A magnetometer based payload for a PTOL UAV with application in geophysical surveys

Darcy Ocker

2014

A dissertation submitted to the Department of Electrical Engineering, University of Cape Town, in partial fulfilment of the requirements for the degree of Master of Science in Engineering.

Supervisors:

Robyn Verrinder	University of Cape Town
Danie Gouws	SANSA Space Science
Elda Saunderson	SANSA Space Science

This page is intentionally left blank.

Abstract

Applying the principles of physics to studying the Earth has given rise to the field of geophysics, which has been recognised as a separate discipline since the 19th century. The practical implementation of this field has led to a separate branch, aptly named exploration geophysics.

Exploration geophysics aims to measure various naturally occurring phenomena associated with the Earth in order to make predictions about what might lie beneath the Earth's surface. One of the fundamental phenomena associated with the Earth is the magnetic field or geomagnetic field. By localising magnetic anomalies within the geomagnetic field one can make predictions or inferences about the localised geophysical makeup and potential ore bodies, hydrocarbon deposits or archaeological artefacts that might exist below the surface. The fundamental sensor used to perform these surveys is the magnetometer.

The concept of an unmanned aerial vehicle (UAV) has been around since 1915, with the first manufactured UAV appearing in 1916. Subsequent to the realisation of the UAV in the 1950s by Ryan Aeronautical for military reconnaissance, the idea of using UAV platforms to perform dull, dirty and dangerous functions has become common-place in the military environment. The first practical use of a UAV came in the 1991 Gulf War.

The subsequent appearance of UAVs in the civilian realm can largely be attributed to the advent of low cost, high power density, lithium based batteries in the 1990s and the growth of the radio controlled (RC) hobbyist market.

This dissertation presents the analysis, design and practical implementation of a magnetometer based payload for a point-take-off-and-land (PTOL) UAV. This includes:

1. The development of a dual tri-axial fluxgate magnetometer datalogger payload and the evaluation thereof.

2. The construction and evaluation of a PTOL UAV.

3. The characterisation of the magnetic signature for the PTOL UAV's propulsion system along with subsequent magnetic noise mitigation techniques.

4. Finally, the payload is used to detect a characterised magnetic dipole target to ascertain the performance of the sensors and associated systems.

To my wife, Heidi. I could never have done it without the nagging!

...you can stop now.

☺

This page is intentionally left blank.

Acknowledgements

I would like to thank the following people and organisations for their valuable assistance in making this research a reality:

- Cybicom Atlas Defence for their support and generous sponsorship and funding towards this research

- Ms Robyn Verrinder for her expert supervision, motivation and dedication to this research

- Mr Danie Gouws and Mrs Elda Saunderson for their supervision of this project and advice and assistance with the experiments conducted in this dissertation

- SANSA Space Science for their generous hospitality and the loan of the magnetometers to make this research possible

- Bruce Johnson for his friendship and invaluable assistance with the construction of the UAV platform and flight trials

- Ismail Jassiem and Jaco van Eeden for their assistance in organising and facilitating of the PCB manufacture and various component construction to aid this research

- Darian van Eck for his assistance during flight trials

- Richard Asher for his long serving friendship and invaluable assistance in proof reading this dissertation

This page is intentionally left blank.

Declaration

I declare that this project is my own work; all sources that I have used have been referenced and are listed in the Bibliography. This work has not been submitted before for any other degree or examination in any other university.

D. H. C. Ocker

_____

Noordhoek

January 2014

This page is intentionally left blank.

Contents

Contents

This page is intentionally left blank.

Figures – Main

Figures

Figures

Figures

Figures – Appendices

Figures

Figures

Figures

Figures

This page is intentionally left blank.

Tables

This page is intentionally left blank.

Chapter 1 – Introduction

1.1 Background

Applying the principles of physics to studying the Earth has given rise to the field of geophysics, and has been recognised as a separate discipline since the 19th century [1]. The practical implementation of this field has led to a separate branch, aptly named exploration geophysics.

Exploration geophysics aims to measure various naturally occurring phenomena associated with the Earth in order to make predictions about what might lie beneath the Earth's surface. One of the fundamental phenomena associated with the Earth is the magnetic field or geomagnetic field. By localising magnetic anomalies within the geomagnetic field one can make predictions or inferences about the localised geophysical makeup and potential ore bodies, hydrocarbon deposits or archaeological artefacts that might exist below the surface. The fundamental sensor used to perform these surveys is the magnetometer.

The concept of an unmanned aerial vehicle (UAV) has been around since 1915 [2], with the first manufactured UAV appearing in 1916 [3]. Subsequent to the realisation of the UAV in the 1950s by Ryan Aeronautical for military reconnaissance, the idea of using UAV platforms to perform dull, dirty and dangerous functions has become commonplace in the military environment [3], [4]. The first practical use of a UAV came in the 1991 Gulf War [3].

The subsequent appearance of UAVs in the civilian realm can largely be attributed to the advent of low cost, high power density, lithium based batteries in the 1990s and the growth of the radio controlled (RC) hobbyist market [4], [5], [6].

Introduction

This dissertation presents the analysis, design and practical implementation of a magnetometer based payload for a point-take-off-and-land (PTOL) UAV. This includes:

1. The development of a dual tri-axial fluxgate magnetometer datalogger payload and the evaluation thereof.

2. The construction and evaluation of a PTOL UAV.

3. The characterisation of the magnetic signature for the PTOL UAV's propulsion system along with subsequent magnetic noise mitigation techniques.

4. Finally, the payload is used to detect a characterised magnetic dipole target to ascertain the performance of the sensors and associated systems.

1.2 Scope

The scope and goals of this project are as follows:

1. Review the existing literature (including -both background and current work) to determine what has and what has not been done previously

2. Review magnetometer technology

3. Construct a datalogger system that will sufficiently digitise and store magnetometer data that can be integrated into a PTOL UAV

4. Review COTS UAV technologies and platforms

5. Construct an inexpensive PTOL UAV with at least one hour flight endurance

6. Perform flight tests with the PTOL UAV in the field

7. Perform DCBL motor and LiPo battery propulsion system characterisation with the data logging system constructed

8. Analyse the data gathered and post process the data to mitigate against the magnetic noise signature of the UAV propulsion system

9. Perform experiments using the data logging system and the magnetometers to detect a known magnetic dipole target

10. Draw conclusions on the results obtained

11. Make recommendations and discuss possible future work

1.3 Limitations

Due to the magnitude and complexity of this project, some limitations have been applied. These are:

1. The magnetometers chosen as the primary detection sensor for the magnetic survey payload have been limited by budget. These magnetometers are tri-axial fluxgate magnetometers[1] and have a limited measurement range of ± 60000 nT. This range is adequate for South African conditions where the local total magnetic field intensity varies from ca. +25000 nT to +30000 nT. Total magnetic field intensity can exceed +65000 nT toward the Earth's poles.

   See Appendix A.

2. The system designed and built during this project will not be for field use but only built as a concept demonstrator.

3. The electronic modules and components chosen for digitisation and data-logging shall be commercial off-the-shelf (COTS) where possible, and should keep costs low but meet the design requirements imposed in Chapter 4.

4. The dissertation shall not consider obstacle detection and/or avoidance applied to UAVs.

[1] The magnetometers have been kindly loaned from SANSA Space Science, formerly the Hermanus Magnetic Observatory (HMO).

1.4 Plan of Development

This dissertation is organised in the following manner:

Chapter 1– Introduction

Provides an introduction to the dissertation and describes how the research was undertaken.

Chapter 2 – Literature Review: Background

This chapter outlines the background research undertaken by others in this field.

Chapter 3 – Literature Review: Current Work

Outlines the current work being undertaken in this field of research.

Chapter 4 – System Design Parameters

The system design parameters that helped provide guidance during my project.

Chapter 5 – Initial Prototype: Design and Evaluation

Initial prototype construction and evaluation.

Chapter 6 – Prototype Datalogger: Design and Evaluation

Datalogger construction and evaluation.

Chapter 7 – PTOL UAV Design, Construction and Evaluation

Construction of the PTOL UAV and the flight trials used to evaluate the platform.

Chapter 8 – Propulsion System Characterisation

Characterisation of the DCBL motor and LiPo battery propulsion system noise.

Chapter 9 – Magnetic Dipole Detection

Summary of the magnetic dipole detection experiments conducted at SANSA Space Science.

Chapter 10 – Conclusion and Recommendations for Future Work

Recommendations based on the outcome of the research found in this dissertation.

Appendices

Supplementary figures, tables and source code to make this research possible.

Chapter 2 – Literature Review: Background

2.1 Geophysical Exploration

Geophysical exploration, or surveying, involves taking measurements near the Earth's surface and analysing the measurements using geophysical models to predict what might be below the Earth's surface [7].

Various surveying techniques exist that allow measurements to be performed and analysed. A broad division of techniques can be made: passive measurements of the Earth's natural field, or active measurements where the effects of local energy inputs (usually electrical, electromagnetic or seismic) are measured to make deductions about buried geological features of interest. Common geophysical survey methods currently employed to do this are [7], [8], [9]:

1. Seismic
2. Geodesy and Gravity
3. Magnetic
4. Electrical
    a. Resistivity
    b. Induced Polarisation
    c. Self-potential
    d. Electromagnetic
    e. Remote Sensing (both active and passive sub-groups)

This dissertation concerns itself with magnetic surveys using unmanned aerial vehicles (UAVs).

2.1.1 Magnetic Surveying

Certain minerals display the property of ferromagnetism. Minerals contained in rocks or soils can display a strong magnetisation and as such can produce a strong localised magnetic field which is superimposed on the Earth's geomagnetic field [7]. The most abundant and strongest magnetic mineral is magnetite, which occurs in the upper regions of the Earth's crust [10].

In Figure 2-1 (a) below the geomagnetic fields are related using Pythagoras' theorem as follows:

$$B^2 = H^2 + Z^2 \qquad (2.1)$$

where:

1. B is the total field vector
2. H the horizontal intensity (normally made up of north (X) and east (Y) components) [11]. This should not be confused with the magnetising force which is often represented by the letter H.
3. Z the vertical intensity

In principle if one wants to model the localised geomagnetic field change when a magnetic anomaly (such as an ore body) is introduced. Figure 2-2 below shows a change in the total field vector by ΔB, which one would like to resolve into a horizontal (ΔH) and vertical (ΔZ) components which relate to the 'localised' change in the magnetic declination and inclination respectively of the geomagnetic field.

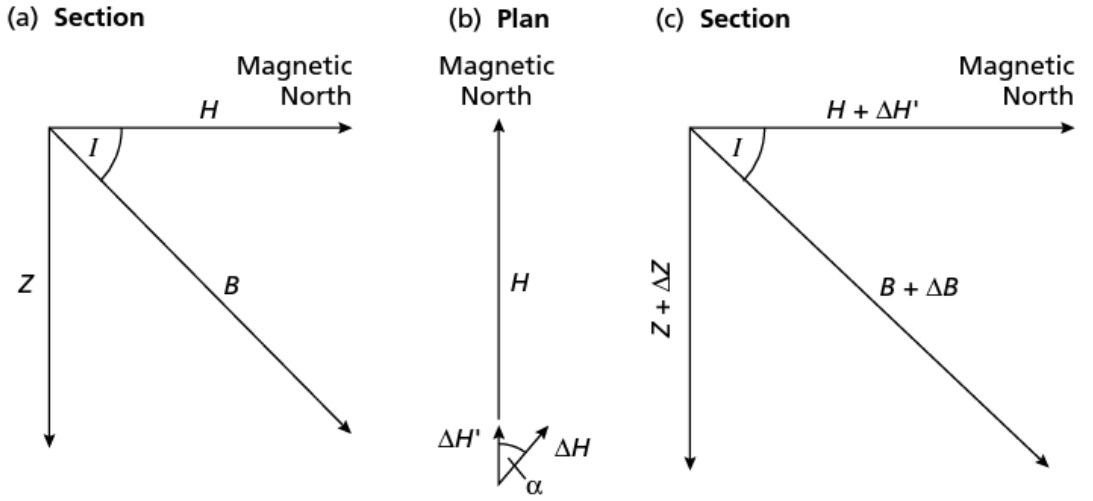


Figure 2-1: A vector representation of the geomagnetic field with (c) and without (a) the introduction of a magnetic anomaly. Figure (b) shows the angle (represented by α) between magnetic north (*H*) and Δ*H* (horizontal component due to the magnetic anomaly) and the vector component of Δ*H* parallel to magnetic north, related as follows: Δ*H*'= Δ*H* cos α. Angle *I* shown in figure (a) and (c) is the angle from the horizontal toward the vertical (*Z*), often referred to as the magnetic inclination or dip angle [12].

Taken from [7].

