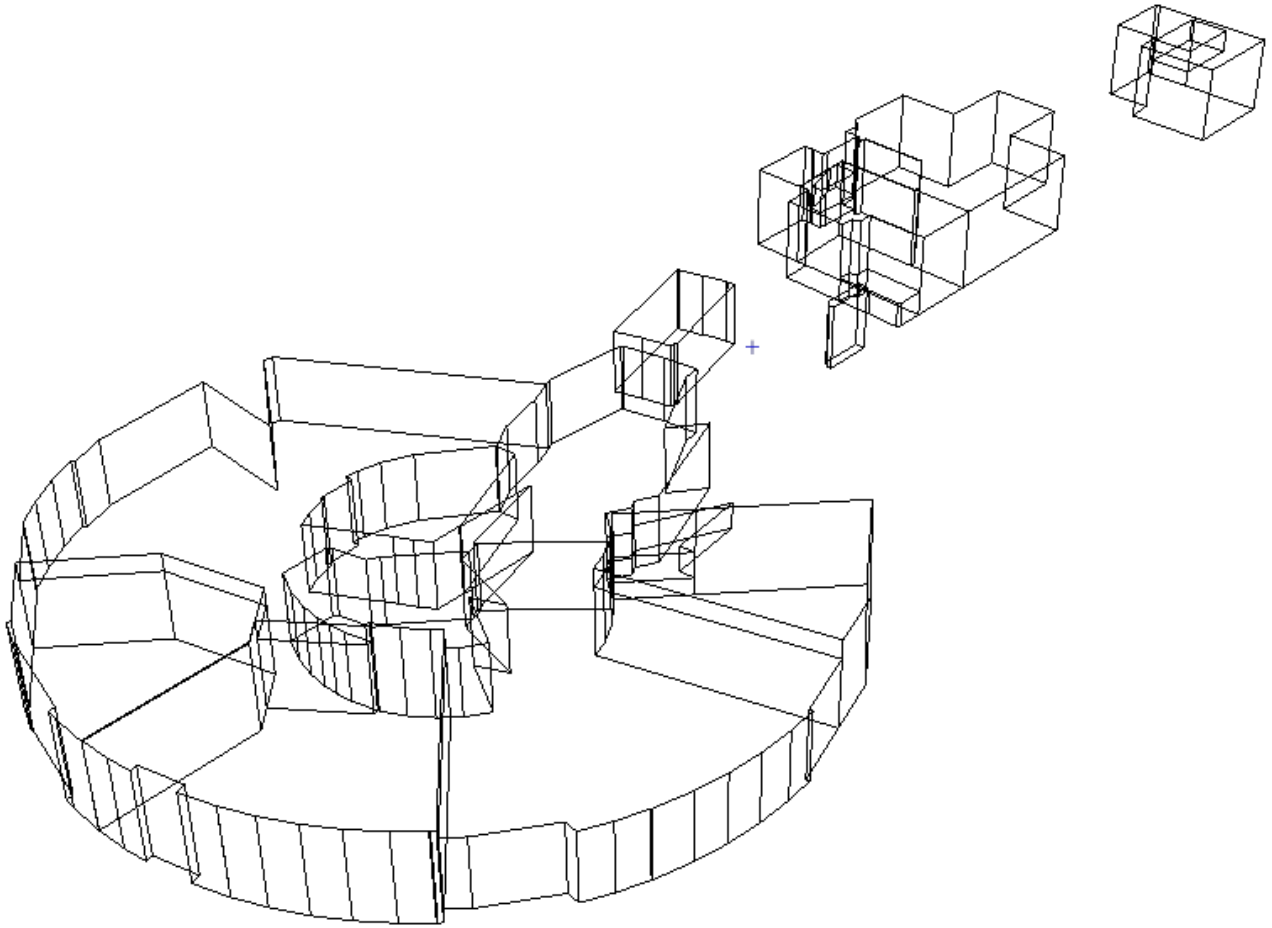
Appendix Figure F.8. F2050 ground floor reconstruction with 50mm voxel filter



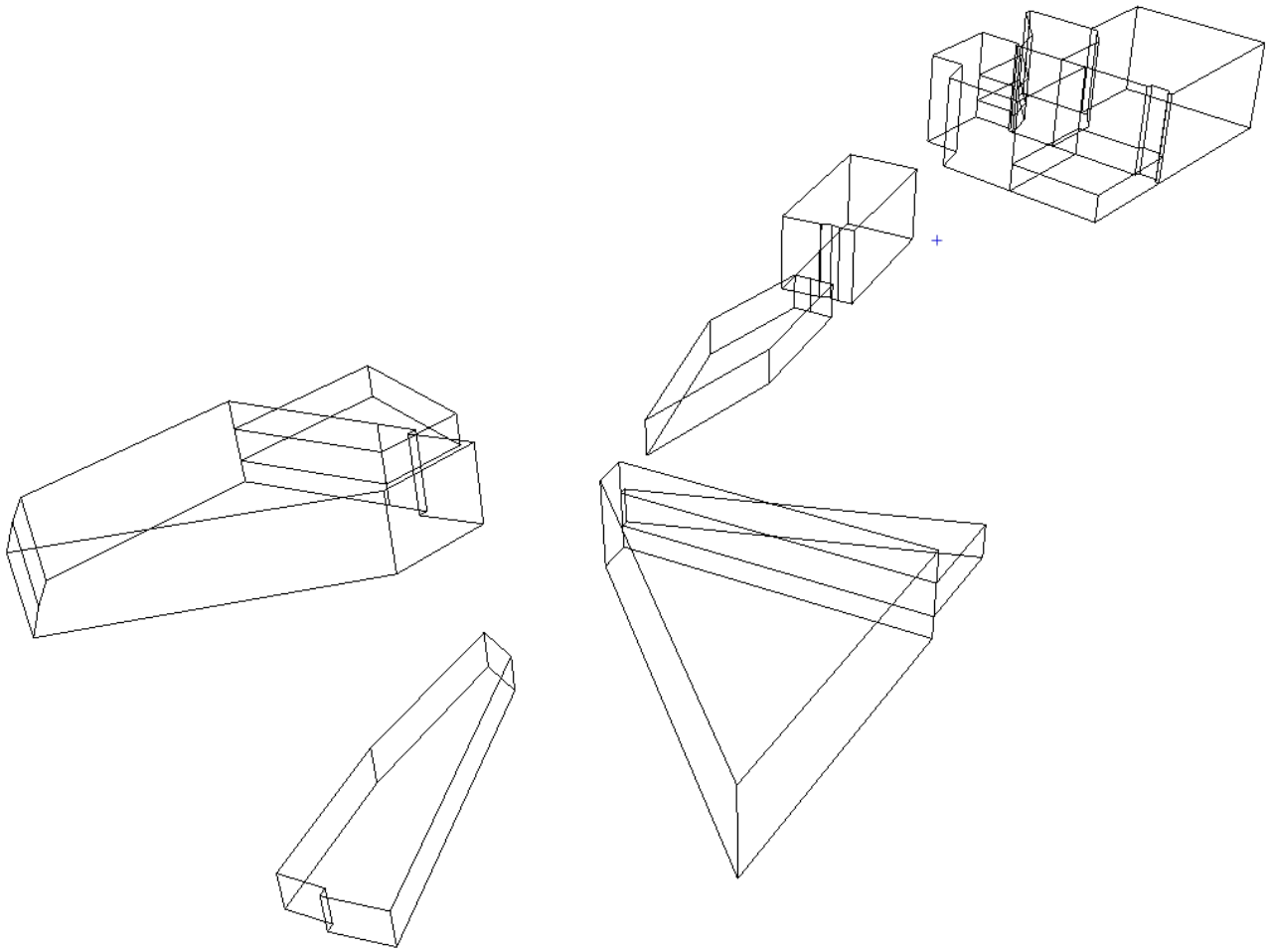
Appendix Figure F.9. F2050 ground floor reconstruction with 60mm voxel filter



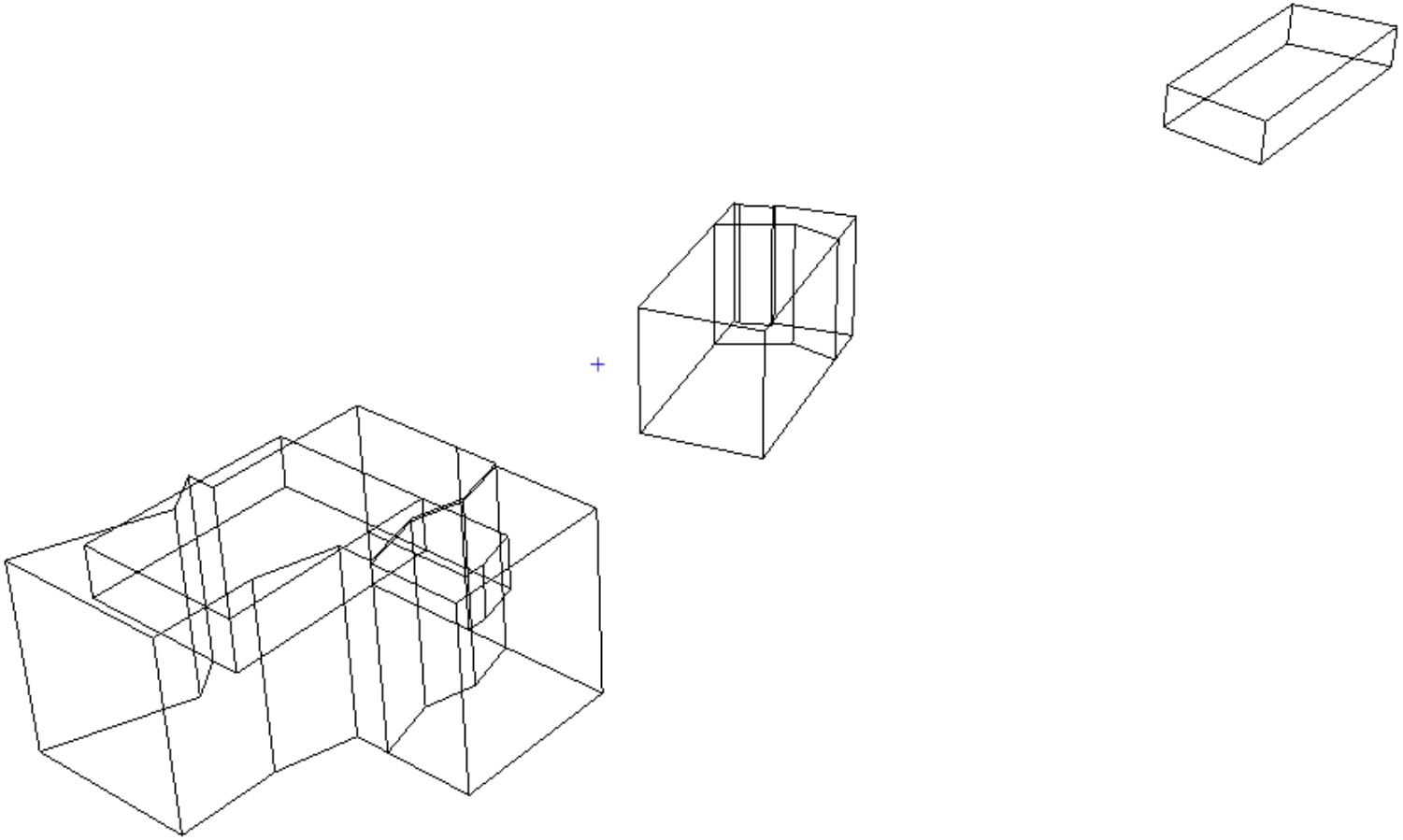
Appendix Figure F.10. F2050 ground floor reconstruction with 65mm voxel filter



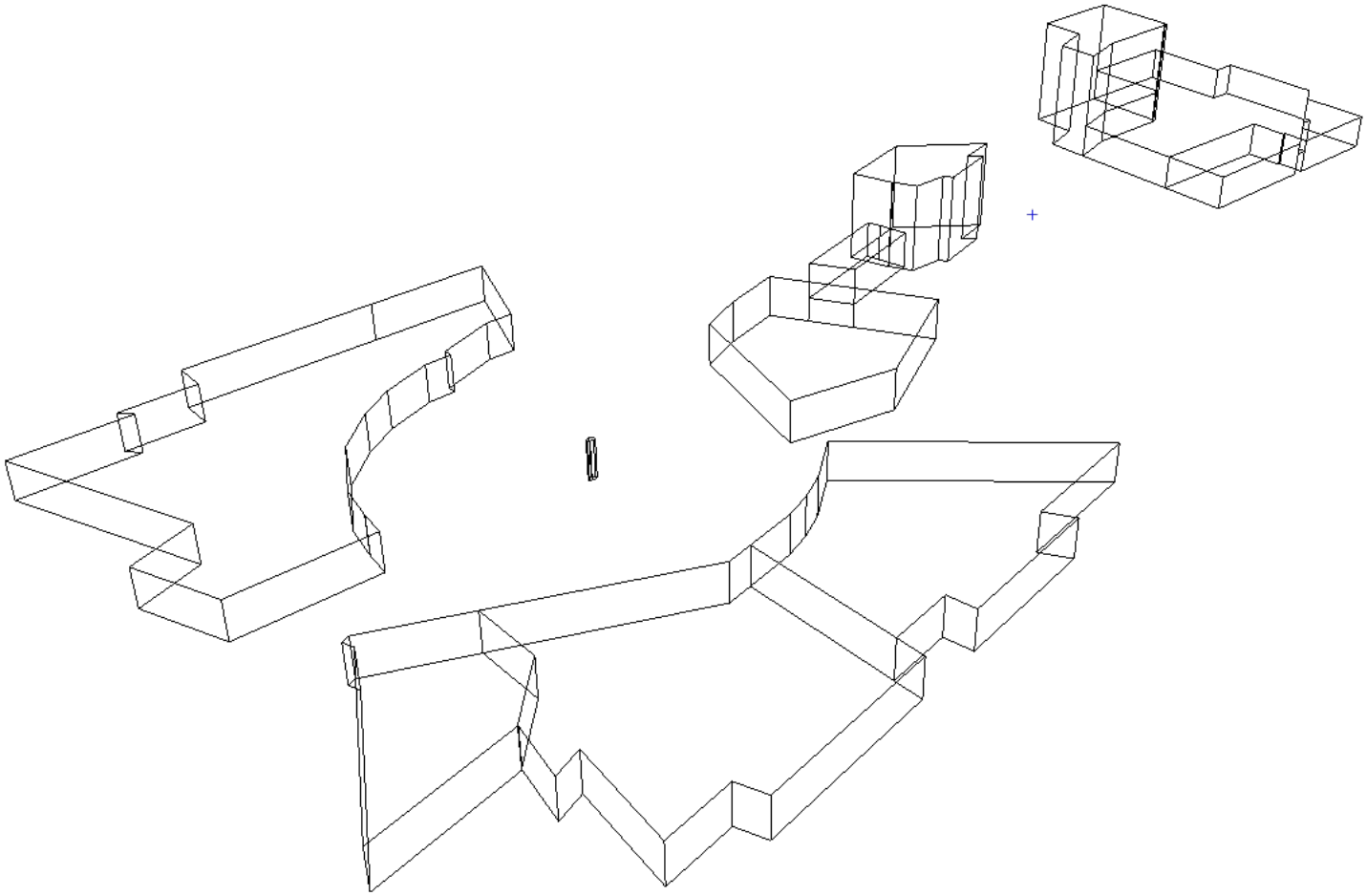
Appendix Figure F.11 F2050 ground floor reconstruction with 75mm voxel filter