In general, the horizontal magnetic field component associated with the anomaly (Δ*H*) will not align with the horizontal geomagnetic field (*H*). Assuming an offset angle α between the two vectors (see Figure 2-1 (b)), the vector component of the horizontal field component aligned with *H* associated with the anomaly (Δ*H'*) can be represented by [7]:

$$\Delta H' = \Delta H \cos \alpha \qquad (2.2)$$

Using equation *(2.2)* to solve for the various component anomalies in into equation *(2.1)* one arrives at [7]:

$$(B + \Delta B)^2 = (H + \Delta H')^2 + (Z + \Delta Z)^2 \qquad (2.3)$$

Which can be expanded and substituted into equation *(2.1)*. Disregarding the insignificant Δ^2 terms simplifies equation *(2.3)* to [7]:

$$\Delta B = \Delta Z \frac{Z}{B} + \Delta H' \frac{H}{B} \qquad (2.4)$$

Substituting the angular descriptions into equation *(2.4)* for localised offset angles to the magnetic declination (angle α) and magnetic inclination or dip angle [12] (angle *I*), of the geomagnetic field gives [7]:

$$\Delta B = \Delta Z \sin I + \Delta H \cos I \cos \alpha \qquad (2.5)$$

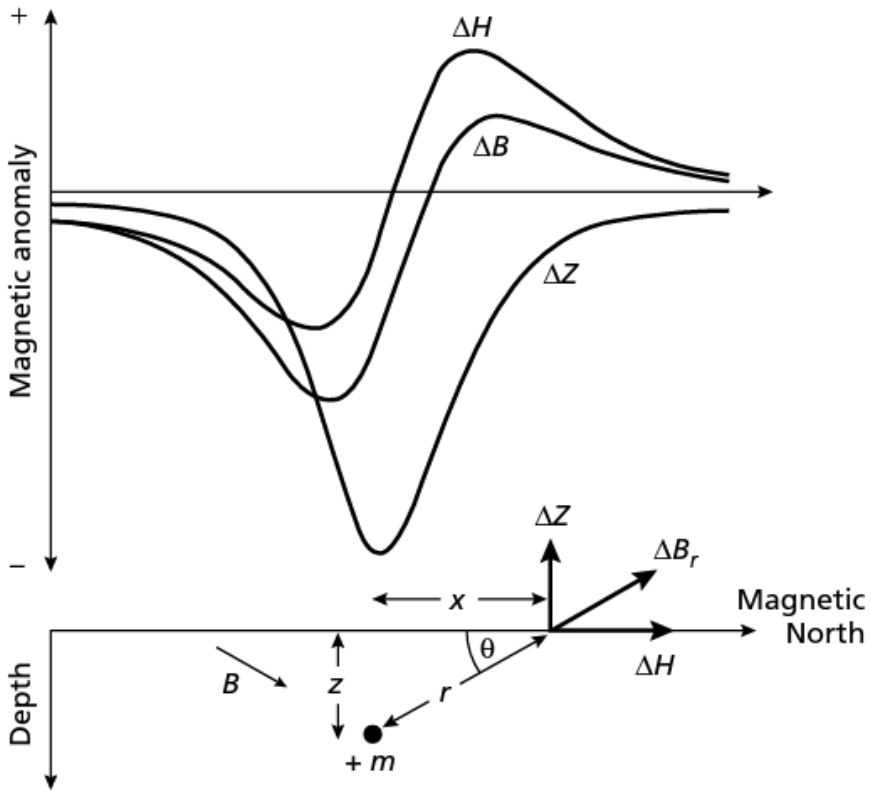


Figure 2-2: Horizontal ΔH, vertical ΔZ and total ΔB field anomalies due to isolated positive magnetic monopole of strength *m*. The angle θ is the angle of the distance *r* from the observation (measurement) point to the Earth's horizontal aligned with magnetic north. *x* and *z* are the respective horizontal and vertical components of vector *r*.

Taken from [7].

One can use equation *(2.5)* to relate the effect of an anomaly caused by a hypothetical magnetic monopole of strength *m* to the geomagnetic field. Assume that the monopole is situated at a distance *r* away from the measurement point (see Figure 2-2 above), with depth *z* and a horizontal distance *x* being vector components of *r*. Therefore, the associated magnetic field (*B*) due to the monopole is given by [7]:

$$B = \frac{\mu_0 \, m}{4\pi\mu_R r} \qquad (2.6)$$

where:

1. μ_0 is the magnetic permeability of free space
2. μ_R is the relative magnetic permeability of the medium (assumed to be 1 for air [13])

Using equation *(2.6)* to solve for ΔB_r (the change in magnetic field associated with the magnetic monopole in the direction of *r*) and the associated horizontal (ΔH) and vertical (ΔZ) vector components [7]:

$$\Delta B_r = \frac{\mu_0 \, m}{4\pi r^2} \qquad (2.7)$$

$$\Delta H = \frac{\mu_0 \, m}{4\pi r^2} \cos\theta = \frac{\mu_0 \, mx}{4\pi r^3} \qquad (2.8)$$

$$\Delta Z = -\frac{\mu_0 \, m}{4\pi r^2} \sin\theta = -\frac{\mu_0 \, mz}{4\pi r^3} \qquad (2.9)$$

It has been assumed that the magnetic profile lies parallel to magnetic north and the horizontal component lies in the same direction (i.e. $\alpha = 0$) for equations *(2.7)*, *(2.8)* and *(2.9)* and that the negative sign in equation *(2.9)* arises from the convention that the z-axis is positive in the downward direction [7].

Solving for the total field anomaly ΔB is done by substituting equations *(2.8)* and *(2.9)* into equation *(2.5)* to get [7]:

$$\Delta B = \frac{\mu_0 mx}{4\pi r^3}\cos I - \frac{\mu_0 mz}{4\pi r^3}\sin I \qquad (2.10)$$

One should take note that ΔB is proportional to $1/r^3$.

2.1.2 Ground Based Surveys

Ground based surveys usually consist of a hand-held sensor with a backpack mounted Global Positioning System (GPS) and integrated data recording system. A complete system's weight is approximately 10 kg (see Figure 2-3 below) [14].

For small areas, ground based surveys are typically preferred to airborne surveys but as the desired survey area increases the labour intensive nature of ground based surveys makes them uncompetitive.

A rough order of magnitude for ground based survey is approximately USD 100 per line kilometre (from a private discussion with a remote exploration company). The actual cost depends on many different factors.

Figure 2-3: Examples of ground based survey equipment used by the man on the ground. As one can see this system is cumbersome, especially in difficult terrain, and also prone to user error.

Taken from [15].

2.1.3 Airborne Surveys

Aeromagnetic surveys have become a popular method for mineral exploration and are used to survey large areas for potential mineral deposits. These surveys are often performed by fixed-wing or rotary-wing aircraft [14]. These airborne platforms bring with them a host of challenges. One of the significant challenges faced by aeromagnetic surveys is platform induced magnetic noise affecting the measurements of the sensing instruments. Compensation techniques need to be applied to these surveys to make them useful [14].

There are often logistic challenges when performing an aeromagnetic survey in a remote area. These include the need for a nearby runway/landing strip, access to airspace licences/permissions, fuel, maintenance of aircraft, etc. Because of these challenges airborne surveys are often done on a large scale and with lower resolution/less detail than that of the surveys conducted using a 'man on the ground'. These surveys can cost between USD 10 to USD 120 per line kilometre surveyed [10].

Figure 2-4: Examples of fixed wing and helicopter airborne magnetic survey platforms. Both platforms require landing strips or helicopter landing areas. They also pose logistic problems in terms of fuel access, airspace licences and access to maintenance facilities.

Taken from [16].

2.2 Magnetometers

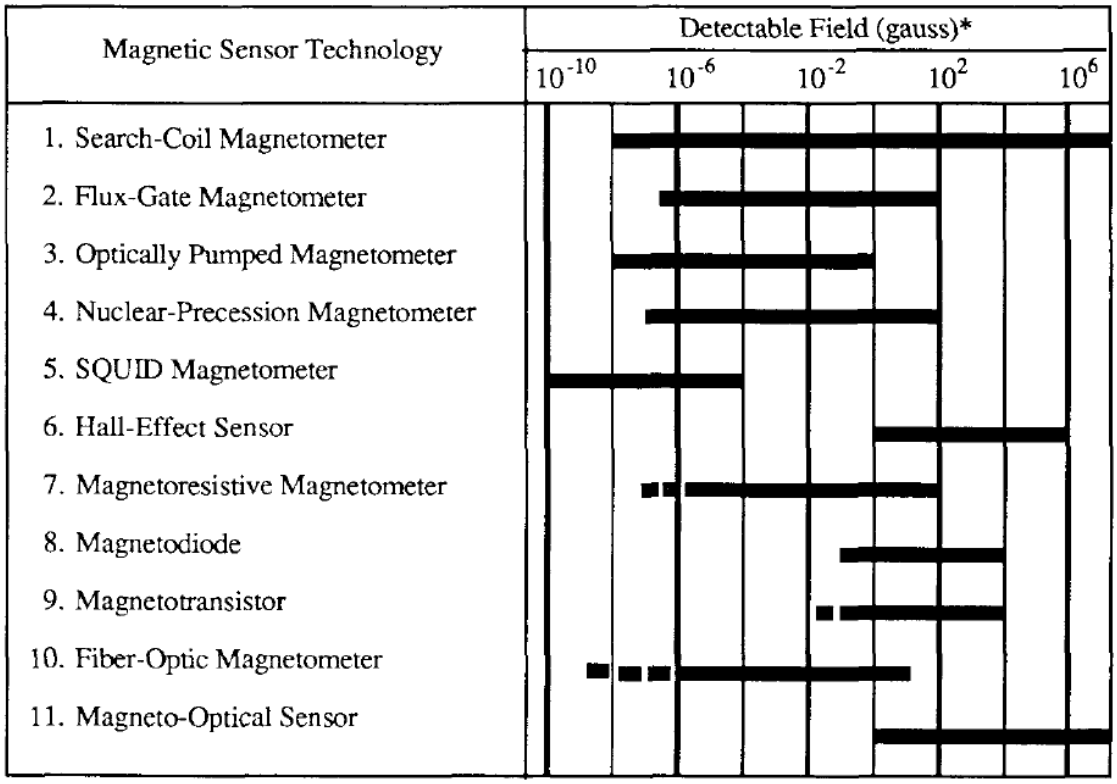
Magnetic sensors have been around for decades. These sensors often rely on the intimate connection between magnetic and electric phenomena to sense the magnetic field [17]. The relationship of these phenomena was initially described by James Clerk Maxwell and published between 1861 and 1862 [18].

In the mid-20th century, Maxwell's equations were later expanded to create a more accurate model of the phenomena and formalised in the theory of quantum electrodynamics [18].

These theories have given rise to many different magnetic sensor technologies. Some of these sensor technologies that exist with their detection range are shown Table 2-1 below.

Table 2-1: Magnetic Sensor Technology and detectable field range. This table was used as a high level guide to determine suitable candidate technologies for application in the system.

Taken from [17].



*Note: 1T = 10^4G = 10^9 γ

[2]

The magnetic sensor technologies listed above have varied applications but for geomagnetic field surveys, sensors with a range of at least 20 000 nT to above 65 000 nT [11], [12] are required. As such the following magnetometer technologies would be suitable:

1. Search coil magnetometer
2. Fluxgate magnetometers
3. Optically pumped magnetometers
4. Nuclear-precession magnetometers
5. Magnetoresistive magnetometers
6. Fibre optic Magnetometers

[2] Note: 1 Gauss = 10^5 nT or 10^{-6} G = 0.1 nT

While the above magnetometers may have a suitable range, they may be unsuitable for field use for various other reasons.

In general, magnetic sensor technology can be divided into three categories, which are specifically related to the way in which the sensor is used in relation to the ever-present geomagnetic field [17]. Category 1 sensors are able to measure fields larger than that of the Earth's magnetic field. Category 2 sensors are able to measure variations in the Earth's magnetic field. This variation comes about in various ways, be it cosmological (e.g. solar flares) or geological. Being able to measuring this variance limits the sensors to Category 2. As such, category 2 sensors are also suitable to geological exploration, while Category 3 magnetometers are limited by the uncontrollable background magnetic noise. The categorisation is summarised in the Table 2-2 below.

Table 2-2: Table summary of the categorisation of Magnetic Sensor applications by sensitivity. Category 2 and 3 sensors are suited to geophysical surveys and offer the best potential 'fit' for the system.

Recreated from [17].

Sensitivity		
10^{-5} G 1 nT		1 G 100 000 nT
Category 3 **High Sensitivity**	**Category 2** **Medium Sensitivity**	**Category 1** **Low Sensitivity**
Definition		
• Measuring field gradients or differences due to induced (in the Earth's field) or permanent dipole moments.	• Measuring perturbations in the magnitudes and/or direction of the Earth's field due to induced or permanent dipoles.	• Measuring fields stronger than the Earth's magnetic field.
Major Applications		
• Brain function mapping • Magnetic anomaly detection	• Magnetic compass • Munitions fuzing • Mineral prospecting	• Noncontact switching • Current measurement • Magnetic memory readouts
Most Common Sensors		
• SQUID Gradiometer • Optically pumped magnetometer	• Search-coil magnetometer • Fluxgate magnetometer • Magnetoresistive magnetometer	• Search-coil magnetometer • Hall-effect sensor

Magnetometers can be further grouped into two sub-categories based on their functionality and principles of operation [19]. These sub-categories are:

1. Vector magnetometers

   These devices can measure the magnitude and direction of the magnetic flux-density in three dimensional space [19]. This means that by orientating the device in the direction of largest magnitude, magnetic north (e.g. compass operation) and the magnetic inclination (angle from the Earth's horizontal surface) can be found. A good example of such a device is the fluxgate magnetometer.

   A common type of fluxgate magnetometer used for geophysical exploration is the tri-axial fluxgate magnetometer. These devices have three orthogonally mounted sensors in a single package (or three orthogonally separately mounted fluxgates) allowing the user to measure the total field strength in the three dimensions or total magnetic intensity (TMI) by using Pythagoras' theorem [20]. This makes knowing the orientation or alignment on a platform of the sensor very important for effective usage.

2. Scalar magnetometers

   These magnetometers measure the total magnetic intensity (TMI) passing through the sensor without the direction of the vector. Examples of scalar magnetometers are quantum magnetometers (discussed in section 2.2.1 below), with the exception of super-conducting quantum interference devices (SQUID) which make use of quantum properties but offer vector performance [19], [20].

   Since SQUID magnetometers offer significant logistic challenges associated with cooling which is intrinsic to their operation. This makes these magnetometers currently unsuitable for use on the UAV platform envisioned and so puts these devices out of the scope of this dissertation.

2.2.1 Quantum Scalar Magnetometers

The basis for quantum magnetometer operation is the spin of sub-atomic particles (usually unpaired valence electrons or the Helium 3 isotope) [19]. These sub-atomic particles precess (or rotate) around a plane perpendicular to the magnetic field direction. The rate of precession is called the Larmor frequency (ω_0) and is defined as follows [19], [21]:

$$\omega_0 = \gamma_p B \qquad (2.11)$$

where B is the ambient magnetic flux density which, in general, is proportional to the magnetic field value and γ_p is the gyromagnetic ratio (a characteristic of each particle) [19], [21].

When using this effect to measure the geomagnetic field (a weak magnetic field) the signal of the Larmor frequency is often required to be to be "boosted" or "polarised" to achieve sufficient sensitivity for successful measurement [19], [21], [22].

This "boosting" or "polarisation" proposed by Hrvoic *et al*. can be achieved in the following ways [19]:

1. Application of strong auxiliary magnetic fields.
2. Making use of the Nuclear Overhauser Effect (transferring the natural polarisation of auxiliary electrons to protons).
3. Optical manipulation or "pumping", a phenomenon based on nuclear and electron resonance effects.

These manipulation techniques give rise to the aptly named: proton precession magnetometers (PPMs), Overhauser magnetometers and optically pumped magnetometers respectively.

2.2.1.1 Proton Precession Magnetometers

Proton precession magnetometers (PPMs) are sequential devices which means that a polarisation field must be created before a measurement can be taken. This is often done by applying a direct current (DC) to a solenoid wrapped around a tube of hydrogen rich liquid (often kerosene). This strong magnetic field causes the protons of the liquid to align with this magnetic field. When the DC is turned off, the strong magnetic field collapses and causes the protons to precess at a frequency that is proportional to the ambient magnetic field [19], [23].

A measurement, using sensitive amplifier electronics, is taken of this precession frequency [19]. A disadvantage of this device (specifically to the project) is that the application of a large DC makes these devices power hungry and only allows sensitive measurements to be taken in the order of once per second. As such, the magnetometers allow measurement sensitivities of a fraction of a nanoTesla (0.01 nT to 0.1 nT) [23]. The frequency of precession (f_0) is related as follows:

$$f_0 = \frac{\gamma_p}{2\pi} B \qquad (2.12)$$

where $\gamma_p/2\pi$ = 42.576 MHz/T for Hydrogen [19], [24].

2.2.1.2 Overhauser Magnetometers

Overhauser magnetometers are based on PPMs but use the Nuclear Overhauser Effect to transfer nuclear spin polarisation from valence electrons to the protons in the measurement fluid [25]. This offers several advantages over PPMs including, an order of magnitude better sensitivity and the use of an RF field to align the free radicals, instead of a DC solenoid, making these devices significantly less power hungry (as little as 1 Watt) and affords them better a measurement sampling rate (up to 10 Hz) [19], [26].

This type of magnetometer would be suitable for this project although the significant cost of the device is a consideration.

2.2.1.3 Optically Pumped Magnetometers

These magnetometers rely on the 'pumping' of circularly polarised light of a specific frequency into an alkali vapour causing an excitation in electron energies into an unstable state. These electrons are unable to remain in this configuration and spontaneously decay into a more stable configuration. During this process, the alkali vapour becomes opaque and then transparent to the polarised light, causing the light to be modulated. In detecting the frequency of this modulation effect one can measure the Larmor Frequency [19], [20], [22].

Common types of alkali vapour used are caesium and potassium based giving rise to their respective names. These magnetometers must be heated constantly (45 °C to 55 °C) to form this alkali vapour [19] but these devices offer better performance when compared to PPMs [20].

These devices are not only costly but require heating and as such are unsuitable for the application intended in this project.

2.2.1.4 Quantum (Scalar) Magnetometer Sensitivity

Interestingly, the sensitivity of quantum magnetometers is not dependent on the Larmor frequency but the sensitivity of these devices are related by the following equation [19]:

$$\Delta B = \frac{k\Gamma}{\gamma_n S_n} \qquad (2.13)$$

where S_n is the signal-to-noise ratio, k the constant of proportionality, Γ the spectral line width and γ_n the gyromagnetic constant.

2.2.2 Fluxgate Magnetometers

The fluxgate magnetometer falls into the category of a vector device that measures direct current (DC) and low-frequency alternating current (AC) magnetic fields (up to tens of Hz) [27], [28]. Sensitivities associated with these devices are in the 1 mT range (orientated in the field direction) and with an achievable resolution of 10 pT [28]. Due to their wide measurement range and low noise level (typically < 1 nT) makes fluxgate magnetometers ideal candidates for both ground-based and space applications.

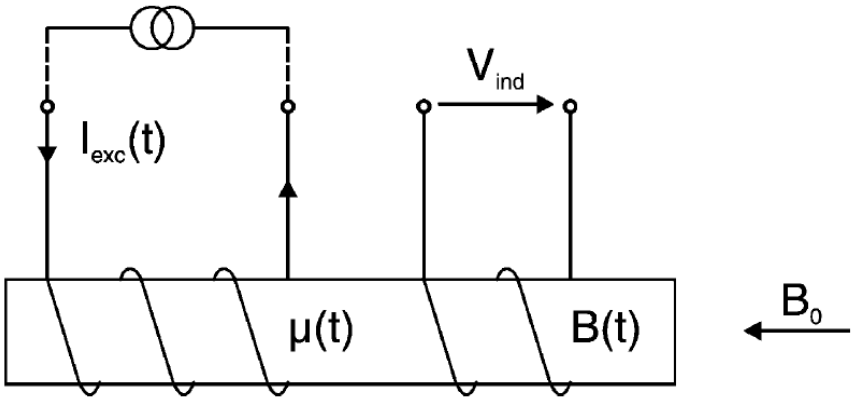


Figure 2-5: Operating principle of a fluxgate magnetometer, showing the relationship between the induced voltage (V_{ind}) on the secondary coil and the measured field (B_0).

Taken from [28].

Figure 2-5 above shows the excitation current I_{exc} applied to the excitation (primary) coil. This excitation current produces a field that periodically saturates the soft magnetic core, around which the coil is wound. This causes the permeability of the material to change, which changes the flux, in turn inducing a voltage (V_{ind}) which is proportional to the measured field on the sensing (secondary) coil. Measuring V_{ind} can then be related back to the measured field (B_0) using the basic fluxgate equation [29]:

$$V_{ind} = nAB_0(1-D)\frac{\frac{d\mu_r}{dt}}{(1+D(\mu_r-1))^2} \qquad (2.14)$$

where:

1. n is the number of turns of the secondary coil
2. A is the cross-sectional area of the core
3. B_0 is the measured field
4. D is the demagnetising factor
5. μ_r is the relative permeability

The basis for the fluxgate action is the time variation of the core permeability (hysteresis) [29].

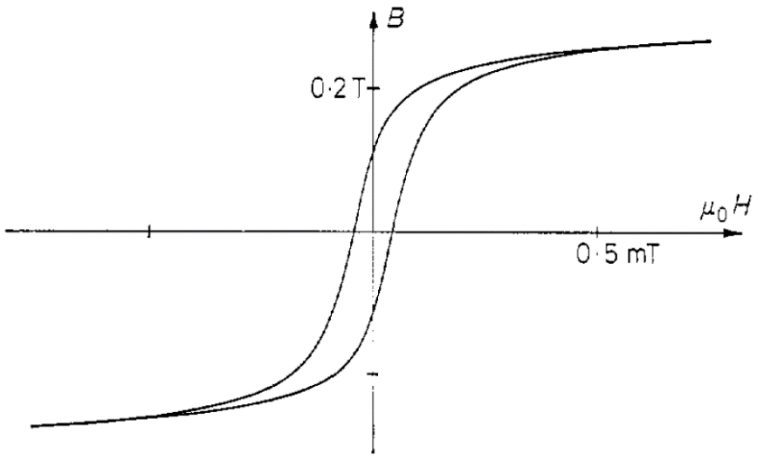


Figure 2-6: The classic hysteresis shape of the plot of the magnetisation curve for B vs $\mu_0 H$ for a tube of Permax 51 ferrite. This hysteresis is responsible for the fluxgate action employed in fluxgate magnetometers.

Taken from [29].

"The most frequently used principle of fluxgate magnetometers is second-harmonic detection of the output voltage" [28]. The output voltage on the secondary (sensing) circuit is demodulated using a phase-sensitive detector (PSD) or lock-in amplifier, in which the output voltage is converted back to very low frequency voltage signal. The integrator in the circuit provides a large feedback gain for which the feedback current is sensed by a differential amplifier across a resistor. This voltage output is sent via a low-pass filter and serves as the magnetometer output [28]. The basic circuit diagram is shown in Figure 2-7 below.

Common excitation frequencies range from 400 Hz to 100 kHz [28], where sensitivity and dynamic performance are proportional to the excitation frequency [28].

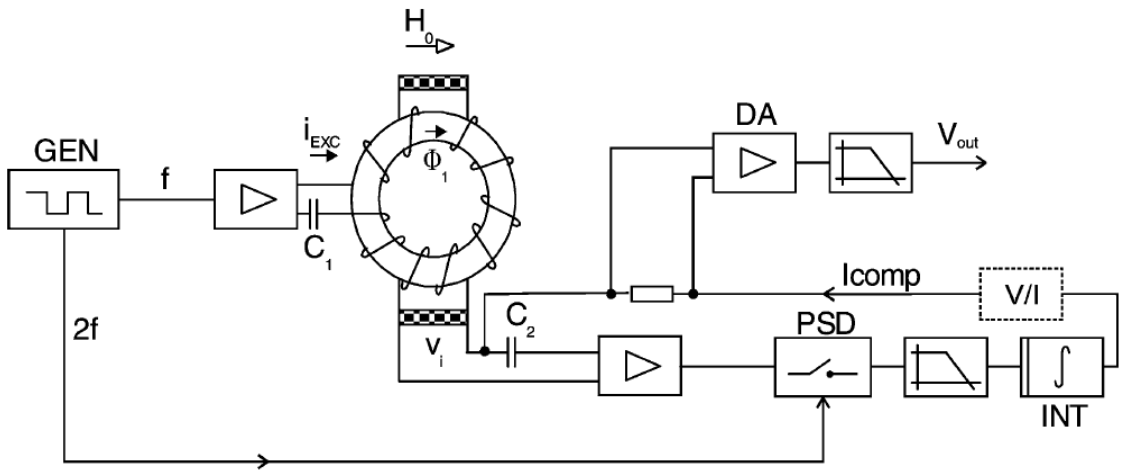


Figure 2-7: Common analogue feedback-type fluxgate magnetometer block diagram. Note the critical components common in fluxgates such as the primary or excitation circuit, secondary or sensing circuit with the phase sensitive detector (PSD)/lock-in amplifier and low-pass filter. Coil configurations vary offering different performance characteristics, a ring core is shown here.

Taken from [28].

2.2.2.1 Fluxgate Magnetometer Sensitivity and Specifications

Fluxgate magnetometers can be divided in to three groups i.e. low- field (< 100 nT), high field (< 100 μT) and very high field (> 100 μT) instruments. The sensitivity is given by [29]:

$$\frac{V_{ind}}{B_0} = nA\frac{d\mu_a}{dt} \qquad (2.15)$$

where μ_a is the apparent permeability of the core which depends on the geometry and permeability of the core material used [28].

Other important specifications of fluxgate magnetometers to note are:

1. Sensor noise
   This is dominated by the Barkhausen (or 1/*f* remagnetisation) noise ranging from 7.9 pT RMS to 0.3 nT RMS in the 0.01 Hz – 10 Hz band [29], [30].

2. Temperature dependence
   Dominated by the temperature dependence of the coil material (usually copper) [29].

3. Fluxgate sensor offset
   This can be measured as the residual magnetometer output when placed in a zero field [29]. Usually this can be calibrated and removed.

4. Long-term stability
   This is largely attributed to environmental conditions and changes over time [29].

5. Exposure to extreme fields
   When the sensor is exposed to extreme fields the performance becomes non-linear, with these extreme fields also being able to change the zero offset of the device [29].

2.2.3 Practical use of Magnetometer Sensitivity

In a paper by Foner *et al.* which discusses magnetometer devices and the merits of their overall sensitivity. It is noted that while overall sensitivity of the device is an important factor, a more important consideration is believed to be the environmental limitations of specific measurements [31]. Taking this into account will help in the choice of device for the task at hand.

This is an important consideration for mounting a very sensitive device on a noisy platform which will effectively limit the performance of the magnetometer device or system. Striking this balance will provide an optimised performance for sensor and platform.

Urquhart *et al.* proposes a table (Table 2-3 below) of survey accuracies for various magnetometer sensors used in real life survey situations [10]. It is interesting to note that fluxgates are only twice as bad in practical aeromagnetic survey situations as the alkali vapour variants especially when price is considered.

Table 2-3: Typical aeromagnetic survey accuracies using various magnetometer sensors. Recreated from [10].

Survey Sensor	Alkali Vapour/[nT]	Proton Precession/[nT]	Fluxgate/[nT]
Resolution	0.01 – 0.25	0.1 – 1	0.1 – 2.0
Instrumental Error	0.01 – 0.5	0.1 – 1.0	0.5 – 1.0
Diurnal etc.	0.5 – 2.0	0.5 – 2.0	0.5 – 2.0
Positioning Errors	0.25 – 5.0	0.25 – 5.0	0.25 – 5.0
Total	0.77 – 4.75	0.95 – 9	1.35 – 10

2.3 Unmanned Aerial Vehicles

The concept of an unmanned aerial vehicle (UAV) has been around since 1915 [2], with the first manufactured UAV appearing in 1916 [3]. Subsequent to the realisation of the UAV in the 1950s by Ryan Aeronautical for military reconnaissance, the idea of using UAV platforms to perform dull, dirty and dangerous functions has become common place in the military environment [3], [4], with the first practical use of a UAV coming in the 1991 Gulf War [3].

The subsequent appearance of UAVs in the civilian realm can largely be attributed to the advent of low cost, high power density, lithium based batteries in the 1990s and the growth of the radio controlled (RC) hobbyist market [4], [5], [6].

The two main categories of UAV are fixed-wing or rotary wing (helicopter) [4]. The ability to fly without a human pilot is made possible by the use of an autopilot control system [4].

2.3.1 Unmanned Aerial Vehicle Classification

Gupta *et al*. proposes a common classification for Unmanned Aerial Vehicle (UAV) platforms to be [3]:

1. Fixed wing UAV
2. Rotary Wing UAV
3. Blimps
4. Flapping wing UAV

Each platform grouping offers significant advantages and disadvantages [3]. This dissertation focuses on fixed and rotary wing UAV platforms.

2.3.1.1 Fixed Wing UAVs

Fixed wing UAVs are unmanned airplanes with wings. These platforms often need runways to take off and land while smaller fixed wing UAVs can be launched by hand or catapult and recovered by net, parachute or simply landing in a suitable area that does not cause damage to the aircraft [2], [3].

The fixed wing UAVs that do not require formal runways are often referred to as point take-off and land (PTOL) UAVs [32].

Fixed wing UAVs generally offer long endurance with high cruising speeds [3].

2.3.1.2 Rotary Wing UAVs

Rotary wing UAVs or rotorcraft UAVs offer the ability to hover while being extremely manoeuvrable [3]. These platforms are often implemented as conventional helicopters with main and tail rotor systems [33], coaxial rotors, tandem rotors or multi-rotor craft [3].

Rotary wing UAV platforms are often referred to as vertical take-off and land VTOL UAVs, although there is work being done on UAV platforms that combine the advantages of rotor-craft and fixed wing as shown by Muraoka *et al.* [34].