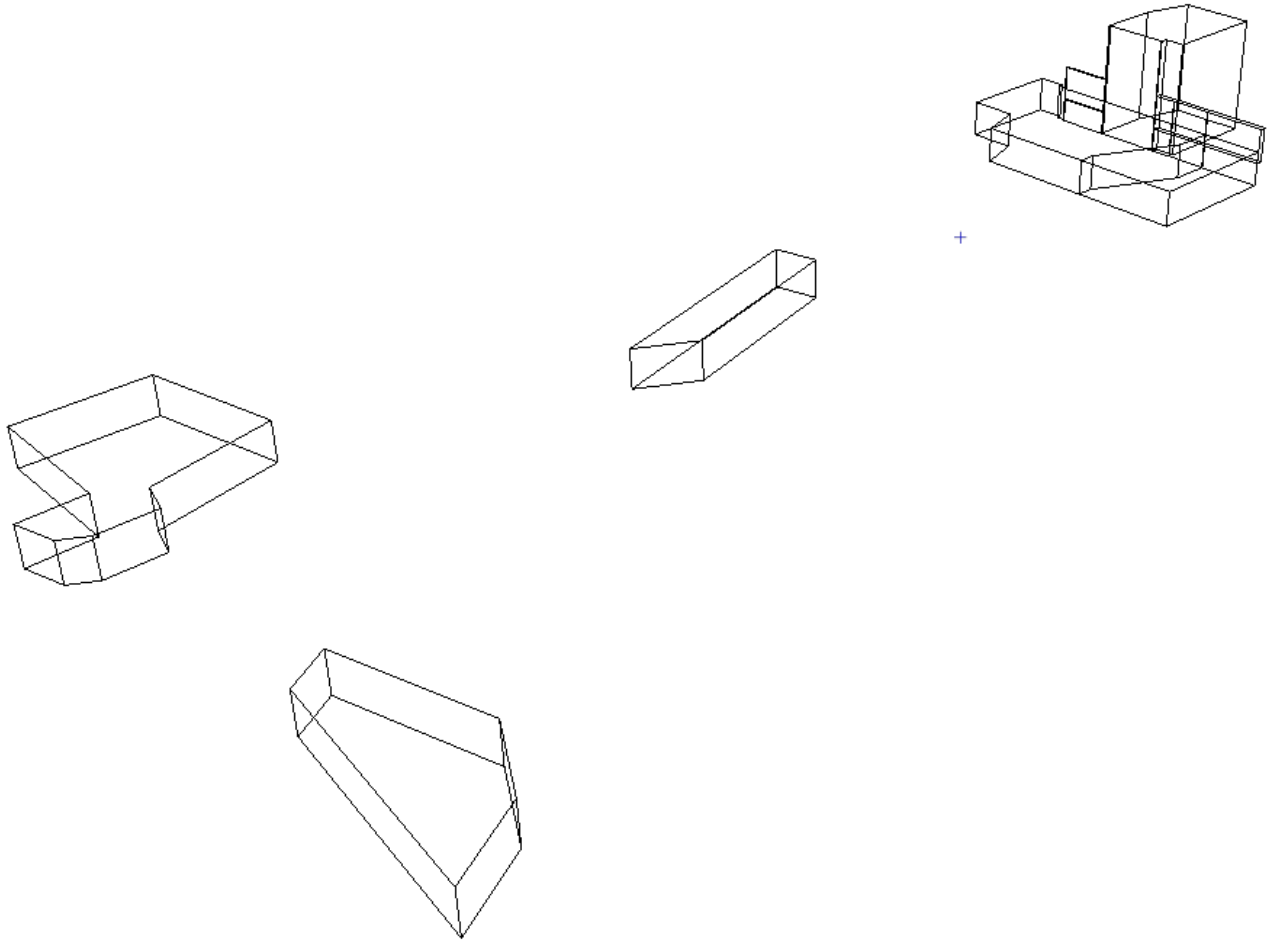
Appendix Figure F.12. F2050 ground floor reconstruction with 80mm voxel filter



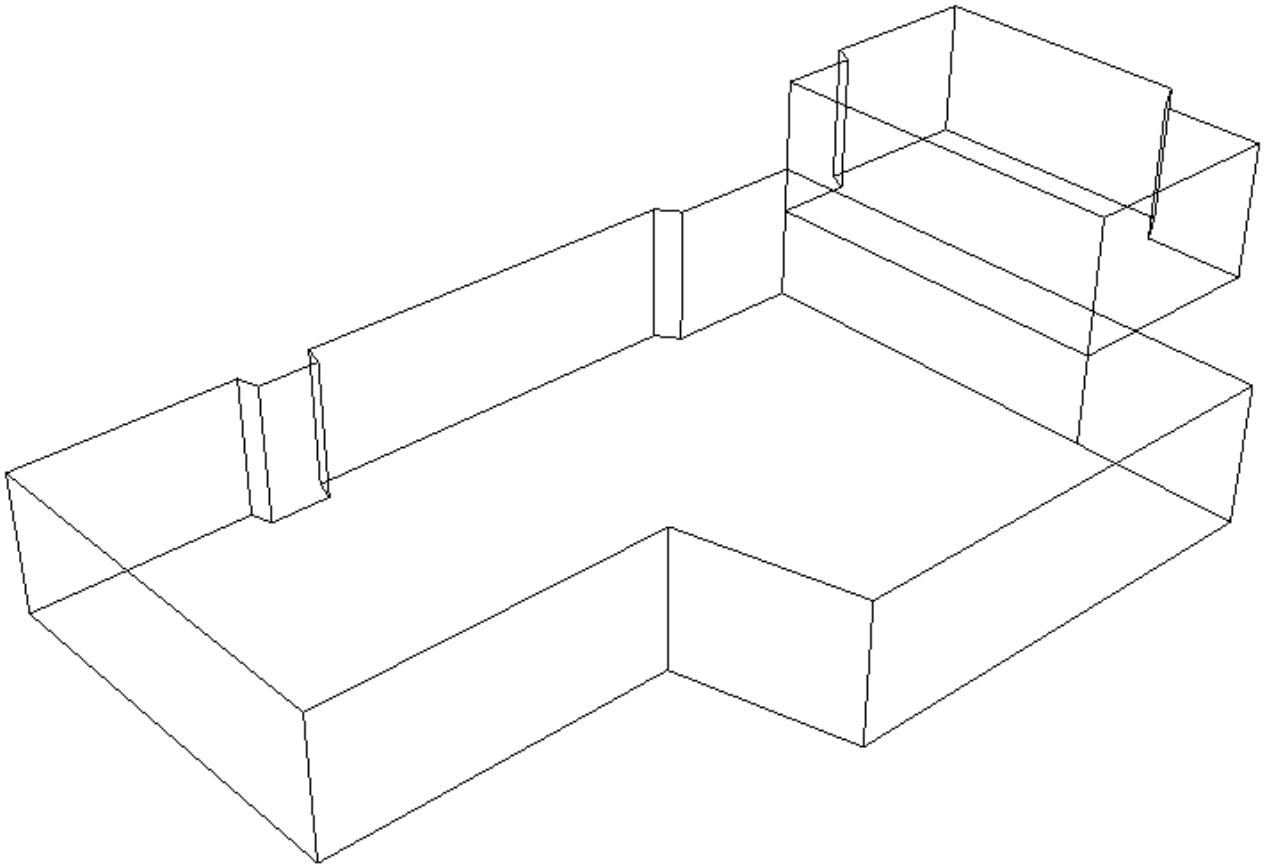
Appendix Figure F.13. F2050 ground floor reconstruction with 85mm voxel filter



Appendix Figure F.14. F2050 ground floor reconstruction with 90mm voxel filter



Appendix Figure F.15. F2050 ground floor reconstruction with 95mm voxel filter



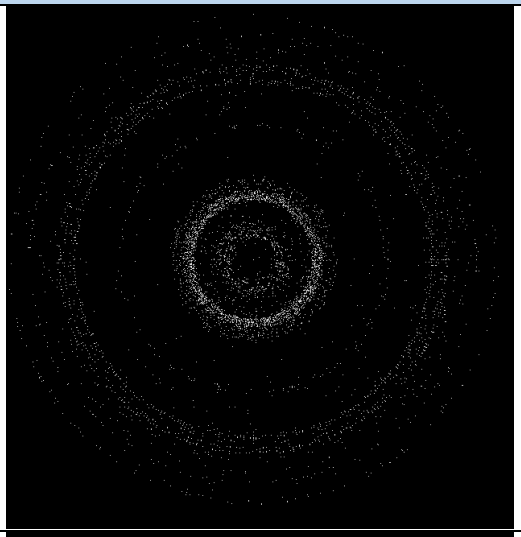
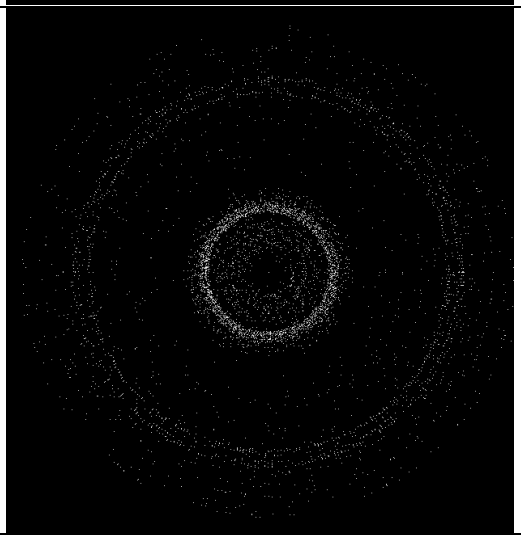
Appendix Figure F.16. F2050 ground floor reconstruction with 100mm voxel filter

Appendix G

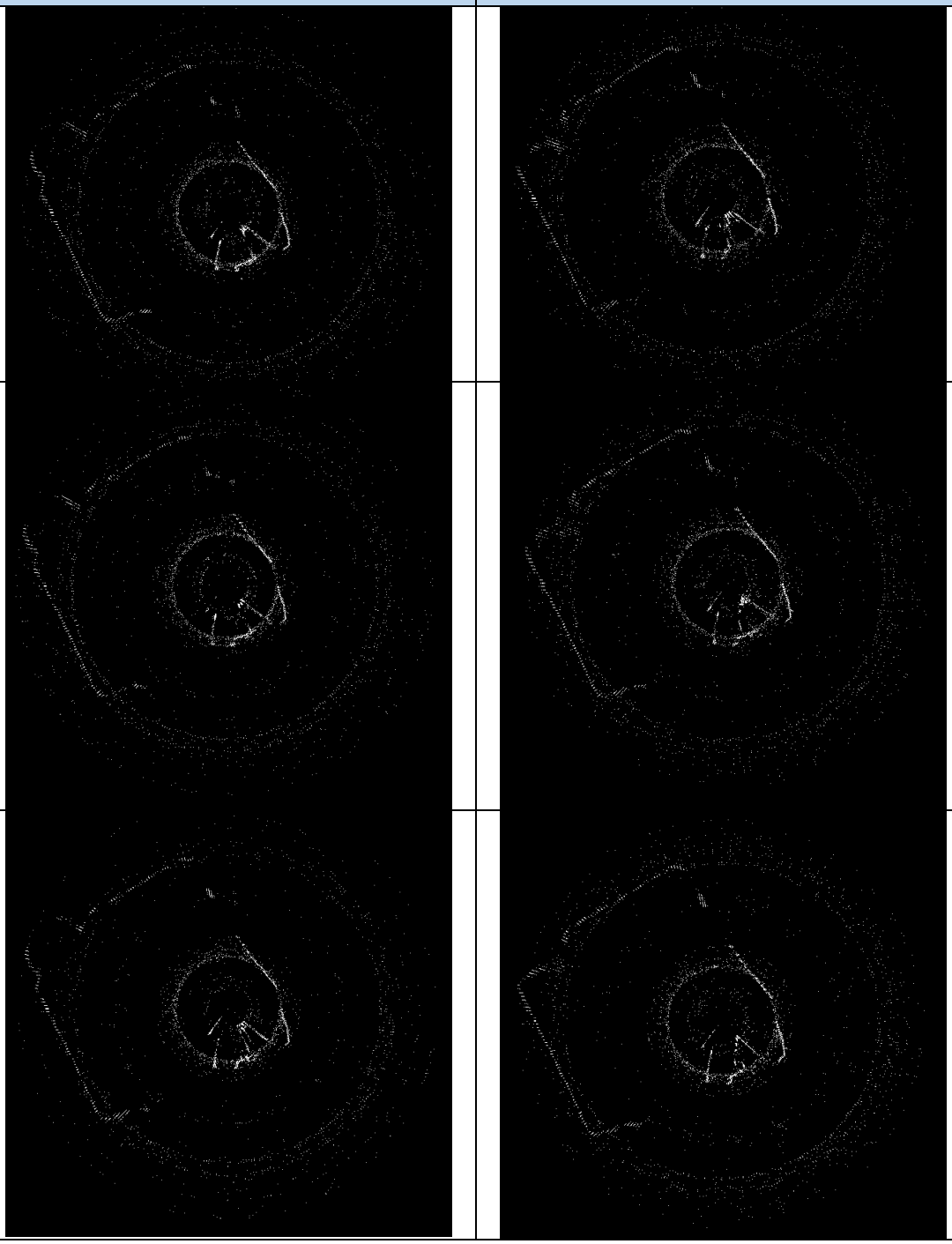
Encoder Calibration Results

This Appendix has presented the results of calibrating the value used for the datum encoder scale in Appendix Table G.1 as discussed in subsection 5.5.3.

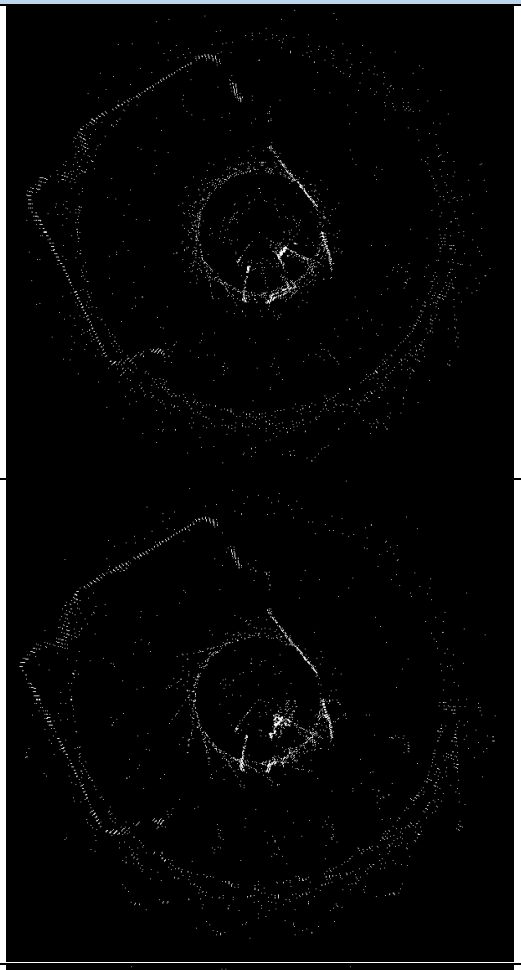
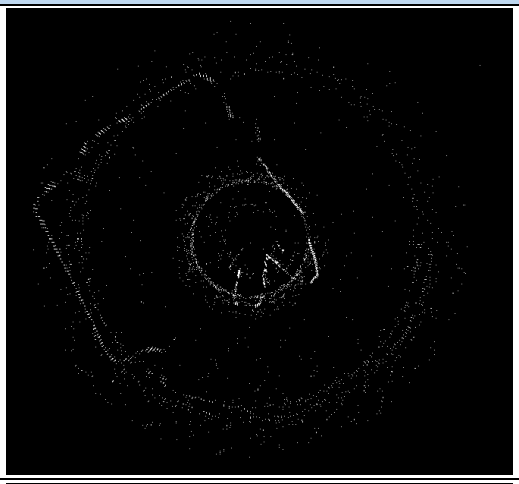
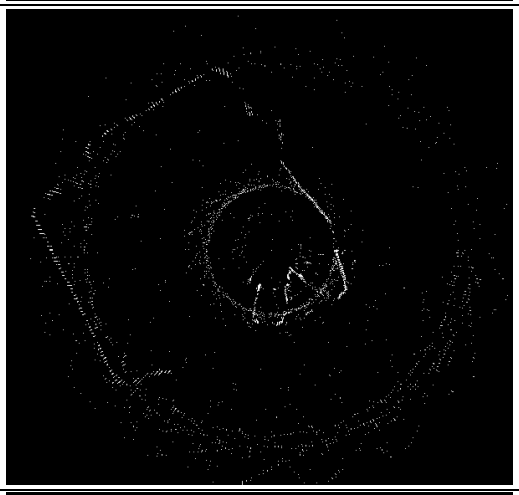
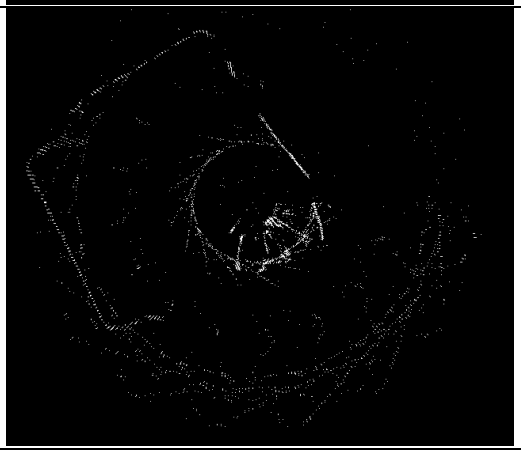
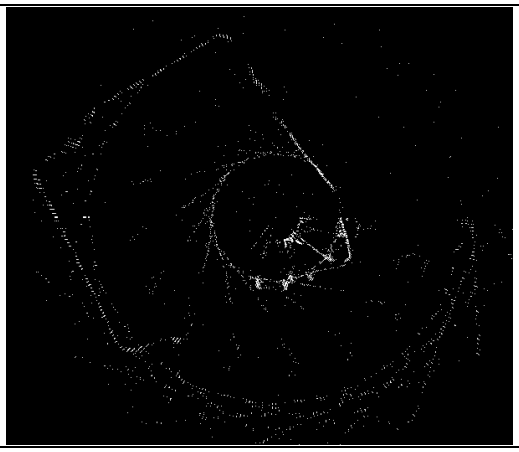
Appendix Table G.1. Encoder datum calibration