2.3.2 Autopilots

The autopilot control system is responsible for various aspects of the flight autonomy of a UAV, which may include take-off, ascent, descent, trajectory-following/way-point navigation and landing [4].

Hobbyist autopilot systems are gaining momentum in the market while some of these systems allow for multiple platform configurations, i.e. fixed and rotary wing platforms such as the Ardupilot system [35].

2.3.3 UAV Powertrains

Wagner *et al.* describe a design for a hand-launchable long endurance fixed wing UAV using commercial-off-the-shelf (COTS) components. The design makes use of a DC brushless motor and lithium polymer batteries that achieves a simulated flight endurance of 5.12 hours for a platform mass of 2.5 kg and wing area of 0.54 m^2 [5]. The simulated results are impressive as the leading long endurance miniature UAV systems have typical flight endurance times of 20 min to 120 min [5].

Figure 2-8: Sample aircraft from [5]. This 2.5 kg test aircraft was used to perform experiments to verify the simulated flight endurance results in the work by Wagner *et al.*

Taken from [5].

2.3.3.1 DC Brushless Motors

Direct current brushless (DCBL) motors are electronically commutated and offer significant advantages over brushed commutation motors and induction motors [36]. Yedamale *et al.* highlights some of the advantages as [36]:

1. Better speed vs. torque characteristics
2. High dynamic response
3. High efficiency
4. Long operating life
5. Noiseless operation
6. Higher speed ranges

These factors and the ratio of torque versus size make DCBL motors ideally suited for use within radio controlled (RC) platforms [37].

Common control methods of DCBL motors are sensor or sensorless control. Common sensors for control of DCBL are Hall Effect sensors to measure the rotor position. While in sensorless control the back electromotive force (EMF) with respect to the phase voltage is used [36].

$$\text{Back EMF} = E \propto NlrB\,\omega \qquad (2.16)$$

where:

1. N is the number windings per turn
2. l is the length of the rotor
3. r the internal radius of the rotor
4. B the rotor magnetic field density
5. ω the motor's angular velocity

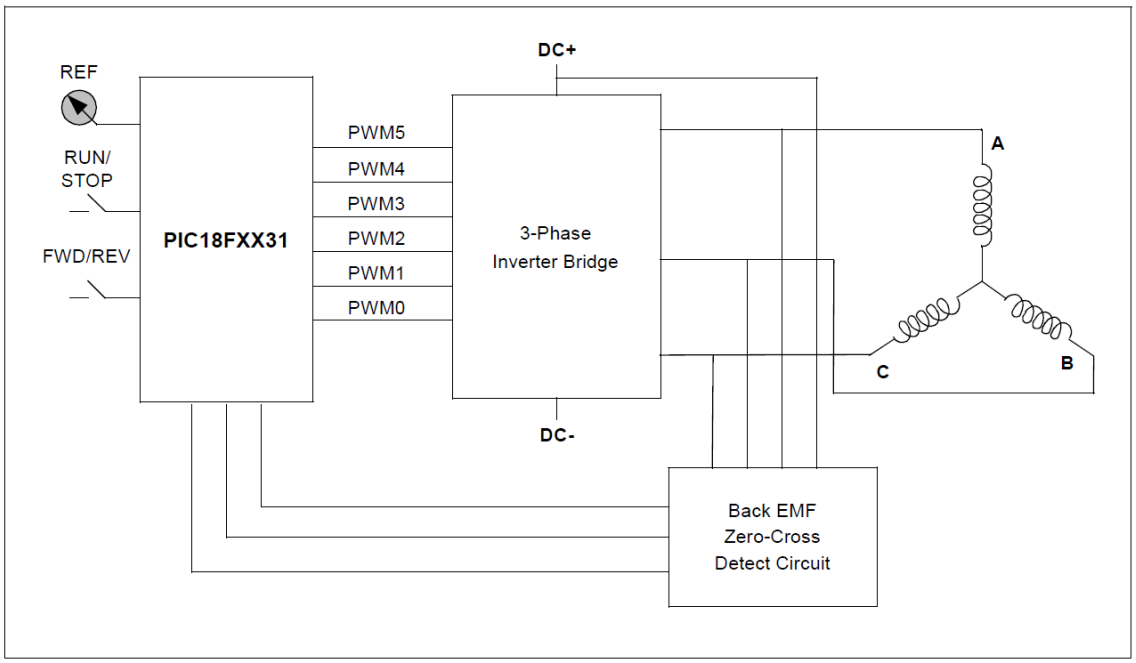


Figure 2-9: Block diagram of sensorless control of DCBL motors. This is very common in RC aircraft in which the electronic speed control (ESC) unit is responsible for detection of the back-EMF zero crossing. The throttle control supplies a PWM signal to the ESC which in-turn controls the DCBL motor speed.

Taken from [36].

Sensorless control of DCBL motors in RC aircraft is very common. The electronic speed control (ESC) unit in an RC aircraft is responsible for the back-EMF zero crossing detection along with the associated power electronics to drive the DCBL motor. The speed of the motor is usually controlled via a PWM input to the ESC from the RC throttle control [36].

2.3.3.2 Lithium Ion Polymer Batteries

Lithium ion polymer batteries have been in commercial use since 1999 [38]. This battery technology offers a specific energy of between 100 to 130 Wh/kg [38], with latest offerings exceeding 135 Wh/kg [39]. Lithium ion polymer cells are very low profile, light weight and therefore ideally suited for use in RC powertrains for which space and weight is of concern.

Varying the specific energy of the power source of an aircraft can be seen in Figure 2-10 below.

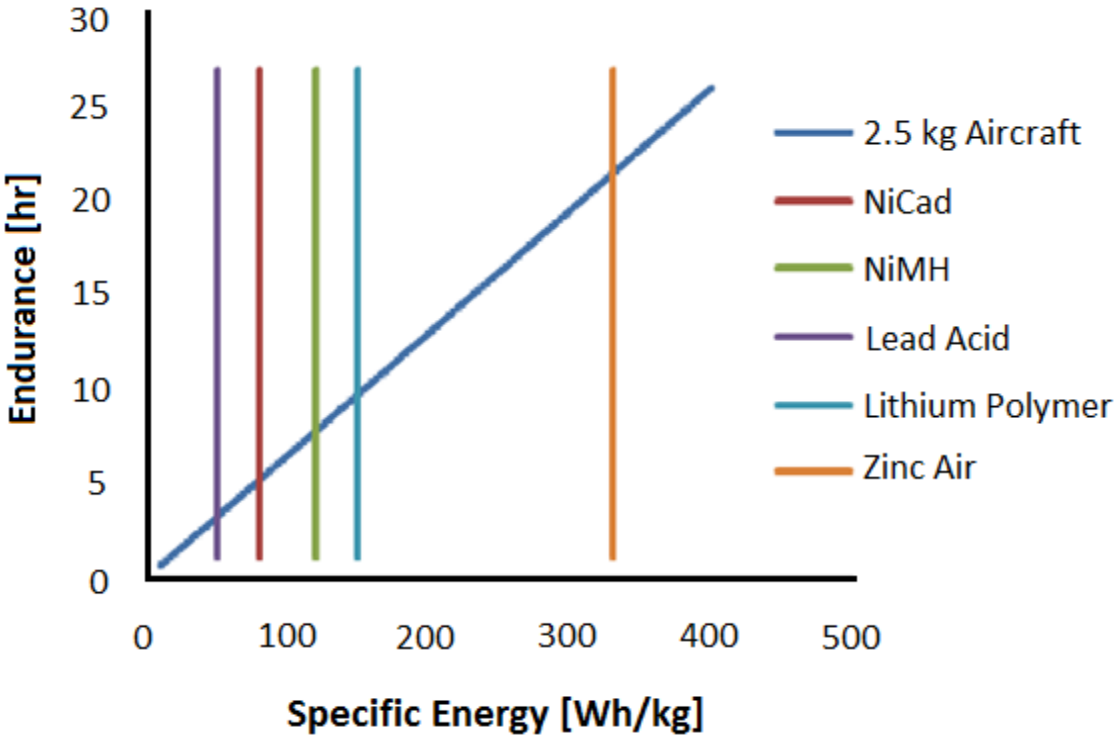


Figure 2-10: Endurance of a sample aircraft vs. battery specific energy. The diagonal line shows the predicted endurance of the 2.5 kg test aircraft using the various battery technologies proposed by Wagner *et al.* [5].

Taken from [5].

Chapter 3 – Literature Review: Current Work

3.1 The use of Magnetometers on UAV Platforms

A feasibility study by Versteeg *et al.* in 2007 to use an autonomous rotary wing UAV and a magnetometer system for the detection of unexploded ordinance (UXO) was described [40]. The study proposes the combination of a caesium magnetometer (commonly used in aeromagnetic surveys) and a commercially available RC based rotary wing UAV platform. In the study a section is dedicated to the magnetic signature of the UAV and the qualification of the noise attributed to the UAV platform and associated propulsion system.

In the study, it was concluded that the UAV platform has significant AC and DC magnetic noise components. It was mentioned that to achieve sufficient data quality some compensation had to be performed. For the DC noise one could place compensation magnets, and/or the replacement of some of the ferrous components would help as well. In addition, a boom of 1 m length was also be used to compensate for these ill-effects [40].

The high frequency effects caused by the UAV rotors were simply filtered out via a digital notch filter which gave rise to noise in the order or 0.5 nT at the chosen boom length. The study also suggests using high speed sampling of the magnetometer data and the use of multiple magnetometers [40].

The feasibility study led to a paper in 2011 by the same authors describing the development of an autonomous rotorcraft for wide area assessment of UXO and munitions and explosives of concern (MEC) [41]. In the paper it was found that one could use magnetometers on rotor craft for the successful detection of small UXO. An experiment was set up using 5 parallel lines with 13 common UXO and MEC targets laid out along a 180 m stretch of the middle of the 5

lines. The data were collected and compared to the same site surveyed without the targets. It was found that all 13 targets could be successfully detected.

Figure 3-1: Common unexploded ordinance (UXO) and munitions and explosives of concern (MEC) targets used in the detection test experiments conducted by McKay *et al.* [41].

Taken from [41].

Figure 3-2: Various RC helicopters used by McKay *et al.* to conduct the trials and experiments for the detection of UXO and MEC with magnetometers mounted on RC based rotorcraft [41].

Taken from [41].

Various papers by Eck *et al.* describe a project between Aeroscout GmbH and Mobile Geophysical Technologies using a *Scout B1-100 UAV* helicopter and a high resolution (24-bit) tri-axial fluxgate magnetometer for precise magnetic surveying of a collapsed open-pit coal mine in Turkey during February 2011 [33], [42], [43].

Figure 3-3: Scout B1-100 UAV helicopter mounted a tri-axial magnetometer and 24-bit ADC and data logging equipment.

Taken from [33].

The results of these papers were similar to those of Versteeg *et al.* [40] and McKay *et al.* [41], in that it was found that successful detection of various magnetic targets could be done with a magnetometer mounted on a rotary wing UAV platform.

3.2 Measurement and Characterisation of Platform Magnetic Noise Signatures

Magnetic interference generated by airborne platforms is one of the most common system noises in airborne surveys [44]. Zhang *et al*. propose that these noise sources can be divided into three categories [44]:

1. Permanent fields
2. Induced fields
3. Eddy-current fields

Magnetic interference is one of the limiting factors for data gathered by traditional airborne platforms used in aeromagnetic surveys [44].

In the paper, Zhang *et al.* propose the use of a traditional 16 term aeromagnetic model for magnetic noise compensation of an airborne platform applied to UAV system [44]. To verify this, some experiments were conducted using a UAV platform which was flown on the same flight line but in the opposite directions. This was done to verify the DC shift associated with the permanent and induced fields when the flight direction changes and the high frequency terms associated with the Eddy-current fields [44].

The results of the measurements confirmed a DC shift of 10 nT when the flight direction changed, however, the high frequency components associated with Eddy-current fields were not present [44]. This was attributed to the low magnetic nature of the construction materials used in the UAV. This makes it possible to ignore the terms in the magnetic compensation model due to the Eddy-current fields, which significantly reduces the complexity of magnetic compensation model down to eight terms from the usual 16 terms [44].

3.2.1 Magnetic Signatures of Brushless Motors

In a paper released by the Australian Department of Defence (DoD), David Clark describes the magnetic signature mapping of a 5 kW and 0.4 kW brushless motors used in manoeuvring a remotely operated vehicle (ROV). Clark points, since the use of the ROV is for mine disposal tasks, the magnetic signature of platform and associated motors is important [45].

Of particular interest, to this dissertation, is the discussion around the influence of the number of poles on the DCBL motor's magnetic signature. Clarke points out that a higher number of poles for similarly sized motors will significantly reduce the magnetic signature of the motor. This is due to the length of the magnetic dipoles diminishing as the number of poles increases. This has to do with the magnetic field of a dipole decreasing with the cube of the distance from the dipole in the far field. The far field condition is described as the distance to the centre of the dipole being much larger than that of the length of the dipole.

In summary, the magnetic signature of a multi-pole brushless motor is decreased by increasing the number of poles as this will place opposing poles closer to each other and therefore reduce the magnitude of the combined magnetic signature of the rotor. In doing so will also most likely cause the strength of each pole to be reduced as each pole will be constructed out of less magnetic material [45]. Figure 3-4 below shows this relationship by the comparison of a two and four pole DCBL motor. This figure shows the dipole length between the poles is reduced when more poles are introduced.

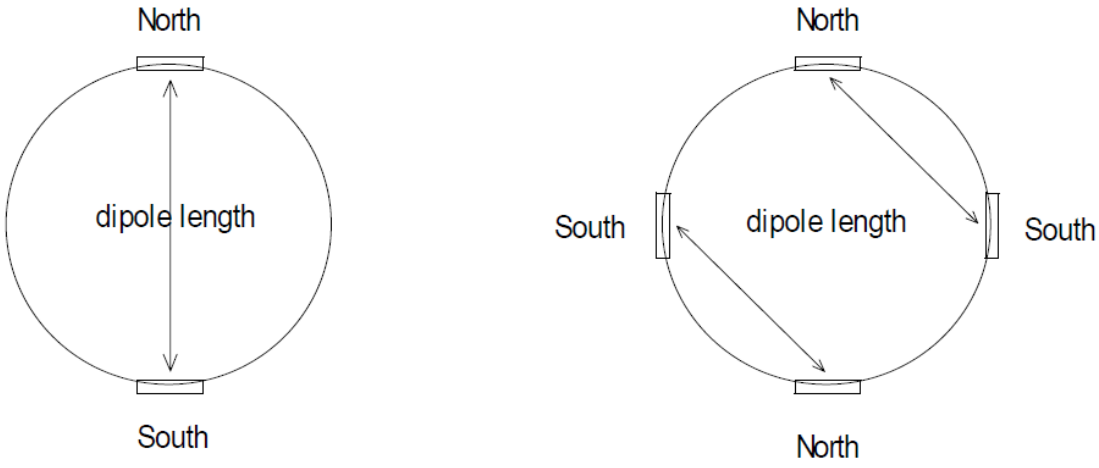


Figure 3-4: Comparison of the dipole lengths of 2 and 4 pole DCBL motor configurations. Note how the length of the dipole is diminished as the number of poles increases.