Encoder Scale	Test 1 – Timer Only	Test 2 – Timer & 12° Correction
0.1		Not conducted
0.2		Not conducted

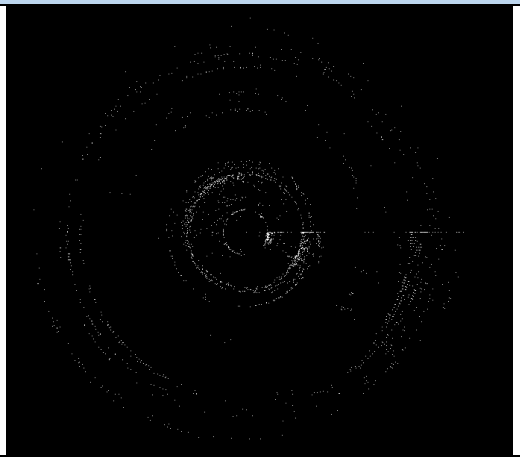
Appendix Table G.1. Encoder datum calibration

Encoder Scale	Test 1 – Timer Only	Test 2 – Timer & 12° Correction
0.3		
0.4		
0.5		

Appendix Table G.1. Encoder datum calibration

Encoder Scale	Test 1 – Timer Only	Test 2 – Timer & 12° Correction
0.6		
0.7		
0.8		

Appendix Table G.1. Encoder datum calibration

Encoder Scale	Test 1 – Timer Only	Test 2 – Timer & 12° Correction
0.9		Not conducted

Appendix H

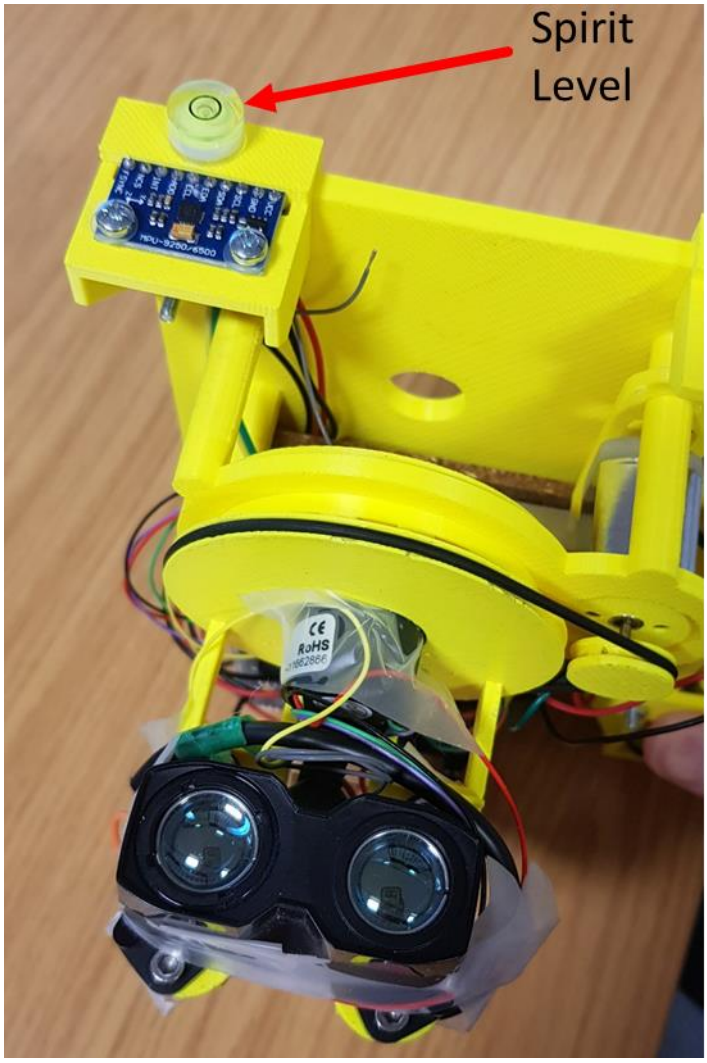
MPU-9250 Calibration & Data Capture

This Appendix has outlined the calibration and data capture methods employed on the MPU-9250 once it had been incorporated into the LiDAR prototype.

To begin, the user interface was modified to correspond to new functions that were included in the Arduino source code that made use of the MPU-9250. In total five command options were available for on the prototype; 1) "*Map Data*" recorded sensor readings from the MPU-9250 along with the distance and angle readings from the LL3HP straight onto the on-board micro-SD card, 2) "Debug *Map Data*" performed the same function as option 1 but the sensor readings were streamed over the wireless connection and displayed on screen, 3) "Stop Measurement" was used to stop any prototype actions at the end of use, 4) "*Cal Accel*" was used to calibrate the accelerometer sensors on-board the MPU9250, and 5) "Cal Mag" was used to calibrate the magnetometer sensors on board the MPU9250.

While calibration was included in the primary prototype code for the accelerometers and magnetometers it was not included for the gyroscopes. This was because the gyroscopes on the MPU-9250 were calibrated separately from the main code as they did not require regular re-calibration. The accelerometers and magnetometers, however, took readings in reference fields (i.e. gravitational and magnetic) that changed each time the prototype was used depending on location and orientation. Each time the prototype was powered it was positioned in a stable and level position, with the aid of a spirit level dot, so that calibration commands could be sent to the prototype, see Appendix Figure H.1.

When the spirit level indicated a level MPU-9250 calibration commands were sent to the prototype for acceleration and magnetic readings; gyroscope calibration occurred during hardware setup. The calibration process for accelerometers and gyroscopes involved taking a sample of 100 readings on each orthogonal sensor and determining the mean value over the sample readings. As the calibrations were conducted with a stationary prototype the mean reading over the sample window indicated the bias of the sensors which was then subtracted from future readings.



Appendix Figure H.1. Prototype in vertical orientation with MPU9250 and spirit level dot

The magnetometer was then calibrated by moving the prototype in a figure of eight pattern while 1000 iterations were conducted to determine the maximum and minimum readings in each orthogonal direction. The mean of these two values for each direction were then taken as the bias present in each magnetometer reading. The magnetometer was moved in a figure of eight pattern so that the three separate components of the Earth's magnetic field were identified separately to provide a heading reading for the prototype.

Upon completion of the MPU-9250 calibration the prototype was correctly setup to record survey data to the SD card. When data recording was requested from the prototype the LL3HP began distance measurements, the motor was activated, the optical switch enabled calculation of the LL3HP angle and the MPU-9250 started reading sensor values on all 9-axis. All of this data was recorded straight to an SD card.

In implementing this functionality, a low memory warning was received when the Arduino code was uploaded to the on-board microprocessor. In total the Arduino programme utilised 77% of its dynamic memory leaving only a small proportion of memory to define local variables during its subroutines. Although a stable functionality was achieved for the prototype such a high memory usage risks unstable behaviour of the software as more features are incorporated.

On data capture request the Arduino firmware performed the following functions;

1. Read programme run time
2. Read LL3HP Distance measurement
3. Read LL3HP angle measurement
4. Read MPU-9250 data (3-axes accelerations, 3-axes orientations and 3-axes magnetic fields)
5. Write all data onto a single line of a text file on micro-SD card
6. Repeat steps 1-5 until stop command

A sample of the data that was written to an SD card has been presented in Appendix Figure H.2 during testing. The first column was the timestamp of each data row in microseconds, the second column was the LL3HP distance measurement (cm) and the third column was the calculated LL3HP angle (°). The fourth, fifth and sixth columns provided accelerometer readings on each axis ($m\,s^{-2}$). The seventh, eighth and ninth columns provided gyroscopic measurements around each axis ($rad\,s^{-1}$). The tenth, eleventh and twelfth columns provided magnetometer readings for each axis ($\mu\mathcal{T}$).

64603348	89	235	-0.677594	-2.588264	-9.568915	-0.069741	-0.219213	0.521151	-58.258312	12.306708	-57.752265
64614980	88	15	-0.677594	-2.588264	-9.568915	-0.069741	-0.219213	0.521151	-58.258312	12.306708	-57.752265
64626556	88	154	-0.465696	-1.709547	-9.830495	-0.218614	-0.098038	0.621021	-58.258312	12.306708	-57.752265
64638144	89	294	-0.220278	-0.742240	-10.132779	-0.245646	-0.013481	0.738735	-58.258312	12.306708	-57.752265
64649796	90	112	-0.220278	-0.742240	-10.132779	-0.245646	-0.013481	0.738735	-58.258312	12.306708	-57.752265
64664600	89	253	-0.213693	0.577032	-10.621220	-0.200904	0.028864	0.786273	-60.775646	11.220822	-57.404361
64676152	74	32	-0.213693	0.577032	-10.621220	-0.200904	0.028864	0.786273	-60.775646	11.220822	-57.404361
64687656	61	170	-0.278340	0.976286	-10.848083	-0.100634	0.035255	0.661635	-60.775646	11.220822	-57.404361
64699184	46	309	-0.395063	-0.058062	-10.679882	0.013217	0.032193	0.495718	-60.775646	11.220822	-57.404361
64713884	46	126	-0.395063	-0.058062	-10.679882	0.013217	0.032193	0.495718	-60.775646	11.220822	-57.404361
64725412	52	265	-0.380099	-1.534762	-10.503898	0.025069	0.043378	0.439125	-60.775646	11.220822	-57.404361
64736944	52	44	-0.380099	-1.534762	-10.503898	0.025069	0.043378	0.439125	-60.775646	11.220822	-57.404361
64748440	59	183	-0.612947	-1.268992	-10.609848	-0.103697	0.052833	0.560833	-60.775646	11.220822	-57.404361

Appendix Figure H.2. Prototype data written to micro-SD card

On close inspection of the data output it was observed that the magnetometer produced repeated readings over several data rows compared with the timestamp, distance, angle, acceleration and orientation readings. As previously discussed the magnetometer had a

much lower update frequency compared with the other sensors which was identified as the most likely cause of this data repetition.

The presented data was recorded to the micro-SD card as individual and independent readings. The data recorded on each axis of the IMU represented the local prototype axes and did not correspond with global movements relative to a fixed Earth reference frame. For this reason, the data required post-processing to make this conversion. To make the most of the separate data streams a form of data fusion was required that enabled the gyroscopic data to use the fixed references of Earth's gravity and magnetic fields.

Appendix I

Gyroscope Data Noise Correction

The Mahony and Madgwick methods that were used for correcting noise in gyroscope data of the prototype have been described within this Appendix.

The first step of the Mahony algorithm estimated the reference directions of gravity and the Earth's magnetic field, relative to the prototype. When powered the prototype was set level allowing assumptions to be made on the initial orientation of the prototype in the Earth reference frame. From this initial position at each time step the reference directions were estimated based on the previously recorded orientation on the assumption that measurements were taken at a sufficiently high frequency that the reference directions should not substantially change between adjacent readings.

Mahony *et al.* [178] calculated the 3D vector $\boldsymbol{a_g}$ representing the Earth's gravitational direction at the sensor as per Appendix Equation I.1 where $\boldsymbol{q} = [\boldsymbol{q_0}\ \boldsymbol{q_1}\ \boldsymbol{q_2}\ \boldsymbol{q_3}]$ is the quaternion representing the last known orientation of the prototype.

$$\boldsymbol{a_g} = \begin{bmatrix} 2(\boldsymbol{q_1 q_3} - \boldsymbol{q_0 q_2}) \\ 2(\boldsymbol{q_0 q_1} + \boldsymbol{q_2 q_3}) \\ \boldsymbol{q_0}^2 - \boldsymbol{q_1}^2 - \boldsymbol{q_2}^2 + \boldsymbol{q_3}^2 \end{bmatrix}$$

Appendix Equation I.1. Estimation of direction of Earth's gravity field in Mahony algorithm

When magnetometer data was used in conjunction with accelerometer data the reference direction of the Earth's magnetic field was calculated in a similar manner. The reference direction of the magnetic field was established as the cross-product of the previous sensor orientation with the magnetometer readings at the current time step. This was then normalised via quaternion mathematics. This produced a pure quaternion $\boldsymbol{h} = [\boldsymbol{h_0}\ \boldsymbol{h_1}\ \boldsymbol{h_2}\ \boldsymbol{h_3}]$ that the defined the magnetic field direction where $\boldsymbol{h_0} = 0$. While $\boldsymbol{h_3}$ represented the magnetic field direction about a vertical axis $\boldsymbol{h_1}$ and $\boldsymbol{h_2}$ represented readings on two horizontal axes. These were combined into a single and normalised magnetic field reading $\boldsymbol{h_{12}}$ as per Appendix Equation I.2.

$$\boldsymbol{h_{12}} = \sqrt{\boldsymbol{h_1} + \boldsymbol{h_2}}$$

Appendix Equation I.2. Normalising reference horizontal magnetic fields around a single horizontal axis

This result was then subsequently used to define a new pure quaternion $\boldsymbol{b}$ as per Appendix Equation I.3.

$$\boldsymbol{b} = [0\ \boldsymbol{h_{12}}\ 0\ \boldsymbol{h_3}] = [0\ \boldsymbol{b_1}\ 0\ \boldsymbol{b_3}]$$

Appendix Equation I.3. Pure quaternion to define magnetic field reference direction

The vector direction of the resulting magnetic field $\boldsymbol{m_G}$ was then calculated via Appendix Equation I.4 where $\boldsymbol{b}$ was a pure quaternion that represented the magnetometer readings about a vertical axis and a single horizontal axis.

$$\boldsymbol{m_G} = \begin{bmatrix} 2\boldsymbol{b_1}\left(\frac{1}{2} - \boldsymbol{q_2}^2 - \boldsymbol{q_3}^2\right) + 2\boldsymbol{b_3}(\boldsymbol{q_1 q_3} - \boldsymbol{q_0 q_2}) \\ 2\boldsymbol{b_1}(\boldsymbol{q_1 q_2} - \boldsymbol{q_0 q_3}) + 2\boldsymbol{b_3}(\boldsymbol{q_0 q_1} + \boldsymbol{q_2 q_3}) \\ 2\boldsymbol{b_1}(\boldsymbol{q_0 q_2} + \boldsymbol{q_1 q_3}) + 2\boldsymbol{b_3}\left(\frac{1}{2} - \boldsymbol{q_1}^2 - \boldsymbol{q_2}^2\right) \end{bmatrix}$$

Appendix Equation I.4. Estimation of direction of Earth's magnetic field in Mahony algorithm

For both cases the cross-product between the accelerometer and magnetometer sensor readings and their respective reference direction was calculated to determine an error err_t present in the readings at each time step t. These error terms for acceleration and magnetic readings were summed and then applied to the latest gyroscope readings ω_{raw} using a preselected proportional gain term k_p to determine a corrected gyroscope reading ω_{mod} as per Appendix Equation I.5.

$$\omega_{mod} = \omega_{raw} + (k_p\ err_t)$$

Appendix Equation I.5. Gyroscope measurement correction

The rate of change of the quaternion that represented the global orientation of the prototype, based on a new and revised gyroscope reading was then calculated as per Appendix Equation I.6 where $\boldsymbol{q_{t-1}}$ was the quaternion calculated at the previous time step of the prototype and ω_{mod} was the revised gyroscope readings.

$$\dot{\boldsymbol{q}}_{\boldsymbol{t}} = \frac{1}{2}\,\omega_{mod} \otimes \boldsymbol{q_{t-1}}$$

Appendix Equation I.6. Rate of change of prototype orientation (Mahony)

The new orientation of the prototype $\boldsymbol{q_{mod}}$, in the global reference frame, was then calculated by integrating the rate of change of rotation $\dot{\boldsymbol{q}}_t$ over the average time step t of all

readings in the data set and adding to the previous orientation of the prototype $\boldsymbol{q_{t-1}}$.as per Appendix Equation I.7.

$$\boldsymbol{q_{mod}} = \boldsymbol{q_{t-1}} + \boldsymbol{\dot{q}_t}\, t$$

Appendix Equation I.7. Prototype orientation in current time step

This process was conducted over every time step in the recorded data set to add drift compensation to the recorded gyroscope results. Similarly, the technique presented by Madgwick *et al.* [119] utilised a gradient descent technique to calculate an error term from acceleration and magnetic field data that was then multiplied by a pre-defined constant β. The result of this calculation was then negated from the calculation of the rate of change of prototype orientation $\boldsymbol{\dot{q}_t}$ as a modification to Appendix Equation I.6.