Taken from [45].

Chapter 4 – System Design Parameters

Due to the multitude of choices available for designing such a system, the system design had to be limited in some way. Inspired by the results of Wagner *et al.* [5], it was decided that the system should be low cost while achieving reasonable performance. The cost effectiveness of the platform and payload combination should allow for scalability and the knowledge gained should be transferrable to different payloads and platform designs/technologies.

4.1 UAV Design Parameters

The following UAV design parameters were imposed:

Table 4-1: UAV Design Parameters

Parameter	Design Metric	Comment
UAV style		PTOL fixed wing UAV
Mass	5 kg	
Flight Endurance	60 minutes or more	
Propulsion System		DCBL motor and LiPo Batteries
Cruising Speed	15 m/s	
Input power	100 W/kg	
Launch and recovery		UAV should be able to take off and land without a runway.
Construction material		The construction of the UAV should minimise the use of ferrous materials where possible.
Autopilot		Shall allow for: 1. Grid flight planning 2. Stabilised flight 3. Telemetry down link 4. Ground station 5. RC compatible
Cost	< ZAR 15 000	

4.2 Datalogger Design Parameters

The following datalogger design parameters were imposed:

Table 4-2: Datalogger design parameters.

Parameter	Design Metric	Comment
Mass	< 1 kg	
Power consumption	< 10 W	Use onboard power from propulsion system.
Size		Shall fit within chosen platform.
Magnetometer	Tri-axial fluxgate magnetometer	Advantages over quantum magnetometers: 1. Low Cost 2. Low Weight 3. Low Power Consumption 4. Small Size 5. Robustness
Digitisation of magnetometer data	≥ 18 bit	Dynamic range ±60 000 nT 1 bit (LSB) < 0.5 nT ADC should have an order of magnitude better specification than the magnetometer.
Micro-controller based system	32 bit	The microcontroller should have a good support and development framework.
GPS time and position stamped data		Up to 10 Hz update rate.
Storage	2 GB or more	Shall be upgradable.
Electronic board design		This should be kept to a minimum and evaluation boards should be used where possible.
Cost	< ZAR 15 000	

Chapter 5 – Initial Prototype: Design and Evaluation

5.1 Choice of Magnetometer Technology

Fundamental to the payload design was the choice of magnetometer technology. Ordinarily, were the choice not constrained by budget, the technology of choice would be one of the scalar quantum magnetometer varieties. These magnetometers are common within ground based and aeromagnetic surveys. Unfortunately these sensors are costly, roughly USD 18 000 to USD 20 000 (as confirmed to me by the manufacturer in 2010) per sensor. Unfortunately this was not feasible within the budget of the project.

The next sensor to consider was the fluxgate magnetometer. Upon further investigation it was found that fluxgate magnetometers offers reasonable performance at an acceptable price. The fluxgate magnetometer that was chosen was the LEMI-011b tri-axial fluxgate magnetometer manufactured by the Lviv Centre of Institute of Space Research (LCISR) [46]. These magnetometers cost in the region of ZAR 5000 per sensor (as confirmed to me by the manufacturer in 2010).

Table 5-1 below shows a comparison of commercially available magnetometers suitable for the application.

Table 5-1: Commercial magnetometer comparison.

Information compiled from [47], [48], [49].

	LEMI-011b	**G-823A**	**UAV Magnetometer**
Manufacturer	LCISR	Geometrics	GEM Systems
Magnetometer Technology	Tri-axial Fluxgate	Caesium Vapour	Potassium Vapour
Picture			
Typical Application	Various	Airborne and vehicle applications	UAV Applications
Operating Range	±60 000 nT Per axis	+20 000 to +100 000 nT Scalar	+15 000 to +120 000 nT Scalar
Operating Mode	Continuous, analogue voltage output	RS 232 Data Output	Up to 20 Hz RS232 Data Output
Sensitivity	27.4 ± 0.3 nT/mV	< 0.004 nT/√Hz RMS	0.0003 nT @ 1 Hz
Accuracy	< 5 nT over ±50 000 nT range	±0.05 nT	< 3 nT throughout range
Size	Sensor: 103 x 60 x 26 mm	Sensor: 60.32 (diam.) x 146 (long) mm Electronics: 63.5 (diam.) x 279.4 (long) mm	Sensor: 64 (diam.) x 158 (long) mm Electronics: 237 x 56 x 39 mm
Mass	Sensor: ≤ 120 g	Sensor: 339 g Electronics: 623 g	Sensor: 650 g Electronics: 630 g
Power Consumption	< 27.5 mW @ 5 Vdc	14 W @ 28 Vdc nominal	14 W @ 28 Vdc nominal
Approximate Cost	ZAR 5 000	USD 20 000	CAD 18 500

5.2 Initial Prototype Design

Once the magnetometer technology was chosen a system design for the payload could be considered. Since the output of the fluxgate is a continuous analogue voltage, one would need to digitise and log the values. How to digitise the output of the magnetometer had to be considered along with the idea of using two fluxgate magnetometers in a gradiometer configuration on a rotary wing VTOL (the initial UAV configuration choice).

The work of Bartington *et al.* [50] was considered as a basis for the use of two magnetometers in a gradiometer configuration taking note of the 1 m spacing between the sensors used in the Bartington *et al.* gradiometer system. This was later deemed unfeasible due to the orientation and configuration of the fixed-wing UAV platform chosen although the dual fluxgate magnetometers remained.

5.2.1 Digitisation of fluxgate magnetometers using ADCs

Korepanov *et al.* [27] analyses the influence of digitisation on the noise of fluxgate magnetometers, specifically the influence of the quantisation noise arising from the analogue-to-digital converter (ADC) on the fluxgate output signal. This input was used to aid in the choice of the ADC.

Each LEMI-011b fluxgate magnetometer has 4 output channels, the magnetic field component for the three orthogonal axes and temperature of the device (B_x, B_y, B_z, T). Each of the magnetic field components has a range of ±60 000 nT and the requirement to have a minimum quantisation of 0.5 nT was imposed, which meant that the minimum resolution for the analogue-to-digital converter (ADC) would be at least 18-bits. The Texas Instruments (TI) ADS 1278 24-bit 8 channel ADC was chosen [51]. The ADC offers two serial interfaces: Frame-Sync (FS) and Serial Peripheral Interface (SPI).

Having an eight channel ADC would allow for simultaneous sampling of all the outputs from both LEMI-011b fluxgate magnetometers. The 24-bit data logger used in conjunction with a fluxgate magnetometers of Eck *et al.* was subsequently noted [43].

The TI ADS 1278 offers an evaluation board (ADS 1278 EVM-PDK) [52], which was modified for use under the initial prototype development. This fitted well with the decision to use commercial-off-the-shelf (COTS) components in the project. The initial prototype block diagram and image is shown in Figure 5-1 below.

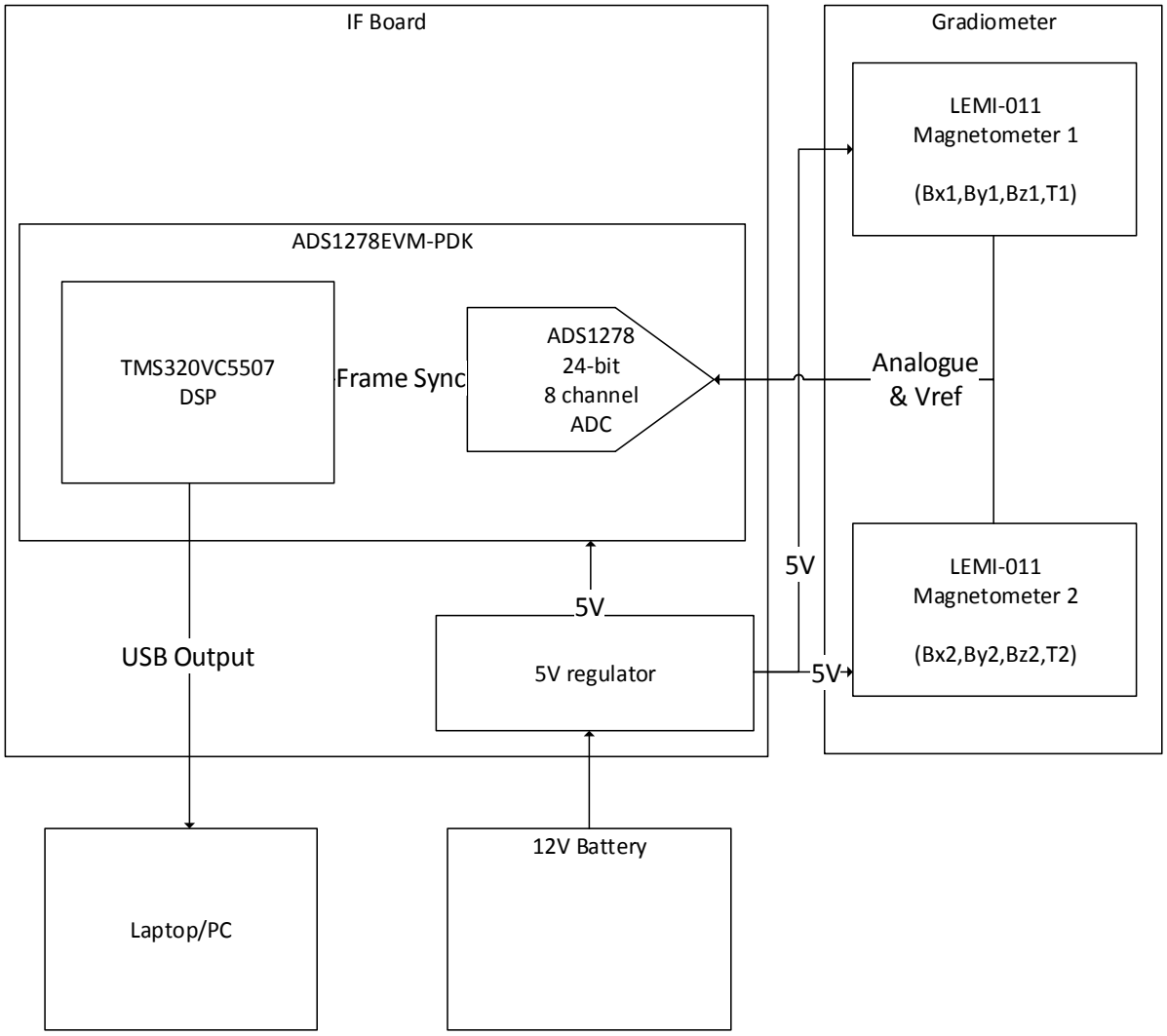


Figure 5-1: Initial payload prototype block diagram. The diagram shows the various components that make up the system.

Of particular concern during the initial tests with the system was the 50 Hz AC noise that was seen in the data. This was thought to be due to the use of alternating current (AC) mains electrical supply to power the system. To mitigate against any spurious signals entering via the mains an interface (IF) board was created to provide power from a 12 V DC battery and route the various I/O pins to and from the magnetometers and ADS 1278 EVM-PDK.

Once the IF board was reworked to run off a 12 V DC battery it was later found that the spurious AC signals remained in the data. The cause was ascribed to the ever-present nature of the electromagnetic interference (EMI) emanating from the electrical power utility. This was later discussed and confirmed with a visiting Canadian scientist to SANSA Space Science in May 2013 who had noted these AC tones (60 Hz found in North America) even in remote parts of Canada. This was interesting especially since the LEMI-011b fluxgate magnetometers have a built-in low pass filter with a 10 Hz cut-off. This became a large noise source in the data and was subject to the noise mitigation techniques discussed later in this dissertation.

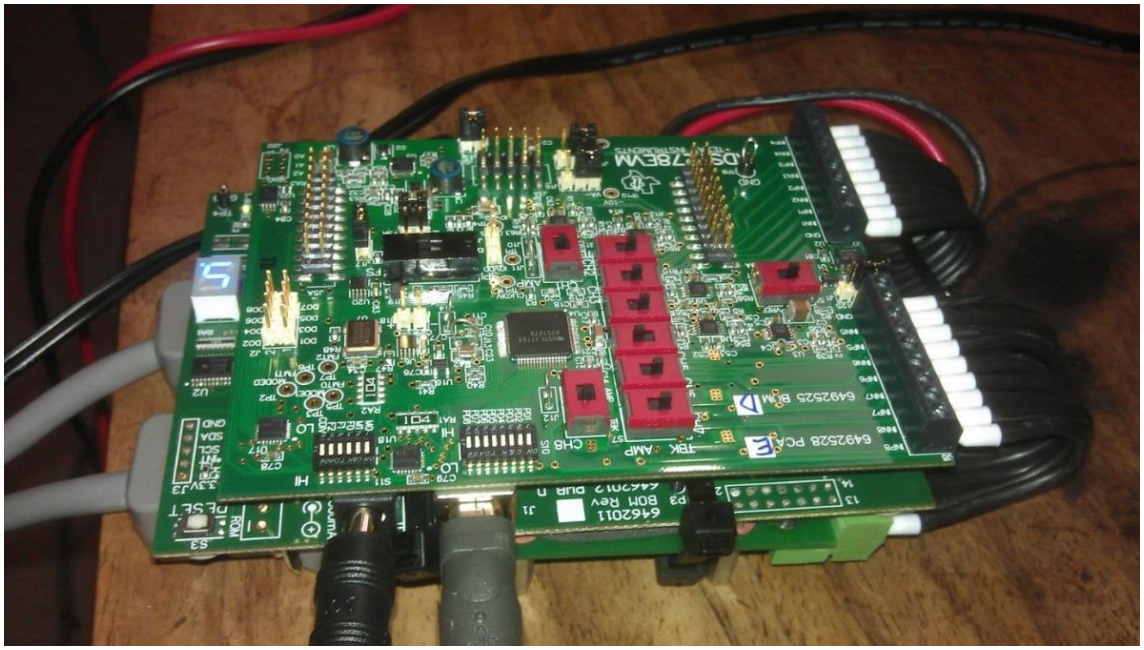


Figure 5-2: Initial prototype - 3 board stack using AC mains PSU. This was later changed to address changes to the new evaluation board and make the system run off a 12 V DC battery.

During the development and testing of the prototype system, the evaluation kit was damaged and a more robust solution was sought for testing. Unfortunately the evaluation kit for the TI ADS 1278 was replaced by a new version (ADS 1x7x EVM-PDK [53]) and required significant rework to the system. The IF board was redesigned to work from a 12 V battery and to be more robust for testing purposes. Figure 5-3 below shows the new test setup.

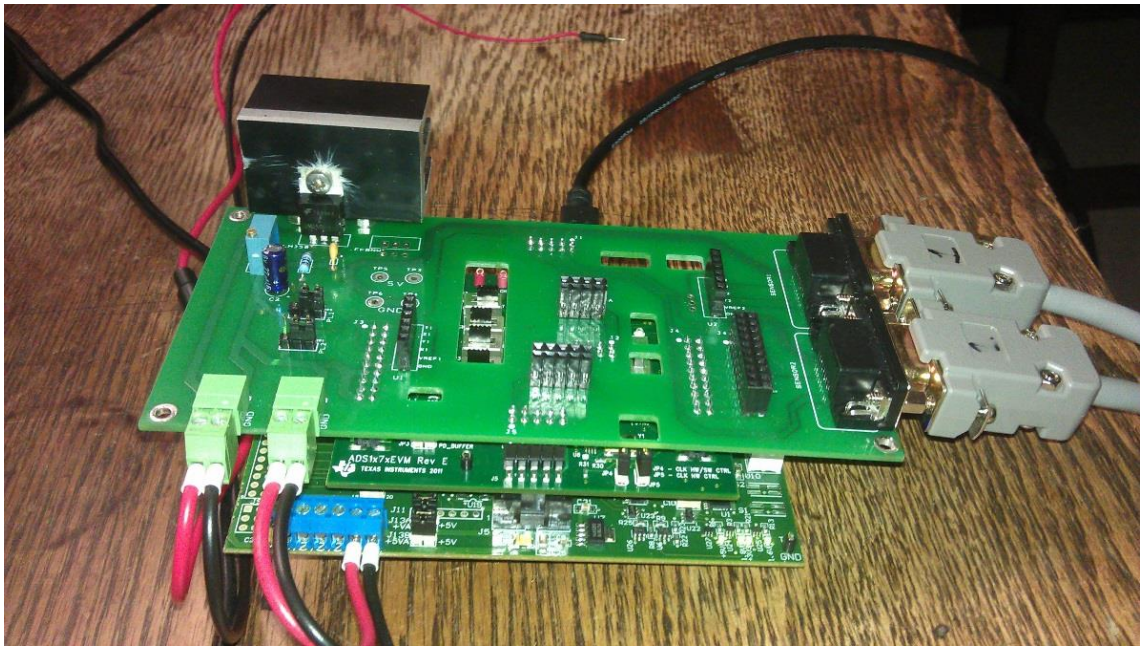


Figure 5-3: Reworked triple-board stack for testing with 12 V battery. The various boards are stacked on top of one another, with the manufactured interface printed circuit board (top) providing power and pinouts to the two boards below.

5.2.1.1 LEMI-011b Fluxgate Magnetometer vs. ADS1278 Specifications

The specifications of the LEMI-011 fluxgate magnetometer were compared to the ADS1278 ADC specifications. The ADC specifications should be an order of magnitude better than that of the sensor being digitised so as to not adversely influence the sensor readings.

Table 5-2: Comparison of Magnetometer and ADC specifications.

Specifications compiled from [47] and [51].

Parameter	**LEMI-011b Fluxgate**	**ADS 1278 ADC**
Bandwidth	0 to 10 Hz (designed for)	70 kHz
DC Accuracy	Zero drift over temperature: < 184.5 µV/°C Non-linearity over temperature: 0.012 %/°C	Offset drift: 0.8 µV/°C Gain drift: 0.00013 %/°C
Analogue requirements	Analogue output: $V_{out} = V_{ref} \pm 2.25$ V DC $V_{ref} = V_{in}/2$ $V_{in} = 5 \pm 0.25$ V DC	Analogue input: 0 to 5 V DC Accepts V_{ref}
Operating modes	Continuous analogue voltage output.	High Speed Mode: Up to 144 kSPS; 106 dB SNR although 105 kSPS was used. High Resolution Mode: 52 kSPS; 111 dB SNR Low Power Mode: 52 kSPS; 31 mW/Channel Low Speed Mode: 10 kSPS; 7 mW/ Channel

Conversion of the digitised sensor output voltage into nanoTesla is done using the following relationship [46]:

$$B_n = \frac{V_{out}}{S} - B_{0n} \quad (5.1)$$

where:

1. B_n is the magnetic field component for the n-th axis (x, y, z) in nT
2. V_{out} is the output voltage in V
3. S is the sensitivity in V/nT (36.5 ± 0.4 µV/nT for the LEMI-011b)
4. B_{0n} is the zero offset for the n-th axis (x, y, z) in nT

5.3 Initial Prototype Evaluation

The prototype design was evaluated and compared against the LEMI-011b magnetometer noise specification of 2 nT peak-to-peak (PTP) and 0.5 nT standard deviation at DC [47].

Figure 5-4 below shows the noise plotted for sensor 1 using the high-resolution sampling function of the ADC. Sampling was done at 52 kSPS. Note that the noise is larger than 6 nT peak-to-peak with a standard deviation of 1.28 nT over the sampled range for B_y. The total magnetic intensity (TMI) for sensor 1 is 25371 nT.

TMI is calculated as [20]:

$$TMI = \sqrt{{B_x}^2 + {B_y}^2 + {B_y}^2} \quad (5.2)$$

Performing spectral analysis for the noise of sensor 1 is shown in Figure 5-5 below. It is interesting to note that even though the LEMI-011b has a built in 10 Hz low-pass filter the 50 Hz mains is clearly visible on the B_y noise along with a prominent 16 kHz line. At first these lines were rather confusing as one does not expect to see such large influences above 10 Hz. It was later confirmed that the 50 Hz tone was caused by the ever-present AC mains supply from the electrical power utility. The 16 kHz line was attributed to the excitation frequency of the LEMI-011b magnetometer and later confirmed in a private conversation with the manufacturer of the LEMI-011b.

An interesting artefact to note is the difference in magnitude of the 16 kHz line seen between the sensor 1 periodogram (Figure 5-5 below) versus that seen in the sensor 2 (Figure 5-7 below) periodogram. It can be seen that sensor 2's 16 kHz line was smaller by at least an order of magnitude when compared with sensor 1. The reason for this is unknown, but is suspected to arise from a manufacturing difference between the two sensors, perhaps in the components used for the excitation circuitry of the magnetometer, or the differences in components used for the low pass filter between the two LEMI-011b magnetometers.

A further example of this can be seen where the ADS1278 was operated in "high speed mode" (> 105 kilo samples per second (kSPS) Figure 5-8 and Figure 5-9). Here one can see the excitation frequency along with the first and second harmonics. Note the significant difference in magnitude between sensor 1 and sensor 2 for these lines above 10 Hz. This seems to point to a construction difference between the two LEMI-011b magnetometers or perhaps a calibration issue.

One should also note the slight differences in the TMI values between sensor 1 and sensor 2 shown at the bottom of the noise plots. These differences indicate that the magnetometers require calibration.

For comparative purposes Figure 5-6 through Figure 5-7 below shows the analysis for sensor 2.

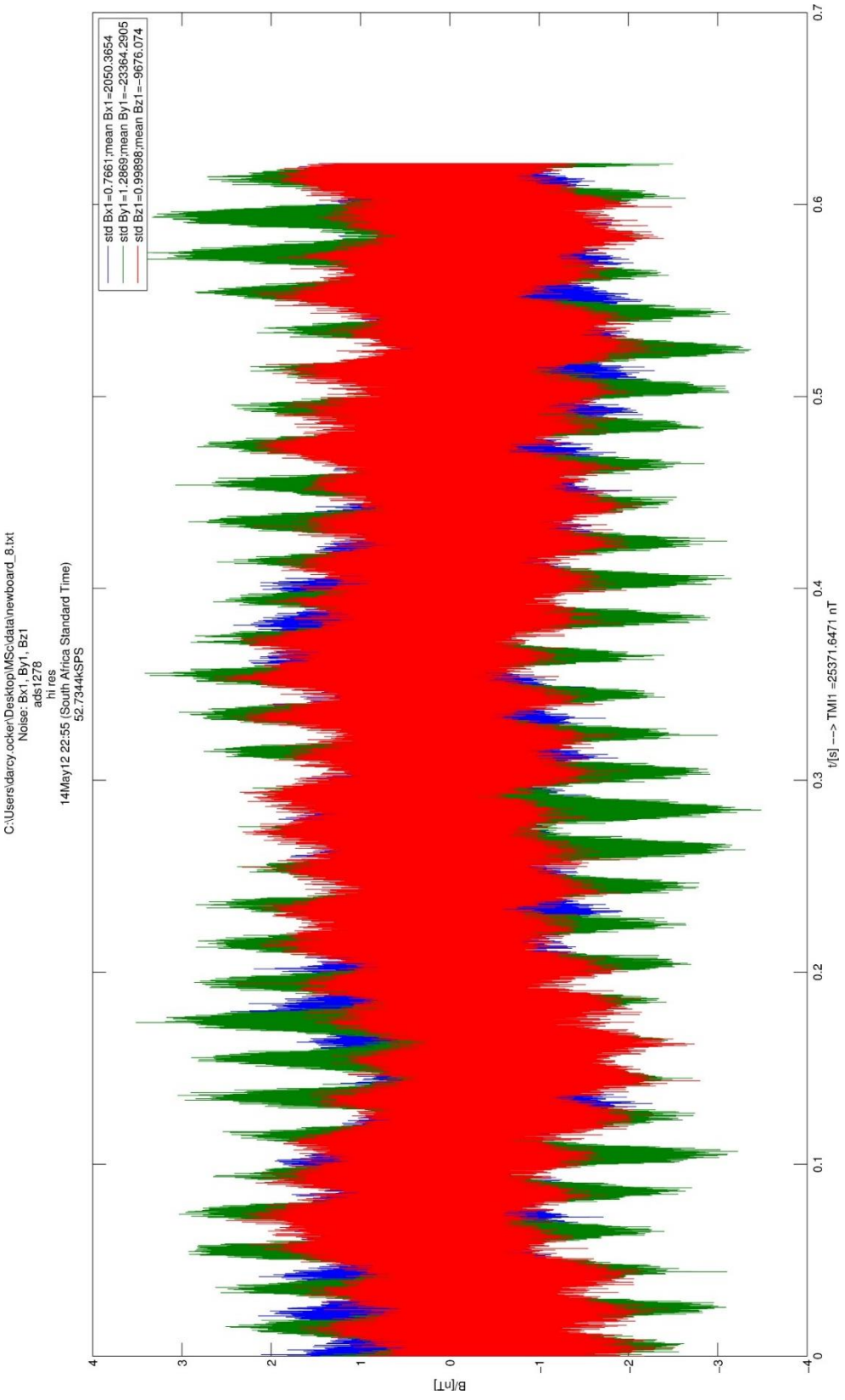


Figure 5-4: Digitised data showing the noise for sensor 1.

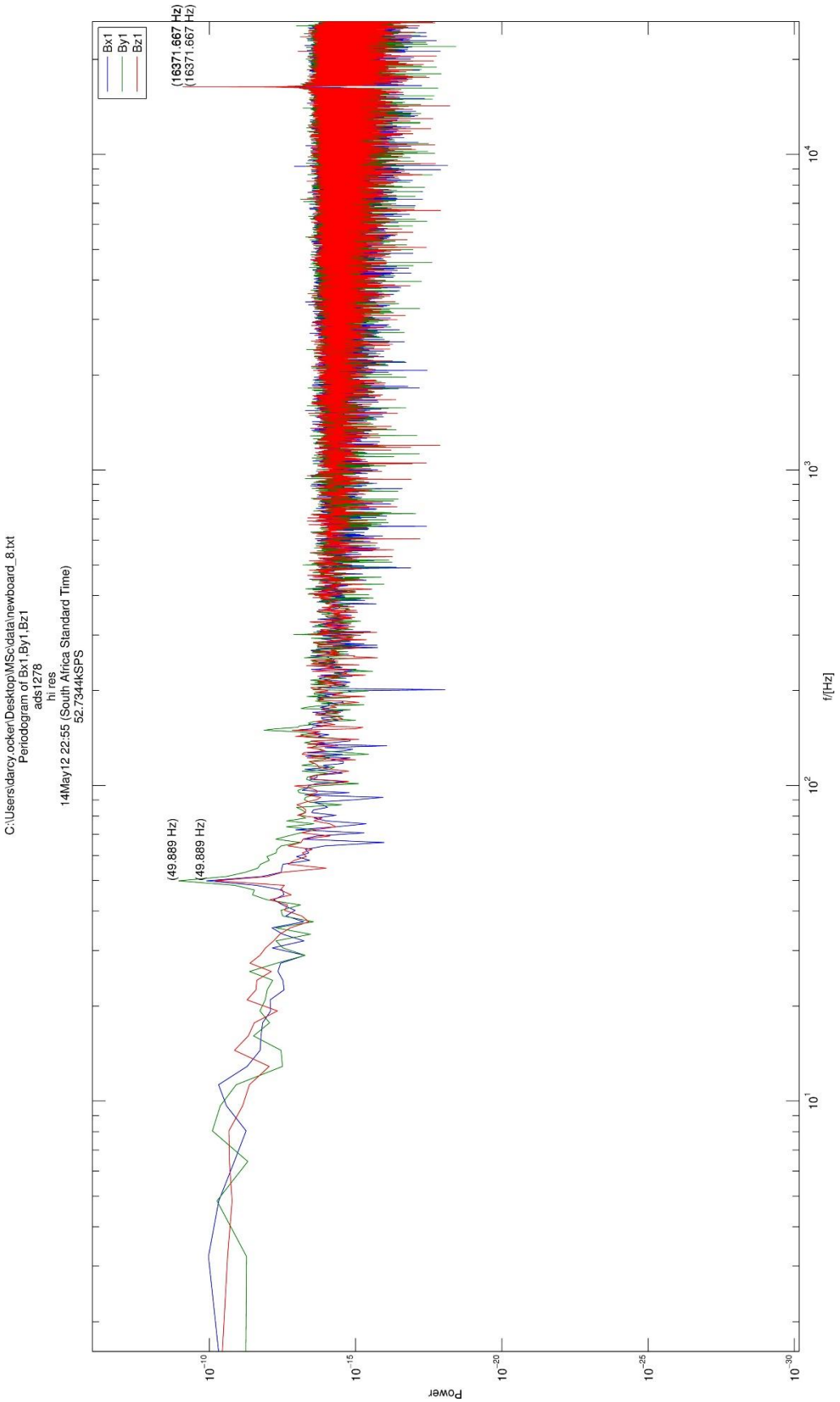


Figure 5-5: Logarithmic plot of the spectral analysis (periodogram) of the noise for sensor 1, noting the tones at 50 Hz and 16 kHz.