The pure quaternion was defined for the reference direction of the Earth's magnetic field as per Appendix Equation I.3. Then the objective function f and corresponding transpose of the Jacobian Matrix $\boldsymbol{J}$ were defined in Madgwick's technique [119] as per Appendix Equation I.8 and Appendix Equation I.9. The components of Appendix Equation I.8 when compared with Appendix Equation I.1 and Appendix Equation I.4 were found to be similar between the two techniques.

$$f = \begin{bmatrix} 2(\boldsymbol{q_1 q_3} - \boldsymbol{q_0 q_2}) - \boldsymbol{a_{L0}} \\ 2(\boldsymbol{q_0 q_1} + \boldsymbol{q_2 q_3}) - \boldsymbol{a_{L1}} \\ 2\left(\boldsymbol{\frac{1}{2}} - \boldsymbol{{q_1}^2} - \boldsymbol{{q_2}^2}\right) - \boldsymbol{a_{L2}} \\ 2\boldsymbol{b_1}\left(\frac{1}{2} - \boldsymbol{{q_2}^2} - \boldsymbol{{q_3}^2}\right) + 2\boldsymbol{b_3}(\boldsymbol{q_1 q_3} - \boldsymbol{q_0 q_2}) - \boldsymbol{m_{L0}} \\ 2\boldsymbol{b_1}(\boldsymbol{q_1 q_2} - \boldsymbol{q_0 q_3}) + 2\boldsymbol{b_3}(\boldsymbol{q_0 q_1} + \boldsymbol{q_2 q_3}) - \boldsymbol{m_{L1}} \\ 2\boldsymbol{b_1}(\boldsymbol{q_0 q_2} + \boldsymbol{q_1 q_3}) + 2\boldsymbol{b_3}\left(\frac{1}{2} - \boldsymbol{{q_1}^2} - \boldsymbol{{q_2}^2}\right) - \boldsymbol{m_{L2}} \end{bmatrix}$$

Appendix Equation I.8. Madgwick objective function for gyroscope correction

$$\boldsymbol{J^T} = \begin{bmatrix} -2\boldsymbol{q_2} & 2\boldsymbol{q_3} & -2\boldsymbol{q_0} & 2\boldsymbol{q_1} \\ 2\boldsymbol{q_1} & 2\boldsymbol{q_0} & 2\boldsymbol{q_3} & 2\boldsymbol{q_2} \\ 0 & -4\boldsymbol{q_1} & -4\boldsymbol{q_2} & 0 \\ -2\boldsymbol{b_3 q_2} & 2\boldsymbol{b_3 q_3} & -4\boldsymbol{b_1 q_2} - 2\boldsymbol{b_3 q_0} & -4\boldsymbol{b_1 q_3} + 2\boldsymbol{b_3 q_1} \\ -2\boldsymbol{b_1 q_3} + 2\boldsymbol{b_3 q_1} & 2\boldsymbol{b_1 q_2} + 2\boldsymbol{b_3 q_0} & 2\boldsymbol{b_1 q_1} + 2\boldsymbol{b_3 q_3} & -2\boldsymbol{b_1 q_0} + 2\boldsymbol{b_3 q_2} \\ 2\boldsymbol{b_1 q_2} & 2\boldsymbol{b_1 q_3} - 4\boldsymbol{b_3 q_1} & 2\boldsymbol{b_1 q_0} - 4\boldsymbol{b_3 q_2} & 2\boldsymbol{b_1 q_1} \end{bmatrix}$$

Appendix Equation I.9. Madgwick Jacobian for gyroscope correction

Matrix multiplication and normalisation of the result between the terms f and $\boldsymbol{J}$ defined in Appendix Equation I.8 and Appendix Equation I.9 was performed to calculate the corrective step value err_t at each time point t of data collection. This corrective step was then multiplied by the pre-defined term β with the result negated from the calculation of the rate of change in sensor orientation $\dot{\boldsymbol{q}}_{\boldsymbol{t}}$, using raw sensor values, as per Appendix Equation I.10.

$$\dot{\boldsymbol{q}}_{\boldsymbol{t}} = \frac{1}{2}\,\omega_{raw} \otimes \boldsymbol{q}_{\boldsymbol{t-1}} - \beta\, err_t$$

Appendix Equation I.10. Rate of change of prototype orientation (Madgwick)

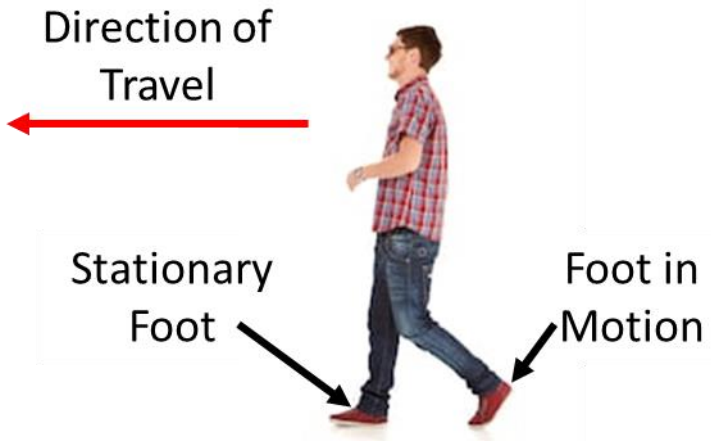
At this point the two techniques, Mahony and Madgwick, converged and the rate of change in prototype orientation $\dot{\boldsymbol{q}}_{\boldsymbol{t}}$ was integrated as per Appendix Equation I.7 to determine the corrected orientation of the prototype $\boldsymbol{q}_{\boldsymbol{mod}}$ in the Earth reference frame. While the Mahony and Madgwick techniques provided a useful method of data fusion they only enabled correction of the gyroscope orientation readings. It is common to find accelerometer readings corrected with an additional data source such as GPS. As the prototype was being developed and tested indoors the addition of GPS capabilities was not deemed practical. For this reason, an alternative method of drift correction in accelerometer data was required to accurately track the position of the prototype in 3D space. One potential method to achieve this was identified as Dead Reckoning, see Appendix J.

Appendix J

Dead Reckoning Software Test

This Appendix describes the testing of recreated software against an available dataset [175] to utilise the Dead Reckoning technique for capturing the 3D movement of the LiDAR prototype.

The Dead Reckoning method, presented by Madgwick *et al.* [119], was based on the premise that at certain points in time when IMU data was captured assumptions were made based on the accelerometer readings. In mounting an IMU to a single foot of a pedestrian a reasonable assumption was made that for every step taken, when a foot was planted, it was stationary for a short amount of time, see Appendix Figure J.1.

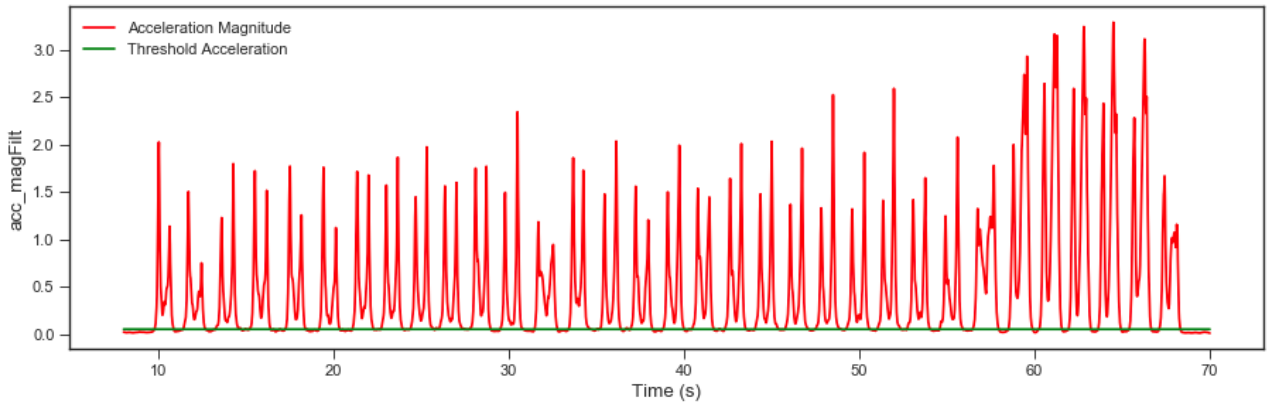


Appendix Figure J.1. Dynamics of walking at feet

When the foot was stationary the accelerations recorded by the foot-mounted IMU were expected to be low which enabled the detection of each discrete stationary time-point. This detection was conducted by calculating the Root Sum of Squares (RSS) of recorded acceleration in each orthogonal direction of the IMU for each time point during data capture. As previously described accelerometer data was found to be prone to capture signal noise in its readings and the calculation of acceleration magnitude using the RSS method still included this captured noise.

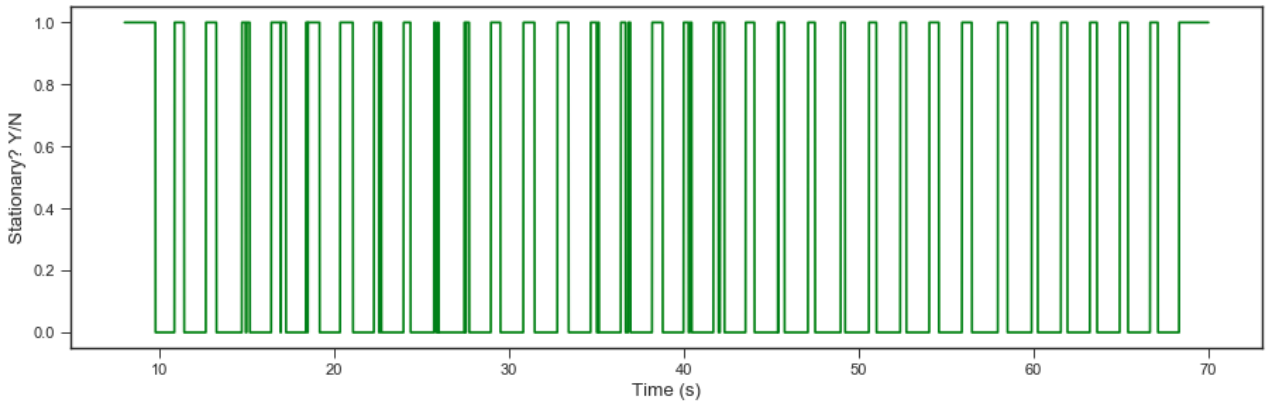
The acceleration magnitude signals were passed through a high-pass filter and a subsequent low-pass filter to attenuate extreme acceleration signal frequencies. This relied on a Butterworth filter that used cut off frequencies of 0.1% at 500% of the mean IMU sampling frequency on the dataset for the high-pass and low-pass filter respectively. Using

Madgwick's available test data [175] for a pedestrian walking down a spiral staircase, these filtered acceleration magnitudes were plotted which enabled the identification of a threshold value of acceleration at which a foot was considered stationary, see Appendix Figure J.2.



Appendix Figure J.2. Identification of stationary acceleration threshold with sample data

When the recorded acceleration fell below this threshold, which was set at $0.05m\ s^{-2}$, the IMU was assumed stationary for each subsequent time step until the acceleration rose above the threshold value, see Appendix Figure J.3.



Appendix Figure J.3. Binary identification of stationary time periods during data capture

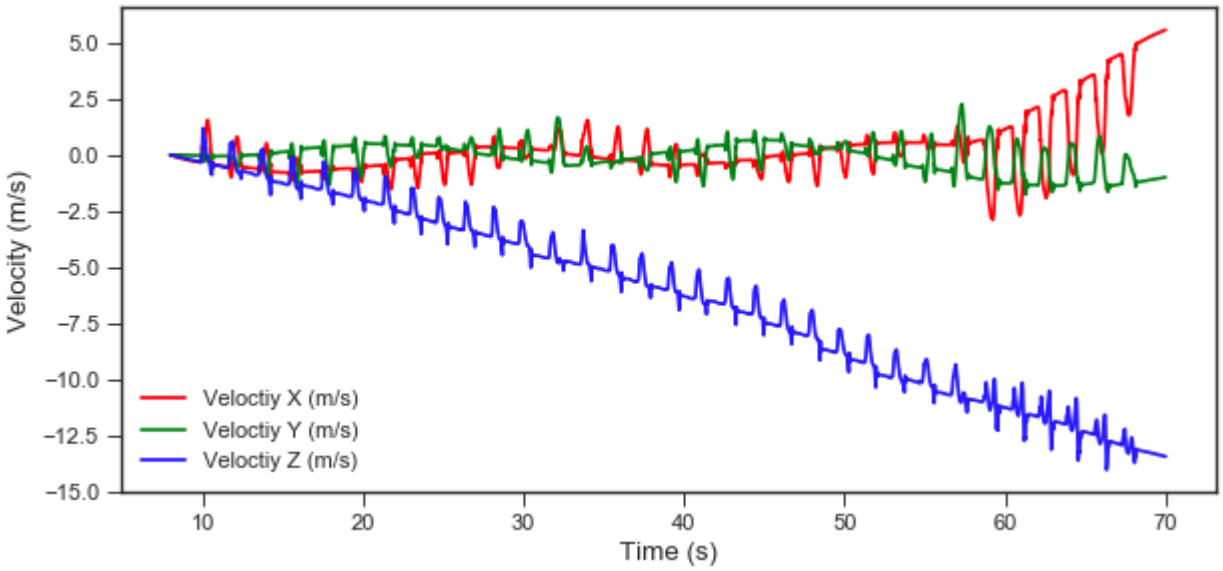
Following identification of stationary periods in each data set the IMU data was passed through one of four potential filters; 1) Mahony algorithm using gyroscope and accelerometer data, 2) Mahony algorithm using gyroscope, accelerometer and magnetometer data, 3) Madgwick algorithm using gyroscope and accelerometer data, or 4) Madgwick algorithm using gyroscope, accelerometer and magnetometer data. This refined the gyroscope data at each time point to account for signal noise through knowledge

of reference gravity and magnetic fields. The subsequent step then used knowledge of the refined orientation at each time step to convert the IMU acceleration readings into each of the Earth reference frame orthogonal axis and convert from g into $m\ s^{-2}$ units.

The acceleration data, rotated to the Earth reference frame, then had the effect of gravity in the vertical direction removed at each time point and the revised acceleration data a_t was then integrated to calculate the corresponding linear velocity v_t in each orthogonal direction at each time point t, see Appendix Equation J.1. The resulting velocity calculated over the time of data capture has been plotted in Appendix Figure J.4.

$$v_t = v_{t-1} + a_t(t_t - t_{t-1})$$

Appendix Equation J.1. Calculation of linear velocity from acceleration data



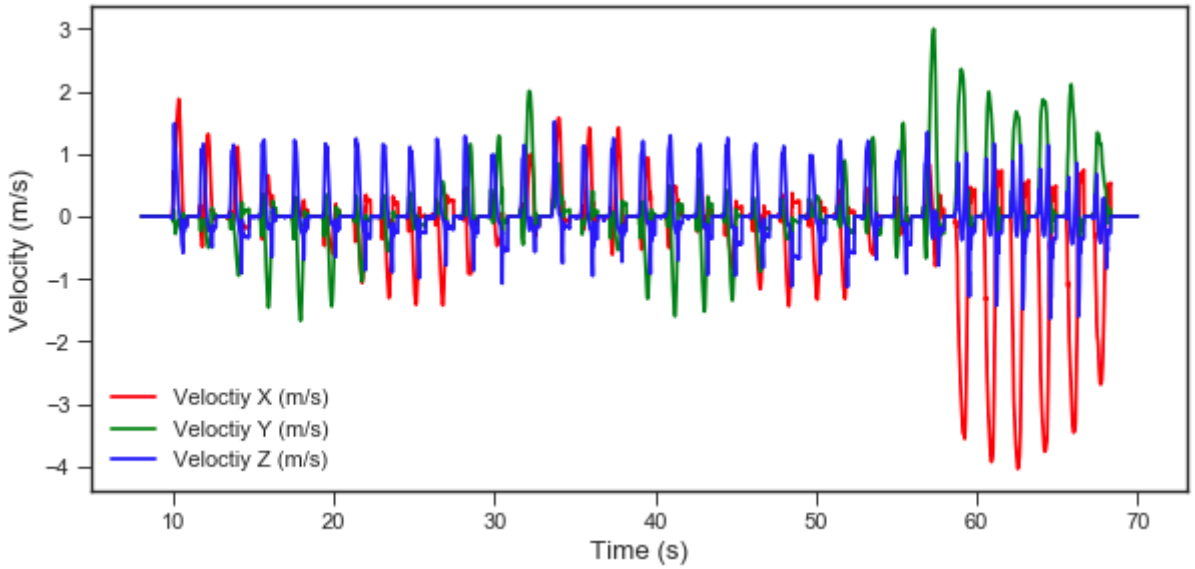
Appendix Figure J.4. Plot of orthogonal velocity by integration of raw acceleration data

Appendix Figure J.4 demonstrated an oscillating velocity in the horizontal xy plane which was expected for pedestrians walking around a spiral staircase however in the vertical direction substantial velocity drift is observed with the velocity calculations detecting an increasing velocity downwards overtime. The repetitive steps taken during data capture can be observed in each orthogonal direction which coincide with our stationary periods previously calculated.

During these stationary periods the velocity of the IMU was assumed to be $0m\ s^{-1}$ in all three orthogonal directions of the IMU. This assumption allowed for a straightforward correction to the velocity plot shown in Appendix Figure J.4 by forcing velocity data on all three

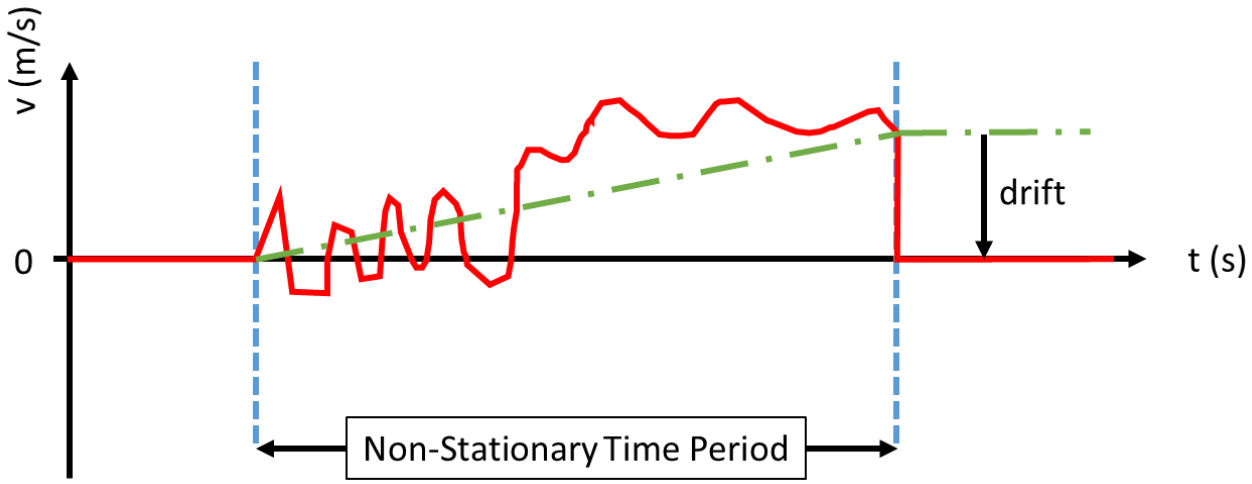
orthogonal directions to $0\,m\,s^{-1}$ when the IMU was calculated to be stationary. This corrected velocity plot is provided in Appendix Figure J.5.

The velocity correction provided in Appendix Figure J.5 removed the drift during all stationary periods such that the velocity plots in each orthogonal direction then oscillated around $0m\,s^{-1}$. Outside of the stationary periods the drift observed in Appendix Figure J.4 was still present and required an additional correction.



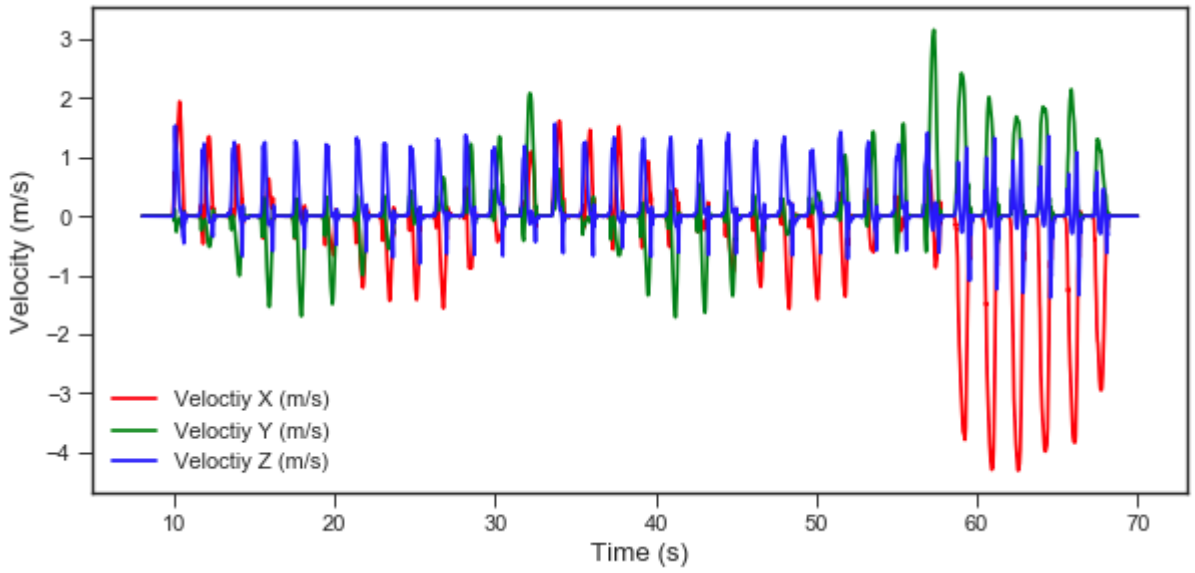
Appendix Figure J.5. Velocity plot corrected during stationary periods

To correct for the drift observed during non-stationary periods in Appendix Figure J.5, the assumed velocity when stationary of $0m\,s^{-1}$ was used to establish the drift observed in velocity calculations outside of stationary periods. An illustration of this drift phenomena has been provided in Appendix Figure J.6. At the start and end of a non-stationary time period the velocity was assumed as $0m\,s^{-1}$. From this assumption, the drift gradient over each time period was calculated as the difference between the calculated velocity immediately prior to a stationary period and $0m\,s^{-1}$. Over the course of a single non-stationary time period this gradient was used calculate a compensation at each time step that was considered non-stationary.

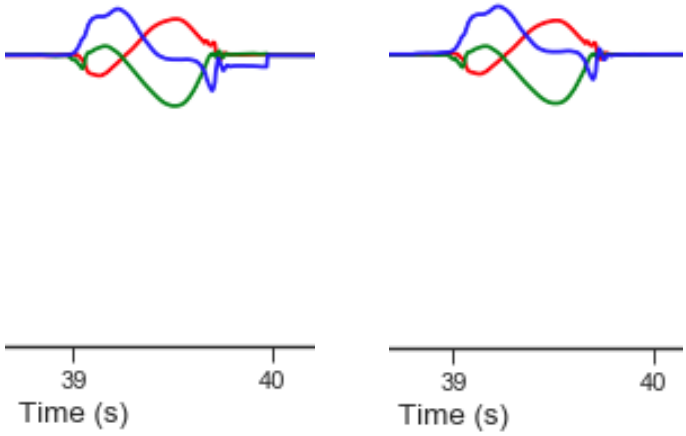


Appendix Figure J.6. Correcting drift during non-stationary periods

The drift compensation seen in Appendix Figure J.6 was negated from the previously calculated values of velocity at each time period of data capture. This resulted in the production of a velocity profile, in three orthogonal directions, over the course of data capture that had been corrected for integral drift phenomena. This drift corrected velocity profile was demonstrated in Appendix Figure J.7. While similar in appearance to Appendix Figure J.5, when zoomed into an individual step, the drift correction was observed more clearly as per the example demonstrated in Appendix Figure J.8.



Appendix Figure J.7. Drift corrected velocity profile over full data capture



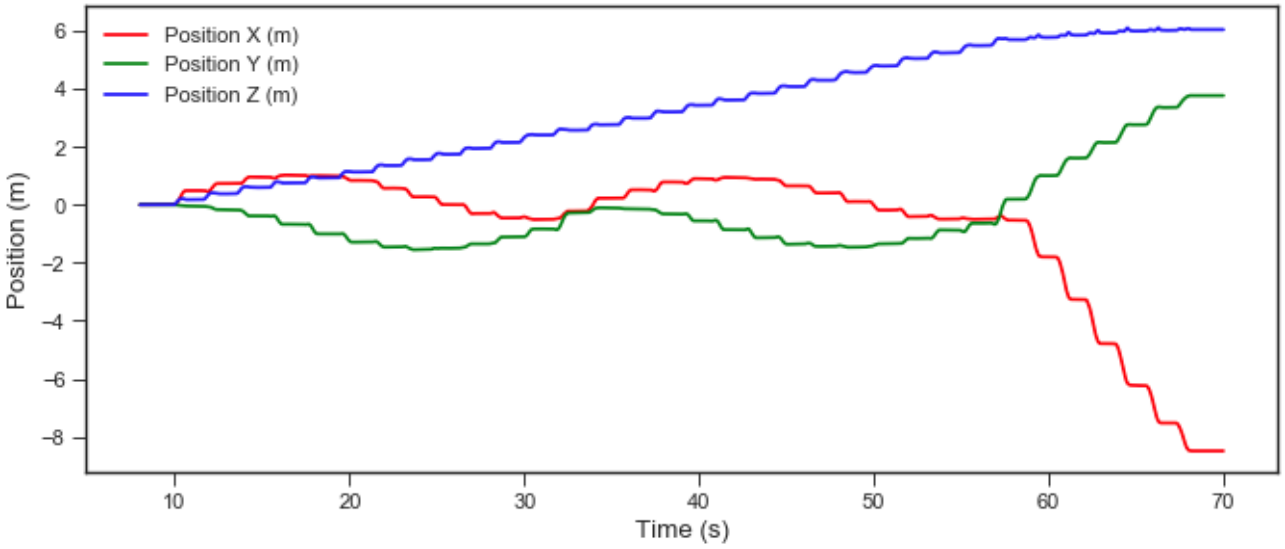
Appendix Figure J.8. (left) Stationary period set to 0m s^{-1}, (right) drift correction applied during non-stationary periods

The corrected velocity profile was then integrated one final time to calculate the 3D position of the IMU at every time step in the data set as per Appendix Equation J.2. The previous drift correction applied to the velocity prevented the appearance of similar drift phenomenon in the second integration to calculate position.

$$x_t = x_{t-1} + v_t(t_t - t_{t-1})$$

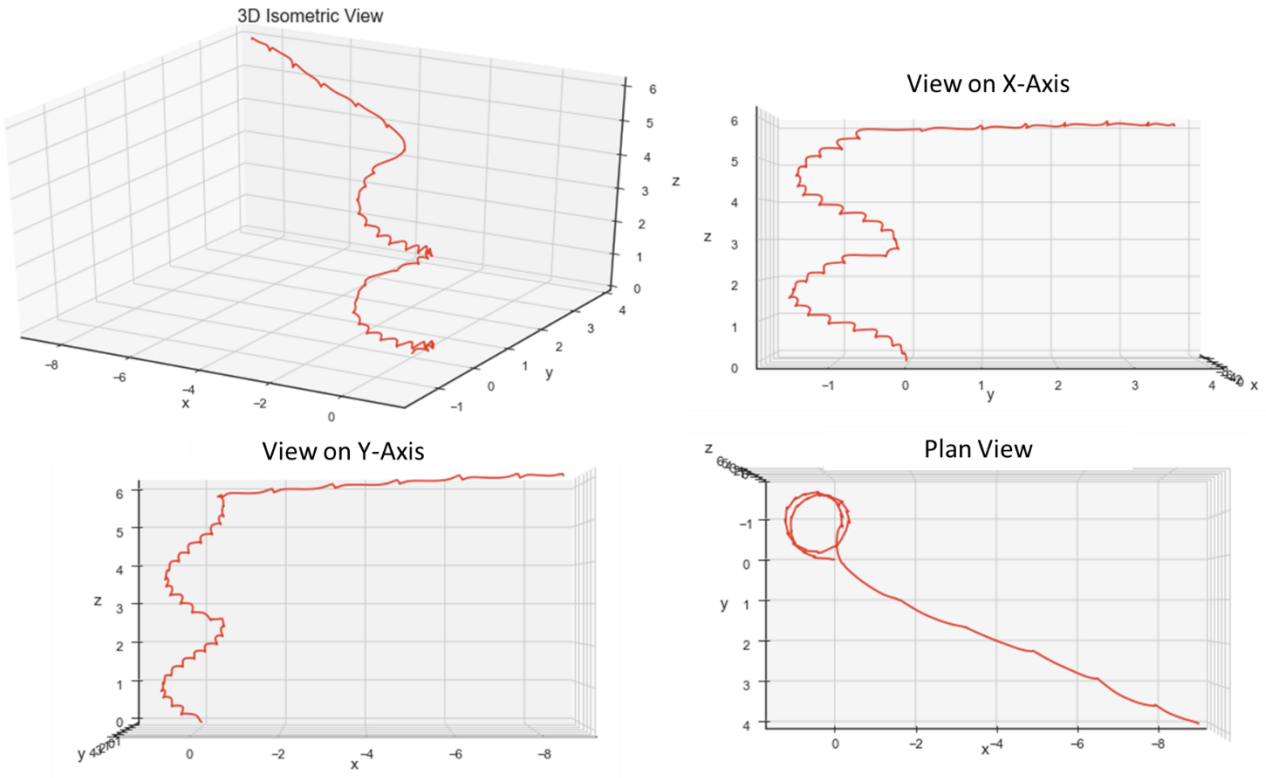
Appendix Equation J.2. Calculation of linear position from drift corrected velocity data

When applied over the full data capture time frame a 3D positional profile of the IMU over time was calculated as plotted for each orthogonal direction in Appendix Figure J.9.



Appendix Figure J.9. Positional profile of IMU over time on each orthogonal axis

Once the positional profile of the IMU had been calculated overtime in each orthogonal direction it was plotted onto a 3D coordinate system to visualise the path of the IMU over time when positioned on the foot of a pedestrian that was walking on a spiral staircase as per Appendix Figure J.10. This visualisation has captured the accurate motion of the IMU over time using only accelerometer and gyroscope data via the Mahony filter. The orientation algorithm applied a proportional gain $k_p = 0.5$ when the IMU was detected as stationary and $k_p = 0$ in all other cases. Similar results were found when the Madgwick filter was used instead of the Mahony filter with $\beta = 0.033$ across all of the time steps as recommended for fusing gyroscope and accelerometer data [119]. When both filters were applied to IMU data that included magnetic data the resulting position profile deteriorated from that demonstrated in Appendix Figure J.10. In the Madgwick and magnetic test case $\beta = 0.041$ was used as recommended [119].