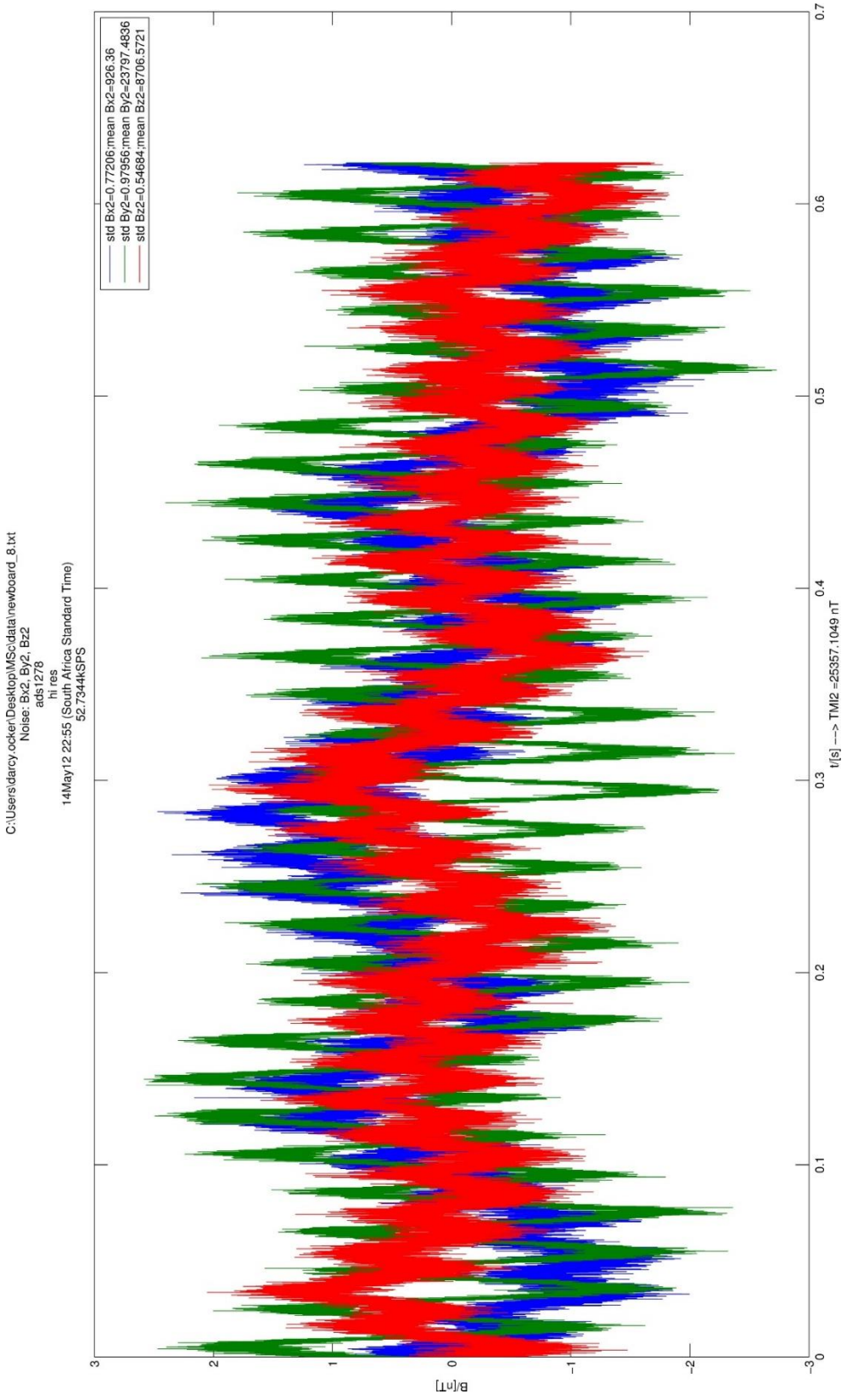


Figure 5-6: Digitised data showing the noise for sensor 2.

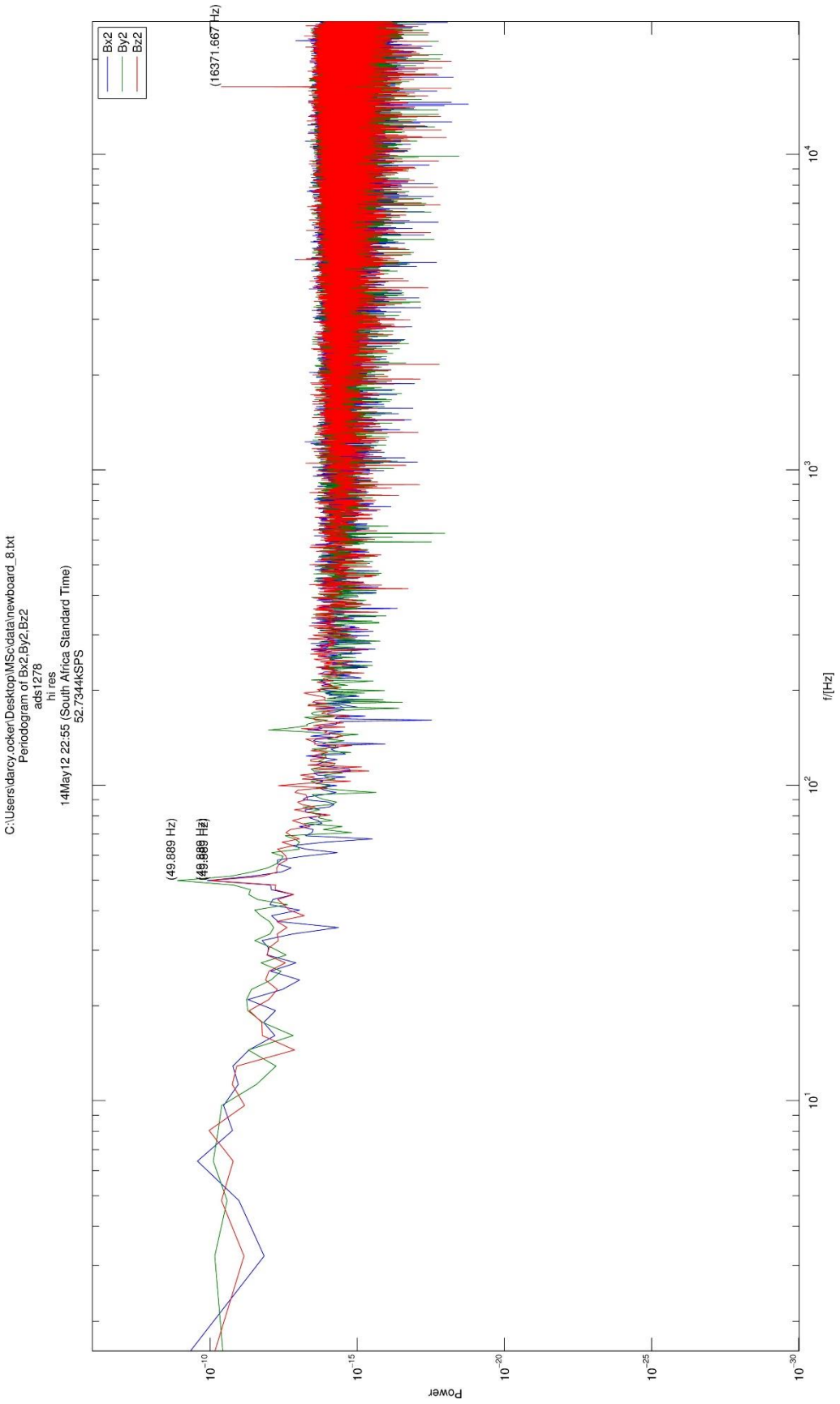


Figure 5-7: Logarithmic plot of the spectral analysis (periodogram) of the noise for sensor 2, noting the tones at 50 Hz and 16 kHz.

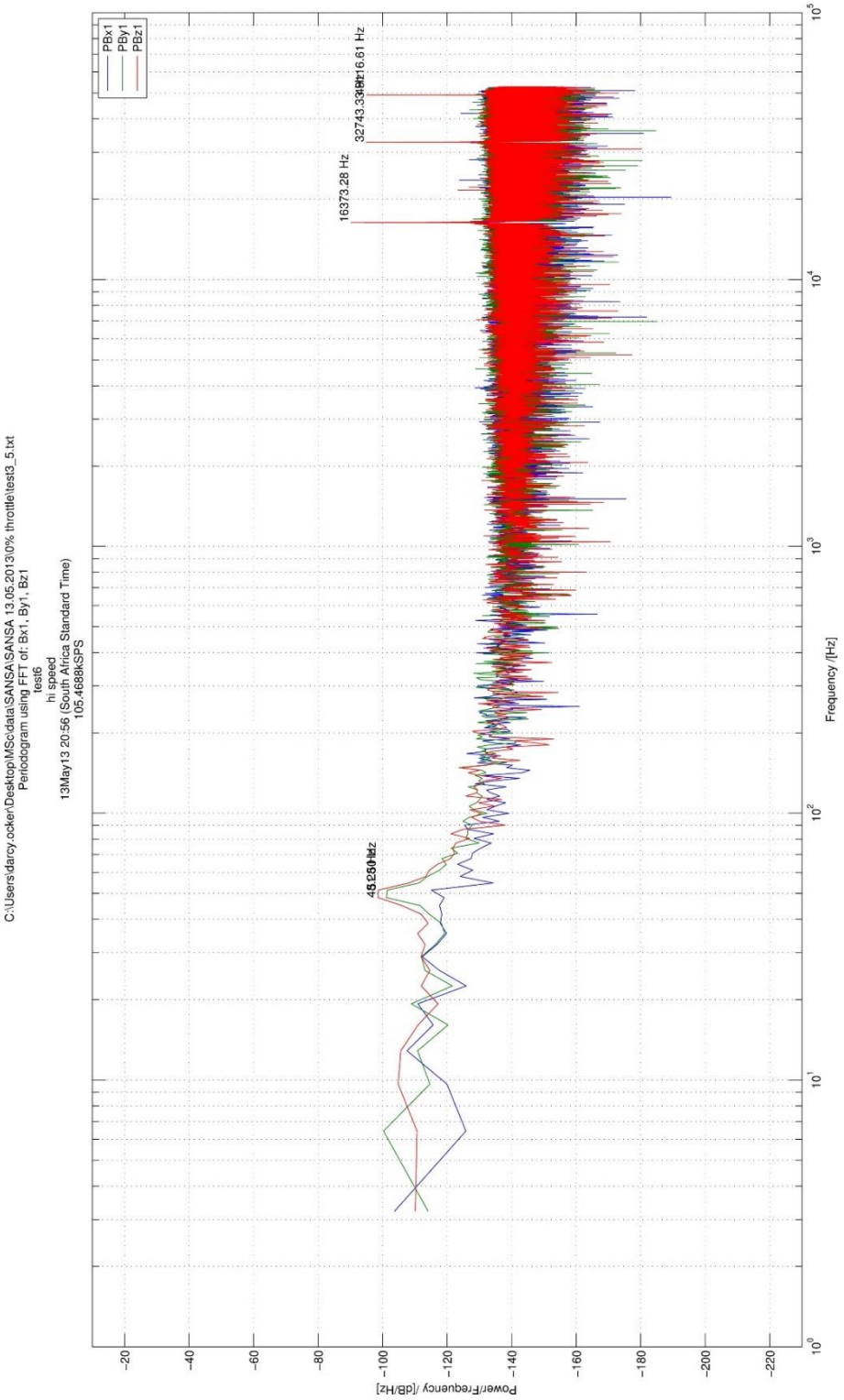


Figure 5-8: Spectral analysis (periodogram) of the noise for sensor 1 sampled at 105 kSPS showing the excitation frequency (16 kHz) and 1st (32 kHz) and 2nd (48 kHz) harmonics. Note the significant difference in magnitude between sensor 1 and 2 indicating that the devices may require calibration or indicate manufacturing differences between the units.