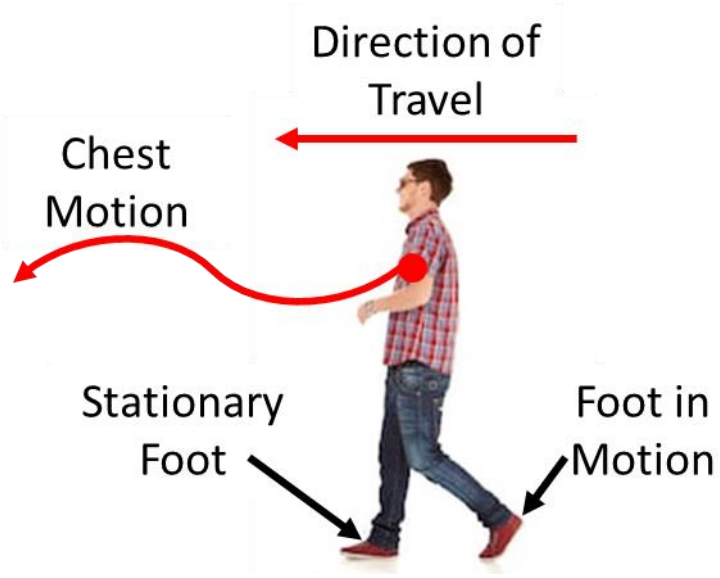
Appendix Figure J.10. 3D Position of IMU overtime on spiral staircase test data

Recreating these algorithms to display the 3D position of an IMU on sample test data provided confidence in the software, written in Python, to perform the same function on data collected from the prototype developed during this research. Two additional sets of sample data were also available to test the software which included 1) a pedestrian walking up a staircase and along a corridor, and 2) a pedestrian walking along a straight corridor. All three data sets used an IMU on a pedestrian foot and produced acceptable results in the software.

Appendix K

Prototype 3D Tracking Test Results

This Appendix describes the testing of efforts to track the position of the prototype in a 3D space. One of the key differences in the prototype data to the test data, demonstrated in Appendix J, was that the prototype was not fixed to the foot of a pedestrian but instead held at chest height close to the body of the user. During walking motion in the test data individual feet were safely assumed to be intermittently stationary which allowed dead reckoning to be applied to accelerometer data. During the same motion, the chest of a pedestrian does not become stationary in the same manner. Instead it is moved, when walking, in a smooth forward direction with oscillations in the vertical plane, see Appendix Figure K.1.

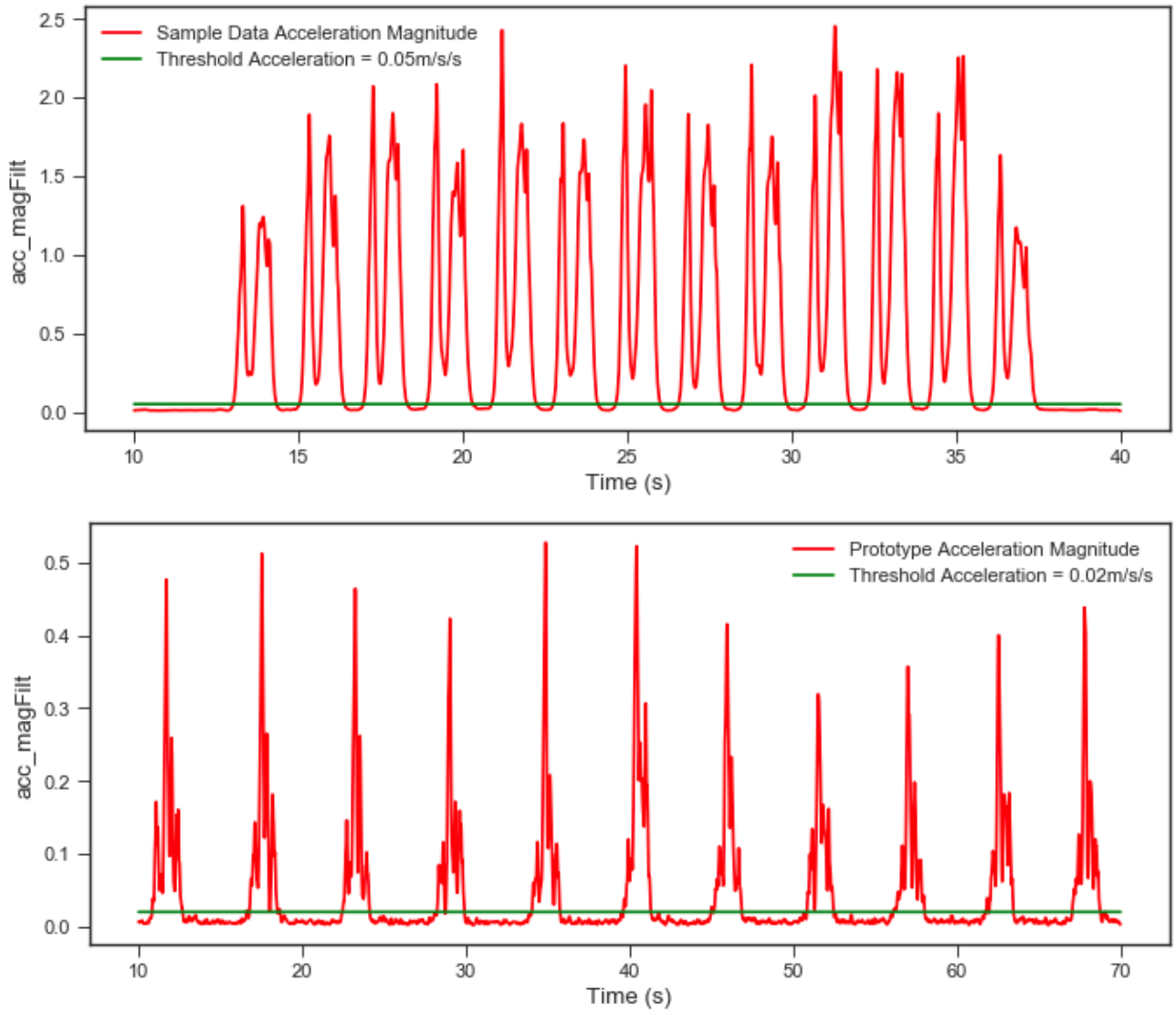


Appendix Figure K.1 Dynamics of walking at feet and chest

It was observed that this feature of the chest height gait made it more difficult to detect a stationary period where accelerometer drift phenomena can be corrected. To compensate for this the prototype data capture took place by using an intermittent walk. Once data capture began the prototype was held at chest height in an attempt to keep it as stationary as possible for a few seconds. Next a single step was taken forward and then the trailing leg was pulled forward to meet the leading leg. Then motion was momentarily paused for a few seconds before repeating the process until the end of data capture.

The test consisted of 11 of these intermittent steps in total to travel across the lab space as indicated in Figure 5.29. The magnitude of all three orthogonal accelerations that were

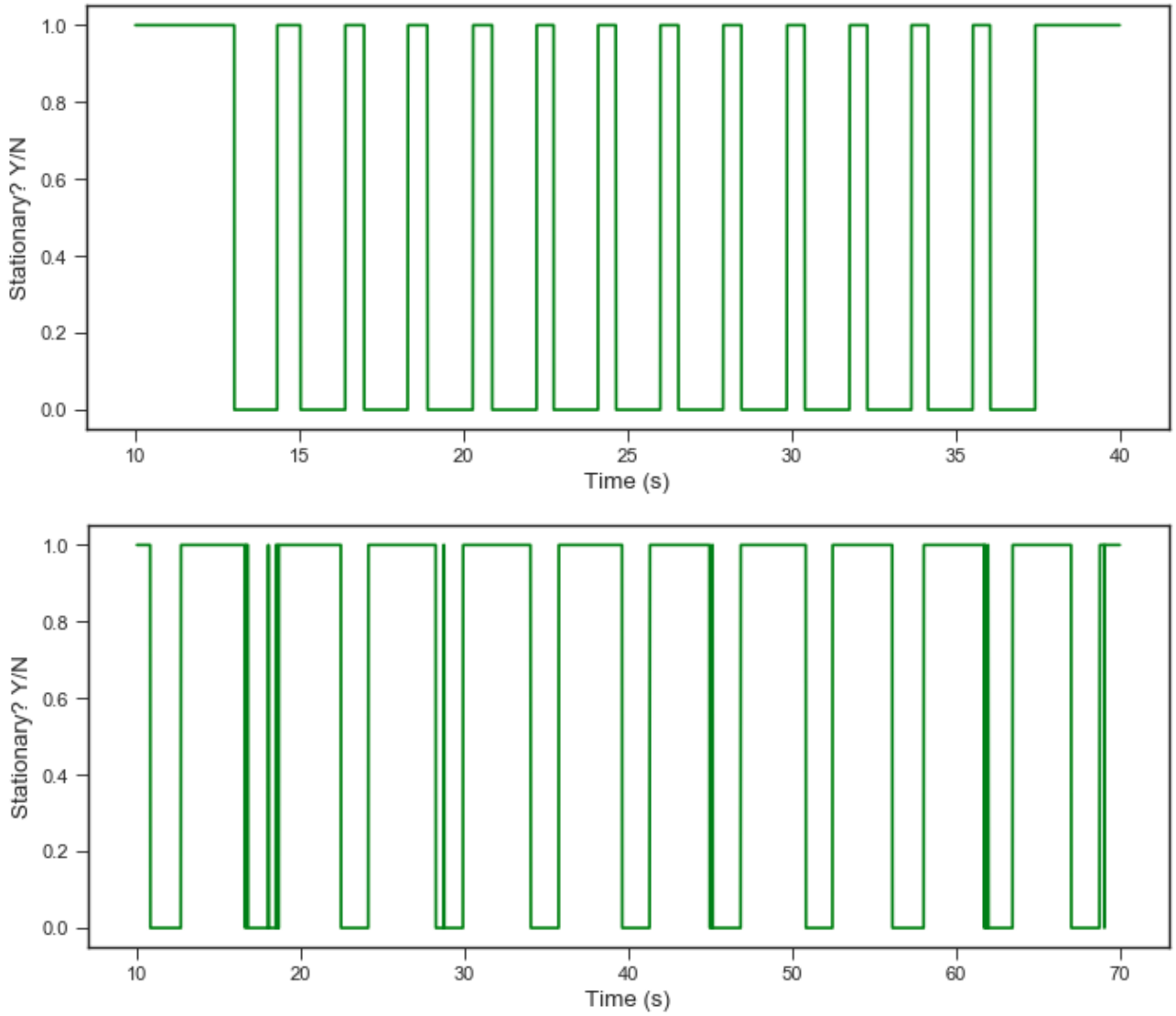
recorded during the data capture have been plotted in Appendix Figure K.2 in which a spike in acceleration is observed for each of the intermittent steps. Data captured from the prototype was compared to sample test data of a pedestrian walking down a corridor, as this data sample best suited the conditions of the prototype lab test. One observation that was made was that the acceleration values found in the sample data are much higher than the data recorded on the prototype.



Appendix Figure K.2. Magnitude of accelerations during data capture on (top) sample data, (lower) prototype data

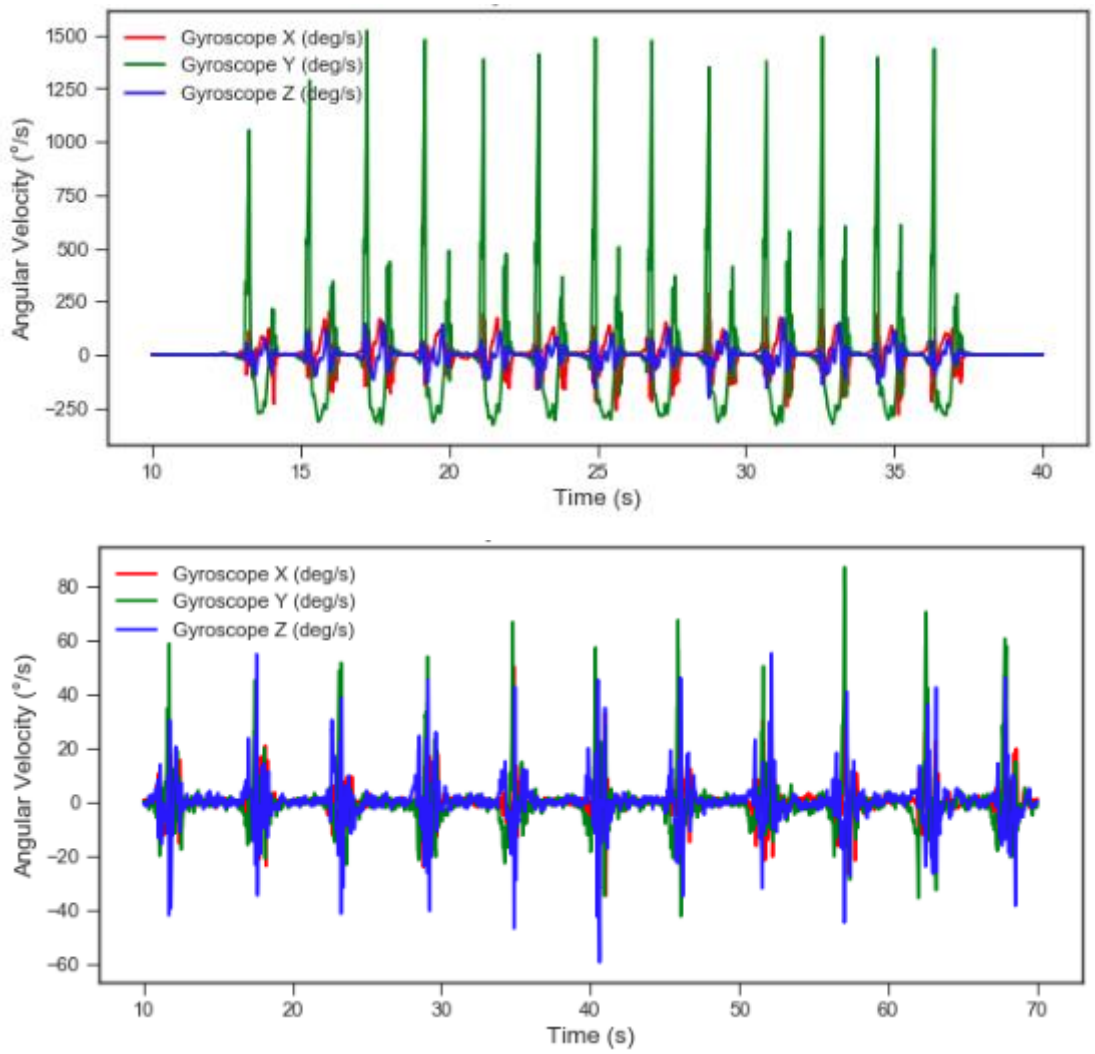
When considering the relative accelerations between the chest and feet of a pedestrian while walking it was intuited that higher accelerations would always been observed at the feet which explained this reduced acceleration level in the prototype data. The threshold used to detect a stationary period for prototype data was reduced in comparison to the

sample data tests to $0.02m\ s^{-2}$ which, for the most part, suitably bounded the stationary and non-stationary periods of the prototype test as shown in Appendix Figure K.2 and Appendix Figure K.3. The higher likelihood of motion at the chest, even when attempts were made to be stationary required this lower threshold value.



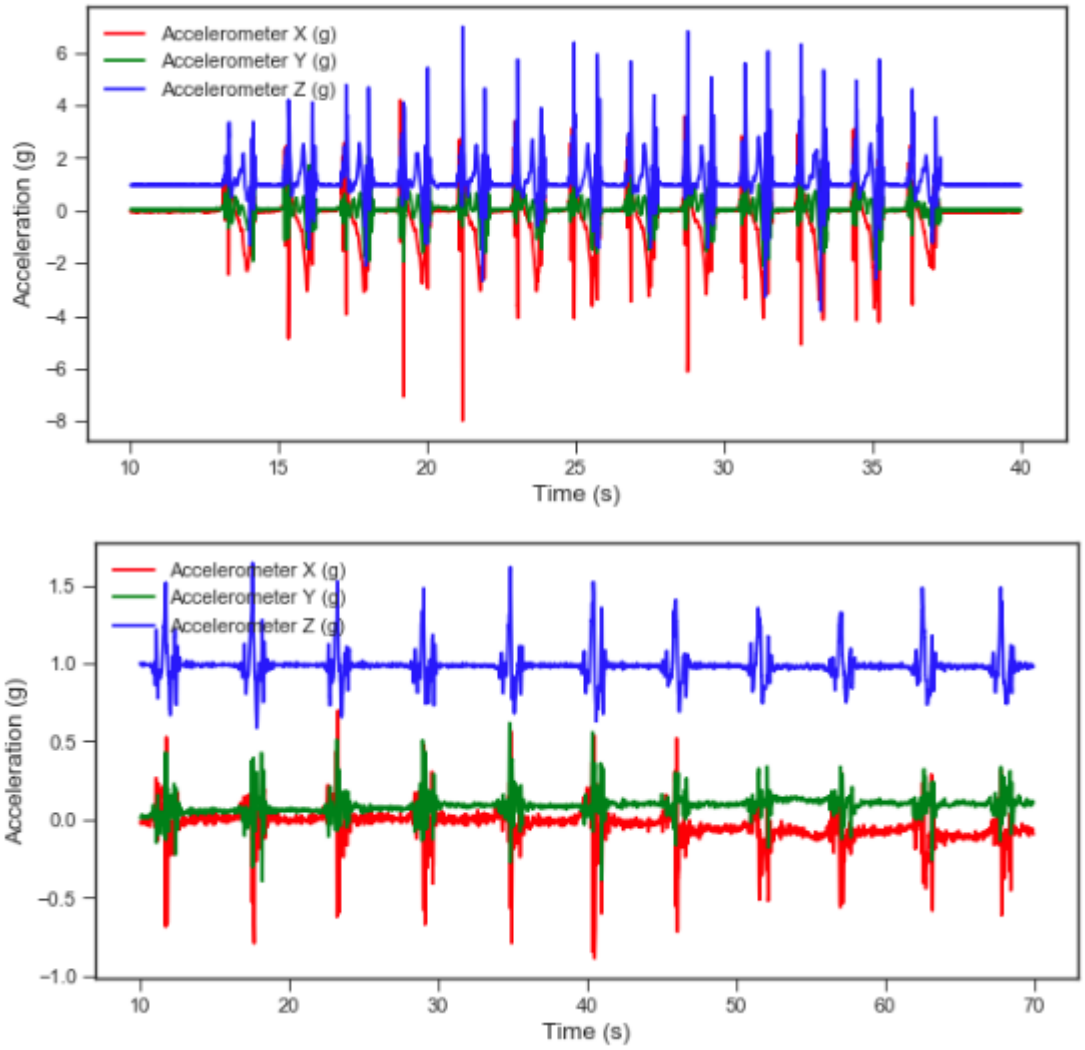
Appendix Figure K.3. Detected stationary periods during data capture on (top)sample data, (lower) prototype data

Following the identification of the discrepancy between the sample test data and prototype accelerations a comparison of raw data readings was made as displayed in Appendix Figure K.4, Appendix Figure K.5 and Appendix Figure K.6.



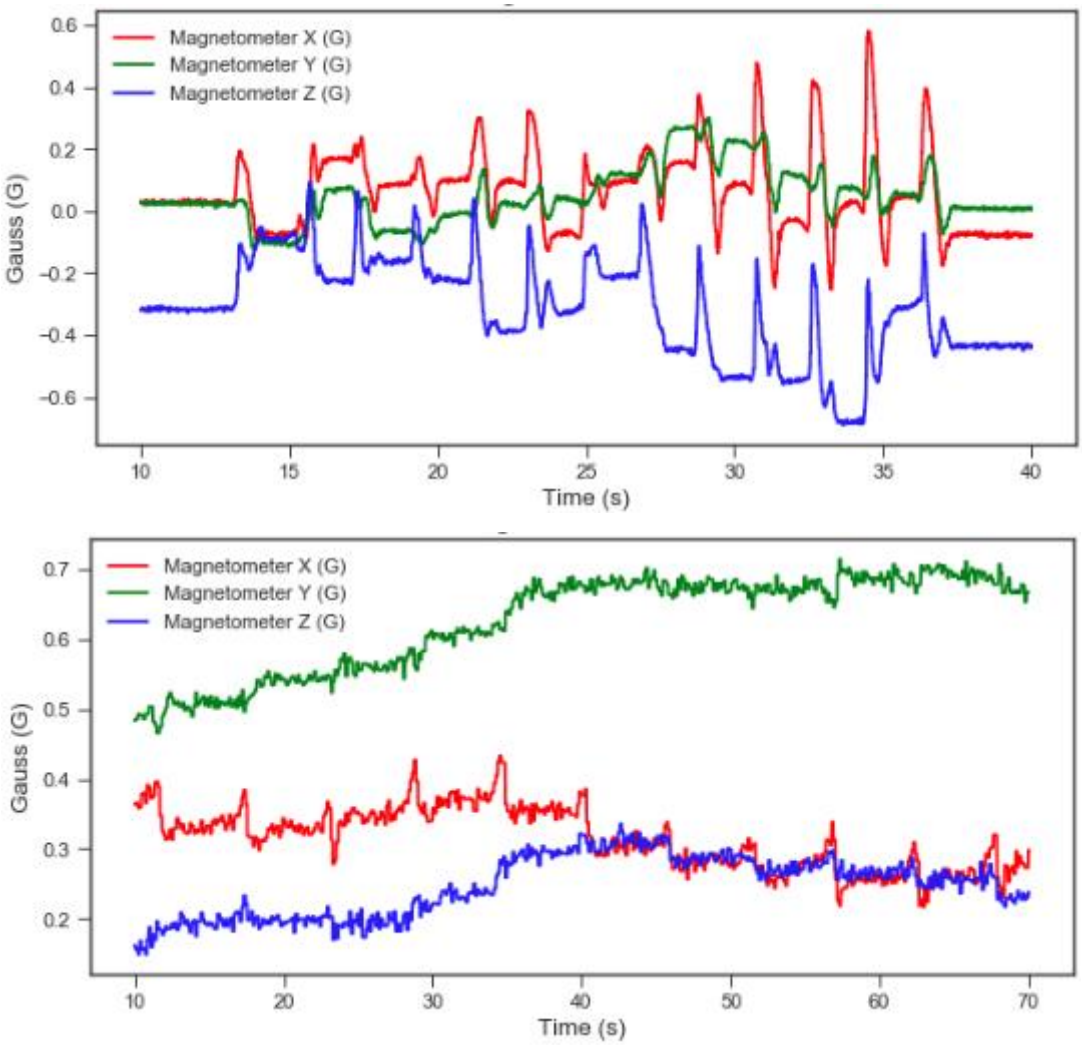
Appendix Figure K.4. (top) Sample raw gyroscope data, (lower) prototype raw gyroscope data

When the gyroscope data between the two data sets was compared in Appendix Figure K.4 a clear difference was observed in the magnitude of gyroscope readings. The sample test data recorded rates of rotation at a pedestrian foot in the range of -283 to +287° s^{-1}, -330 to +1520° s^{-1} and -204 to +169° s^{-1} on each orthogonal axis. In comparison, the recorded gyroscope data taken by the prototype at chest height was observed to be in the range of -35 to +50° s^{-1}, -43 to +87° s^{-1} and -60 to +55° s^{-1}on each orthogonal axis. When the motion of a foot and ankle in relation to the chest is considered much larger and quicker rotations were expected to be seen in the sample data which provided an explanation for some of the discrepancy. It was possible that the IMU used to capture sample data was more sensitive than the MPU-9250 implemented in the prototype which could have led to the observation of discrepancies too.



Appendix Figure K.5. (top) Sample raw accelerometer data, (lower) prototype raw accelerometer data

A similar observation was made when comparing the sample accelerometer data with that recorded by the prototype during the intermittent walking test. The accelerations observed in horizontal directions for both data sets were recorded as oscillating around $0g$, which was expected, and likewise the vertical acceleration oscillated around $1g$ in both data sets. These common data anchor points on each axis provided confidence in the values being recorded by the IMU installed on the prototype. As with the gyroscopic data sets the sample data captured much greater extreme acceleration values than compared with the prototype data. For acceleration the sample data recorded ranges, on each orthogonal axis, of-8 to $+4g$, -2 to $+3g$ and -4 to $+7g$. The equivalent ranges recorded by the accelerometers on the prototype were -0.9 to $+0.7g$, -0.4 to + $0.6g$ and +0.6 to $1.6g$. As previously described for rotational velocity, peak acceleration at a foot was also expected to be greater than the chest and this, along with the sensitivity of the sample data IMU, may have caused this discrepancy.



Appendix Figure K.6. (top) Sample raw magnetometer data, (lower) prototype raw magnetometer data

The magnetic data that was recorded between the sample data and the prototype did not display any correlation between orthogonal axes other than the fact that the orders of magnitude of the data were reasonably close compared to data captured on the gyroscopes and accelerometers. The sample data recorded magnetic ranges of -0.3 to +0.6G, -0.1 to +0.3G and -0.7 to +0.1G across each of the axis while the prototype recorded +0.22 to +0.4G, +0.5 to +0.7G and +0.1 to +0.3G on each of its axes. The magnetic field experienced at any single point around the globe was expected to be unique and as the two sets of data were captured in different locations differing magnetic fields were also expected.

Appendix L

Quaternions

Quaternions have been observed to be a powerful and complex mathematical tool that is used to describe the orientation of an object in a 3D coordinate system or the rotation of an object between two discrete orientations. They are used heavily in computer graphics and aerospace navigation applications to describe orientations as, unlike Euler angles, they do not experience Gimbal Lock where two degrees of freedom become aligned. The general form of a quaternion has been provided in Appendix Equation L.1.

$$\boldsymbol{q} = \boldsymbol{q_0} + \boldsymbol{q_1 i} + \boldsymbol{q_2 j} + \boldsymbol{q_3 k}$$

Appendix Equation L.1. General form of a quaternion

Quaternions were developed by Sir William Rowan Hamilton who first established the quaternion formula to enable quaternion multiplication as per Appendix Equation L.2.

$$\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{i\,j\,k} = -1$$

Thus;

$$\boldsymbol{i\,j} = \boldsymbol{k}$$

$$\boldsymbol{j\,k} = \boldsymbol{i}$$

$$\boldsymbol{k\,i} = \boldsymbol{j}$$

$$\boldsymbol{j\,i} = -\boldsymbol{k}$$

$$\boldsymbol{k\,j} = -\boldsymbol{i}$$

$$\boldsymbol{i\,k} = -\boldsymbol{j}$$

Appendix Equation L.2. Fundamental expressions of quaternion multiplication

This mathematical development allowed the imaginary numbers $\boldsymbol{i}$, $\boldsymbol{j}$ and $\boldsymbol{k}$ to represent the Cartesian vectors in a three-dimensional system to represent an object orientation. As with complex number notation quaternions can be represented by an ordered pair, see Appendix Equation L.3 where $\boldsymbol{q_0}$ represented the "real" scalar part of the quaternion and $\boldsymbol{v}$ represented the "imaginary" vector part of the quaternion.

$$\boldsymbol{q} = [\boldsymbol{q_0}, \boldsymbol{v}]$$

Appendix Equation L.3. Quaternion as an ordered pair

Some rules have been developed to perform mathematical operations between two quaternions such as addition, see Appendix Equation L.4, and products, see Appendix Equation L.5. Quaternions are expressed as "real" quaternions when the vector term is 0, and vice-versa they are expressed as "pure" quaternions when the scalar term is 0.

$$\boldsymbol{q_A} + \boldsymbol{q_B} = [(\boldsymbol{q_{0,A}} + \boldsymbol{q_{0,B}}), (\boldsymbol{v_A} + \boldsymbol{v_B})]$$

$$\boldsymbol{q_A} - \boldsymbol{q_B} = [(\boldsymbol{q_{0,A}} - \boldsymbol{q_{0,B}}), (\boldsymbol{v_A} - \boldsymbol{v_B})]$$

Appendix Equation L.4. Addition and subtraction of two quaternions

$$\boldsymbol{q_A}\,\boldsymbol{q_B} = [(\boldsymbol{q_{0,A}}\,\boldsymbol{q_{0,B}} - \boldsymbol{v_A} \cdot \boldsymbol{v_B}), (\boldsymbol{q_{0,A}}\,\boldsymbol{v_B} + \boldsymbol{q_{0,B}}\,\boldsymbol{v_A} + \boldsymbol{v_A} \times \boldsymbol{v_B})]$$

Appendix Equation L.5. Product of two quaternions

Four quaternion manipulations have been used when determining the angle between two quaternions, or the rotation from one quaternion to a second. These manipulations are the, magnitude, conjugate, inverse and normalisation as presented in Appendix Equation L.6, Appendix Equation L.7, Appendix Equation L.8 and Appendix Equation L.9 respectively.

$$|\boldsymbol{q}| = \sqrt{\boldsymbol{q_0}^2 + |\boldsymbol{v}|^2}$$

Appendix Equation L.6. Quaternion magnitude

$$\boldsymbol{q}^* = [\boldsymbol{q_0}, -\boldsymbol{v}]$$

$$\boldsymbol{q}\,\boldsymbol{q}^* = [\boldsymbol{q_0}^2 + |\boldsymbol{v}|^2, 0] = |\boldsymbol{q}|^2$$

Appendix Equation L.7. Quaternion conjugate and product

$$\boldsymbol{q}^{-1} = \frac{\boldsymbol{q}^*}{|\boldsymbol{q}|^2}$$

$$\boldsymbol{q}\,\boldsymbol{q}^{-1} = [1, 0] = 1$$

Appendix Equation L.8. Quaternion inverse and product

$$\boldsymbol{q}' = \frac{\boldsymbol{q}}{|\boldsymbol{q}|}$$

Appendix Equation L.9. Quaternion normalisation

The dot product of two quaternions has enabled the calculation of the angle between two quaternions. The dot product has been defined in Appendix Equation L.10.

$$\boldsymbol{q_A \cdot q_B = q_{0,A}\, q_{0,B} + q_{1,A}\, q_{1,B} + q_{2,A}\, q_{2,B} + q_{3,A}\, q_{3,B}}$$

Appendix Equation L.10. Quaternion dot product

Subsequently the cosine of the angle between two quaternions has been defined by Appendix Equation L.11.

$$\cos\theta = \frac{\boldsymbol{q_A \cdot q_B}}{|\boldsymbol{q_A}||\boldsymbol{q_B}|} = \frac{\boldsymbol{q_{0,A}\, q_{0,B} + q_{1,A}\, q_{1,B} + q_{2,A}\, q_{2,B} + q_{3,A}\, q_{3,B}}}{|\boldsymbol{q_A}||\boldsymbol{q_B}|}$$

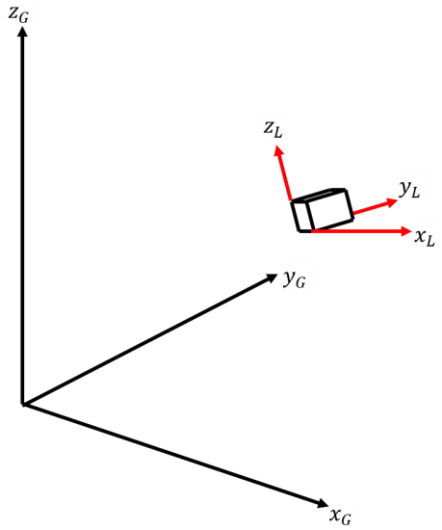
Appendix Equation L.11. Cosine of the angle between two quaternions

Given a vector $\boldsymbol{v}$ that represented the initial orientation of an object in a 3D coordinate system, its rotation described by a quaternion $\boldsymbol{q}$ determined the new orientation $\boldsymbol{v'}$ as defined in Appendix Equation L.12. Following the first multiplication the result was multiplied by the inverse quaternion $\boldsymbol{q^{-1}}$ to ensure the magnitude of the resulting vector $\boldsymbol{v'}$ remained unchanged through the transformation.

$$\boldsymbol{v' = q\, v\, q^{-1}}$$

Appendix Equation L.12. Calculate new orientation from quaternion rotation

In addition to calculating a new orientation based on a quaternion, the same mathematics has been used to estimate the direction vector of reference fields by consideration of a quaternion measured by a sensor. While a gyroscope measured the rate of rotation of an object in a local three-dimensional axes knowledge of the sensors orientation relative to an Earth based three-dimensional axes has of benefit to determine the final global orientation, see Appendix Figure L.1.



Appendix Figure L.1. Orientation of object, with local coordinate system, in global coordinate system

The sensor readings from a gyroscope have been used to estimate the direction of gravity relative to the local sensor coordinates. The direction of gravity in the global Earth-frame has been estimated by the normalised gravity vector $\boldsymbol{g}_G$ as per Appendix Equation L.13

$$\boldsymbol{g}_G = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Appendix Equation L.13. Estimate of global gravity vector

Likewise, the orientation of an object in that global coordinate system has been defined by the unit quaternion $\boldsymbol{q}_G$ in Appendix Equation L.14 where $|\boldsymbol{q}_G| = 1$.

$$\boldsymbol{q}_G = \boldsymbol{q}_{G,0} + \boldsymbol{q}_{G,1}\boldsymbol{i} + \boldsymbol{q}_{G,2}\boldsymbol{j} + \boldsymbol{q}_{G,3}\boldsymbol{k}$$

Appendix Equation L.14. Object orientation

The direction of gravity at the sensor was calculated by first rotating the known gravity vector $\boldsymbol{g}_G$, expressed as a pure quaternion $[0,\ \boldsymbol{g}_G]$, by the quaternion that represented the object orientation $\boldsymbol{q}_G$, expressed as an ordered pair $[\boldsymbol{q}_{G,0}, \boldsymbol{v}_G]$ where $\boldsymbol{v}_G = \boldsymbol{q}_{G,1}\boldsymbol{i} + \boldsymbol{q}_{G,2}\boldsymbol{j} + \boldsymbol{q}_{G,3}\boldsymbol{k}$. This provided the direction of $\boldsymbol{z}_L$, expressed as a pure quaternion $[0,\ \boldsymbol{z}_L]$, at the sensor, relative to the Earth frame, and the direction of gravity was then calculated by calculating the inverse quaternion ${\boldsymbol{z}_L}^{-1}$ which provided the direction of gravity relative to the sensor inertial frame $\boldsymbol{g}_L$ which has been derived below in Appendix Equation L.15.

$$\boldsymbol{z}_L = \boldsymbol{q}_G\, \boldsymbol{g}_G\, {\boldsymbol{q}_G}^{-1}$$

$$[0, \boldsymbol{z}_L] = [\boldsymbol{q}_{G,0}, \boldsymbol{v}_G][0, g_G]D[\boldsymbol{q}_{G,0}, -\boldsymbol{v}_G], where\ D = |\boldsymbol{q}_{G,0}|^{-2}$$

$$as\ |\boldsymbol{q}_G| = 1\ then\ D = 1.$$

$$[0, \boldsymbol{z}_L] = \Big[[\boldsymbol{q}_{G,0}, \boldsymbol{v}_G][0, g_G]\Big][\boldsymbol{q}_{G,0}, -\boldsymbol{v}_G]$$

$$[0, \boldsymbol{z}_L] = \Big[\big[(\boldsymbol{q}_{G,0} \times 0) - (\boldsymbol{v}_G \cdot g_G)\big], (\boldsymbol{q}_{G,0} g_G) + 0\boldsymbol{v}_G + (\boldsymbol{v}_G \times g_G)\Big][\boldsymbol{q}_{G,0}, -\boldsymbol{v}_G]$$

$$[0, \boldsymbol{z}_L] = \big[-(\boldsymbol{v}_G \cdot g_G), (\boldsymbol{q}_{G,0} g_G) + (\boldsymbol{v}_G \times g_G)\big][\boldsymbol{q}_{G,0}, -\boldsymbol{v}_G]$$

$$[0, \boldsymbol{z}_L] = \Big[-(\boldsymbol{v}_G \cdot g_G) - \Big(\big(\boldsymbol{q}_{G,0} g_G + (\boldsymbol{v}_G \times g_G)\big) \cdot (-\boldsymbol{v}_G)\Big), -(\boldsymbol{v}_G \cdot g_G)(-\boldsymbol{v}_G) + \boldsymbol{q}_{G,0}\big((\boldsymbol{q}_{G,0} g_G) + (\boldsymbol{v}_G \times g_G)\big) + \big(\boldsymbol{q}_{G,0} g_G + (\boldsymbol{v}_G \times g_G)\big) \times (-\boldsymbol{v}_G)\Big]$$

The real part of the quaternion was reduced to zero.

$$-(\boldsymbol{v}_G \cdot g_G) - \Big(\big(\boldsymbol{q}_{G,0} g_G + (\boldsymbol{v}_G \times g_G)\big) \cdot (-\boldsymbol{v}_G)\Big)$$

$$-\big(\boldsymbol{v}_G \cdot g_G \boldsymbol{q}_{G,0}\big) - \big(\boldsymbol{q}_{G,0} g_G \cdot (-\boldsymbol{v}_G) + (\boldsymbol{v}_G \times g_G) \cdot (-\boldsymbol{v}_G)\big)$$

$$-\boldsymbol{v}_G \cdot g_G \boldsymbol{q}_{G,0} + \boldsymbol{q}_{G,0} g_G \cdot \boldsymbol{v}_G - (\boldsymbol{v}_G \times g_G) \cdot (-\boldsymbol{v}_G)$$

$$0 - (\boldsymbol{v}_G \times g_G) \cdot (-\boldsymbol{v}_G)$$

$$0 - -g_G \cdot (\boldsymbol{v}_G \times \boldsymbol{v}_G)$$

$$0$$

Thus,

$$[0, \boldsymbol{z}_L] = \Big[0, -(\boldsymbol{v}_G \cdot g_G)(-\boldsymbol{v}_G) + \boldsymbol{q}_{G,0}\big((\boldsymbol{q}_{G,0} g_G) + (\boldsymbol{v}_G \times g_G)\big) + \big(\boldsymbol{q}_{G,0} g_G + (\boldsymbol{v}_G \times g_G)\big) \times (-\boldsymbol{v}_G)\Big]$$

$$[0, \boldsymbol{z}_L] = \Big[0, \boldsymbol{v}_G(\boldsymbol{v}_G \cdot g_G) + \boldsymbol{q}_{G,0}{}^2 g_G + \boldsymbol{q}_{G,0}\boldsymbol{v}_G \times g_G + \big(\boldsymbol{q}_{G,0} g_G + (\boldsymbol{v}_G \times g_G)\big) \times (-\boldsymbol{v}_G)\Big]$$

Then substituting back in $\boldsymbol{v}_G = \boldsymbol{q}_{G,1}\boldsymbol{i} + \boldsymbol{q}_{G,2}\boldsymbol{j} + \boldsymbol{q}_{G,3}\boldsymbol{k}$ and $\boldsymbol{g}_G = \boldsymbol{k}$,

$$[0, \boldsymbol{z}_L] = \Big[0, \boldsymbol{v}_G(\boldsymbol{v}_G \cdot g_G) + \boldsymbol{q}_{G,0}{}^2 g_G + \boldsymbol{q}_{G,0}\boldsymbol{v}_G \times g_G + \boldsymbol{v}_G \times \big(\boldsymbol{q}_{G,0} g_G + (\boldsymbol{v}_G \times g_G)\big)\Big]$$

$$[0, \boldsymbol{z}_L] = \left[0, \begin{bmatrix}\boldsymbol{q}_{G,1}\\ \boldsymbol{q}_{G,2}\\ \boldsymbol{q}_{G,3}\end{bmatrix}\left(\begin{bmatrix}\boldsymbol{q}_{G,1}\\ \boldsymbol{q}_{G,2}\\ \boldsymbol{q}_{G,3}\end{bmatrix}\cdot\begin{bmatrix}0\\0\\1\end{bmatrix}\right) + {\boldsymbol{q}_{G,0}}^2\begin{bmatrix}0\\0\\1\end{bmatrix} + \boldsymbol{q}_{G,0}\begin{bmatrix}\boldsymbol{q}_{G,1}\\ \boldsymbol{q}_{G,2}\\ \boldsymbol{q}_{G,3}\end{bmatrix}\times\begin{bmatrix}0\\0\\1\end{bmatrix} + \begin{bmatrix}\boldsymbol{q}_{G,1}\\ \boldsymbol{q}_{G,2}\\ \boldsymbol{q}_{G,3}\end{bmatrix}\times\left(\boldsymbol{q}_{G,0}\begin{bmatrix}0\\0\\1\end{bmatrix} + \left(\begin{bmatrix}\boldsymbol{q}_{G,1}\\ \boldsymbol{q}_{G,2}\\ \boldsymbol{q}_{G,3}\end{bmatrix}\times\begin{bmatrix}0\\0\\1\end{bmatrix}\right)\right)\right]$$

$$[0, \boldsymbol{z}_L] = \left[0, \begin{bmatrix}\boldsymbol{q}_{G,1}\\ \boldsymbol{q}_{G,2}\\ \boldsymbol{q}_{G,3}\end{bmatrix}\boldsymbol{q}_{G,3} + \begin{bmatrix}0\\0\\{\boldsymbol{q}_{G,0}}^2\end{bmatrix} + \boldsymbol{q}_{G,0}\begin{bmatrix}\boldsymbol{q}_{G,2}\\ -\boldsymbol{q}_{G,1}\\ 0\end{bmatrix} + \begin{bmatrix}\boldsymbol{q}_{G,1}\\ \boldsymbol{q}_{G,2}\\ \boldsymbol{q}_{G,3}\end{bmatrix}\times\left(\begin{bmatrix}0\\0\\\boldsymbol{q}_{G,0}\end{bmatrix} + \begin{bmatrix}\boldsymbol{q}_{G,2}\\ -\boldsymbol{q}_{G,1}\\ 0\end{bmatrix}\right)\right]$$

$$[0, \boldsymbol{z}_L] = \left[0, \begin{bmatrix}\boldsymbol{q}_{G,1}\boldsymbol{q}_{G,3}\\ \boldsymbol{q}_{G,2}\boldsymbol{q}_{G,3}\\ {\boldsymbol{q}_{G,3}}^2\end{bmatrix} + \begin{bmatrix}0\\0\\{\boldsymbol{q}_{G,0}}^2\end{bmatrix} + \begin{bmatrix}\boldsymbol{q}_{G,0}\boldsymbol{q}_{G,2}\\ -\boldsymbol{q}_{G,0}\boldsymbol{q}_{G,1}\\ 0\end{bmatrix} + \begin{bmatrix}\boldsymbol{q}_{G,1}\\ \boldsymbol{q}_{G,2}\\ \boldsymbol{q}_{G,3}\end{bmatrix}\times\begin{bmatrix}\boldsymbol{q}_{G,2}\\ -\boldsymbol{q}_{G,1}\\ \boldsymbol{q}_{G,0}\end{bmatrix}\right]$$

$$[0, \boldsymbol{z}_L] = \left[0, \begin{bmatrix}\boldsymbol{q}_{G,1}\boldsymbol{q}_{G,3}\\ \boldsymbol{q}_{G,2}\boldsymbol{q}_{G,3}\\ {\boldsymbol{q}_{G,3}}^2\end{bmatrix} + \begin{bmatrix}\boldsymbol{q}_{G,0}\boldsymbol{q}_{G,2}\\ -\boldsymbol{q}_{G,0}\boldsymbol{q}_{G,1}\\ {\boldsymbol{q}_{G,0}}^2\end{bmatrix} + \begin{bmatrix}\boldsymbol{q}_{G,2}\boldsymbol{q}_{G,0} + \boldsymbol{q}_{G,1}\boldsymbol{q}_{G,3}\\ \boldsymbol{q}_{G,2}\boldsymbol{q}_{G,3} - \boldsymbol{q}_{G,0}\boldsymbol{q}_{G,1}\\ -{\boldsymbol{q}_{G,1}}^2 - {\boldsymbol{q}_{G,2}}^2\end{bmatrix}\right]$$

$$[0, \boldsymbol{z}_L] = \left[0, \begin{bmatrix}2\left(\boldsymbol{q}_{G,2}\boldsymbol{q}_{G,0} + \boldsymbol{q}_{G,1}\boldsymbol{q}_{G,3}\right)\\ 2\left(\boldsymbol{q}_{G,2}\boldsymbol{q}_{G,3} - \boldsymbol{q}_{G,0}\boldsymbol{q}_{G,1}\right)\\ {\boldsymbol{q}_{G,0}}^2 - {\boldsymbol{q}_{G,1}}^2 - {\boldsymbol{q}_{G,2}}^2 + {\boldsymbol{q}_{G,3}}^2\end{bmatrix}\right]$$

This provided the direction of $\boldsymbol{z}_L$ which was relative to the global Earth frame. The direction of gravity relative to the local inertial frame is thus $\boldsymbol{g}_L = {\boldsymbol{z}_L}^{-1}$

$$[0, \boldsymbol{g}_L] = [0, \boldsymbol{z}_L]^{-1} = \left[0, \begin{bmatrix}2\left(\boldsymbol{q}_{G,2}\boldsymbol{q}_{G,0} - \boldsymbol{q}_{G,1}\boldsymbol{q}_{G,3}\right)\\ 2\left(\boldsymbol{q}_{G,2}\boldsymbol{q}_{G,3} + \boldsymbol{q}_{G,0}\boldsymbol{q}_{G,1}\right)\\ {\boldsymbol{q}_{G,0}}^2 - {\boldsymbol{q}_{G,1}}^2 - {\boldsymbol{q}_{G,2}}^2 + {\boldsymbol{q}_{G,3}}^2\end{bmatrix}\right]$$

$$\boldsymbol{g}_L = \begin{bmatrix}2\left(\boldsymbol{q}_{G,1}\boldsymbol{q}_{G,3} - \boldsymbol{q}_{G,0}\boldsymbol{q}_{G,2}\right)\\ 2\left(\boldsymbol{q}_{G,0}\,\boldsymbol{q}_{G,1} + \boldsymbol{q}_{G,2}\boldsymbol{q}_{G,3}\right)\\ {\boldsymbol{q}_{G,0}}^2 - {\boldsymbol{q}_{G,1}}^2 - {\boldsymbol{q}_{G,2}}^2 + {\boldsymbol{q}_{G,3}}^2\end{bmatrix}$$

Appendix Equation L.15. Direction of gravity relative to sensor

In exactly the same manner the direction of the Earth's magnetic field $\boldsymbol{b}_L$, expressed relative to the sensor, was calculated by taking its orthogonal components such that $\boldsymbol{b}_L = \begin{bmatrix}0 & \boldsymbol{b}_{L,1} & 0 & \boldsymbol{b}_{L,3}\end{bmatrix}$ and substituting this vector into the same set of equations as Appendix

Equation L.15, instead of the gravity vector. This results in a similar vector for the direction of the Earth's magnetic field in the global frame $\boldsymbol{b}_L$ as shown in Appendix Equation L.16.

$$\boldsymbol{b}_G = \begin{bmatrix} 2\boldsymbol{b}_{L,1}\left(\frac{1}{2} - \boldsymbol{q}_{G,2}{}^2 - \boldsymbol{q}_{G,3}{}^2\right) + 2\boldsymbol{b}_{L,3}\left(\boldsymbol{q}_{G,1}\boldsymbol{q}_{G,3} - \boldsymbol{q}_{G,0}\boldsymbol{q}_{G,2}\right) \\ 2\boldsymbol{b}_{L,1}\left(\boldsymbol{q}_{G,1}\boldsymbol{q}_{G,2} - \boldsymbol{q}_{G,0}\boldsymbol{q}_{G,3}\right) + 2\boldsymbol{b}_{L,3}\left(\boldsymbol{q}_{G,0}\boldsymbol{q}_{G,1} + \boldsymbol{q}_{G,2}\boldsymbol{q}_{G,3}\right) \\ 2\boldsymbol{b}_{L,1}\left(\boldsymbol{q}_{G,0}\boldsymbol{q}_{G,2} + \boldsymbol{q}_{G,1}\boldsymbol{q}_{G,3}\right) + 2\boldsymbol{b}_{L,3}\left(\frac{1}{2} - \boldsymbol{q}_{G,1}{}^2 - \boldsymbol{q}_{G,2}{}^2\right) \end{bmatrix}$$

Appendix Equation L.16. Direction of Earth's magnetic field relative to sensor