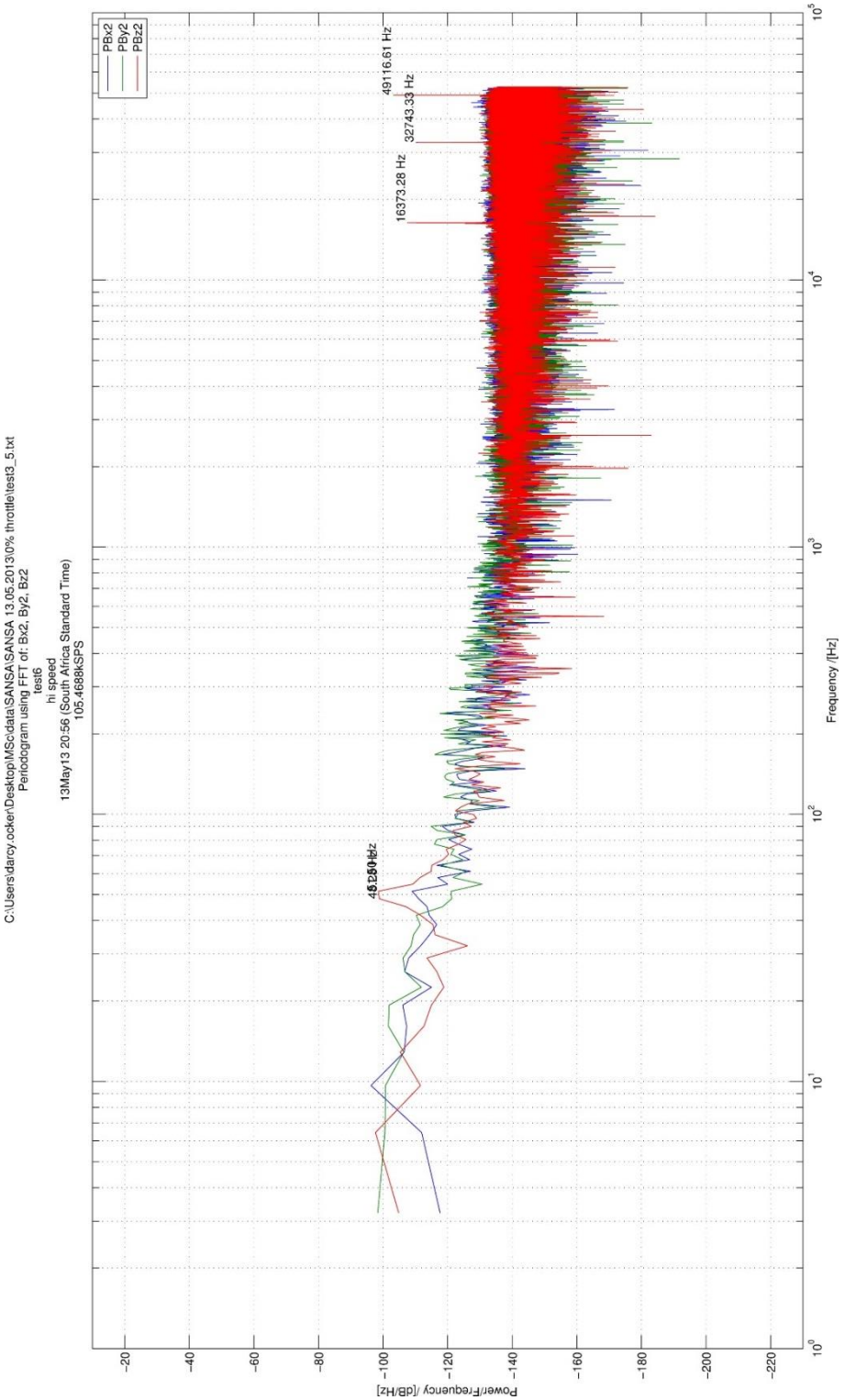


Figure 5-9: Spectral analysis (periodogram) of the noise for sensor 2 sampled at 105 kSPS showing the excitation frequency (16 kHz) and 1st (32 kHz) and 2nd (48 kHz) harmonics. Note the significant difference in magnitude between sensor 1 and 2 indicating that the devices may require calibration or indicate manufacturing differences between the units.

Chapter 6 – Prototype Datalogger: Design and Evaluation

6.1 Introduction

Following the success of the initial prototype, the focus shifted to that of creating a suitable payload for the chosen UAV platform. While the initial project considered the work of Versteeg *et al.* using a magnetometer with a VTOL rotary wing UAV [40], [41] the initial focus was only to construct the payload for such a platform and not a platform itself.

As time passed the requirements grew to building a platform for sufficient characterisation and integration. The challenges posed by such a project would only be sufficiently addressed if there was a UAV platform to test and integrate with. After considering the advantages and disadvantages along with requirements of a suitable UAV for an aeromagnetic survey it was decided to build a fixed wing UAV that did not necessitate the need for a runway i.e. a PTOL fixed wing UAV. These design criteria would go a long way as a proof of concept for potential future work.

The main reason for choosing a fixed wing UAV is that it would offer more endurance and it would be easier to construct using non-magnetic materials which in turn had an impact on the design of the prototype payload.

The focus of the payload became that of creating a datalogger system that would allow total magnetic intensity (TMI) data to be logged against a GPS position and time. The idea being that one could subtract the ambient geomagnetic field logged by another base station with similarly logged GPS position (stationary) and time. This would allow for minor variations

and fluctuations in the geomagnetic field to have a time record so as to not adversely affect the TMI results of the UAV aeromagnetic survey. Logging the TMI and performing the subtraction with the geomagnetic field should give an indication that some kind of geomagnetic anomaly is present in the area surveyed.

6.2 Prototype Datalogger Design Elements

To be able to successfully integrate such a data logging system onto a UAV the removal of the PC/laptop shown in Figure 5-1 above was fundamental to the payload. The idea came to interface the ADS 1278 EVM-PDK with a microcontroller that provided the necessary interfaces to log the output of the ADS 1278 EVM-PDK and GPS data to some form of non-volatile memory.

6.2.1 Choice of Microcontroller

The choice of microcontroller was based on good engineering practice and experience. The basis for the choice of the microcontroller platform was formed by the following criteria:

1. The platform should be well supported and offer a development board/kit
2. The platform should be fast and powerful
3. The platform should offer at least the required interfaces i.e.:
    a. SPI interface for the SD card
    b. Frame Sync interface
    c. UART interface for the GPS NMEA data
    d. Timer counter interface for the engine RPM interface
4. The development environment should be well supported and offer an easy to use framework
5. The platform should offer low power consumption
6. The platform should offer good value

Based on the above criteria the Atmel AT91SAM7X-EK evaluation board [54] was chosen. The evaluation platform based on the Atmel AT91SAM7X and AT91SAM7XC 32-bit RISC microcontrollers [55].

6.2.1.1 µTasker Operating System for the Atmel AT91SAM7X

To augment the functionality and expedite the development process of the prototype datalogger the µTasker Operating System (OS) [56] was chosen. This OS is well supported by its creator M.J. Butcher and offers good support for the types of interfaces and the microcontroller architecture chosen [57]. The µTasker OS framework integrates well with Microsoft Visual Studio (MSVS) and offers a user friendly C++ development environment and simulation facility to test and debug code.

6.2.2 Choice of GPS Receiver Module

The global positioning system (GPS) would be used as the timing and position reference for the datalogger. The GPS fix update would be used to update data and log the digitised magnetometer data relative to a GPS position and time. As such the GPS receiver module should have the following criteria:

1. The GPS receiver module should be well supported and use NMEA 0183 message format via a UART interface
2. The GPS receiver module should have an update rate of up to 10 Hz
3. The GPS receiver module should offer a small form factor and include an integrated antenna
4. The GPS receiver module should offer low power consumption
5. The GPS receiver module should offer high sensitivity
6. The GPS receiver module should offer good value

The above criteria were used to select the Fastrax UP501 GPS antenna module [58], [59].

6.2.3 DCBL Motor Speed Calculation

In the initial tests with the propulsion system characterisation (described in Chapter 8) it was noted that the electromagnetic noise (EM) from the DC brushless (DCBL) motor was related to the engine RPM. If this was known one could remove this noise at a later stage by post processing, therefore an interface to measure the engine RPM from the DCBL motor back electromotive force (EMF) was added. The engine RPM is related to the frequency of commutation of the DCBL motor in the following way [60]:

$$n = \frac{f * 120}{Z_{pol}} \qquad (6.1)$$

where:

1. n is motor shaft speed in RPM
2. f is the frequency of commutation in Hz
3. Z_{pol} is the number of magnetic poles for the motor

6.2.4 Prototype System Overview

These ideas and choices gave rise to the following block diagram shown in Figure 6-1 below.

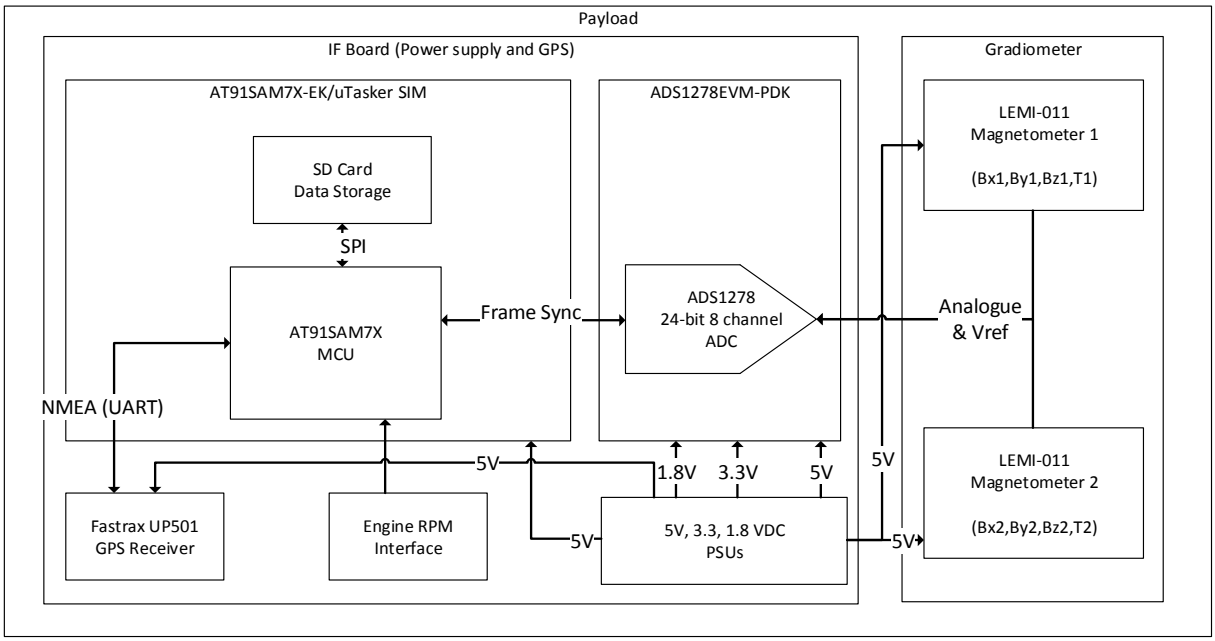


Figure 6-1: Prototype datalogger block diagram design.

6.3 Software Design

Once the hardware elements had been chosen, the design of the software for the microcontroller could begin. The idea was to use the GPS update fix as a trigger for sampling of the magnetometer data and engine speed. The GPS data would be checked for validity before performing any logging. This led to the following software design flowchart (see Figure 6-2 below):

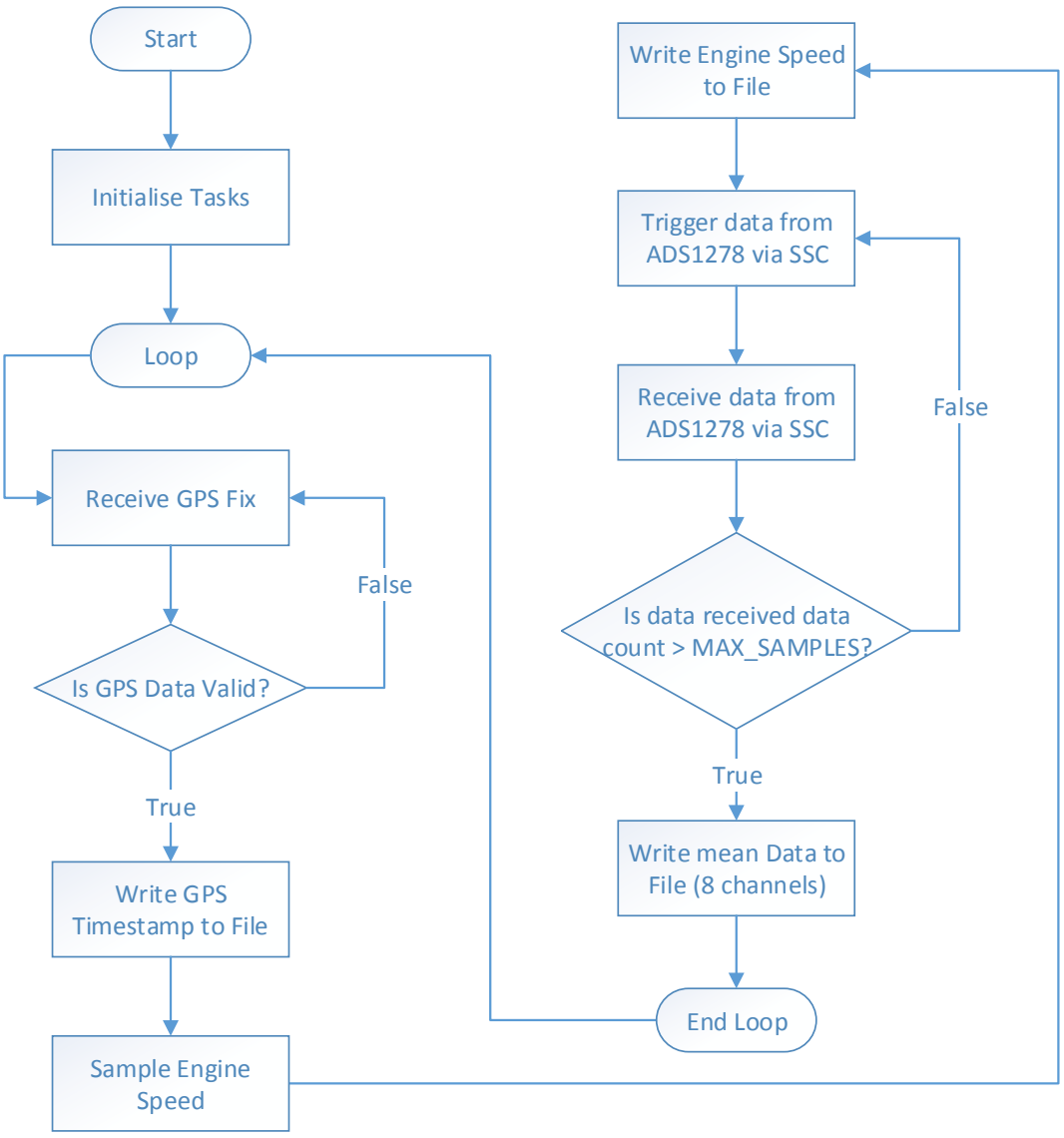


Figure 6-2: Datalogger software design flowchart.

6.4 Hardware Design

While the prototype datalogger elements were commercial-off-the-shelf (COTS) components, there were some modifications required. An interfacing (IF) printed circuit board (PCB) also needed to be produced. The approach was to create a stacked setup with the two modified Texas Instruments ADS1278 EVM-PDK and Atmel AT91SAM7X-EK boards interfacing via a central interface board that provided the associated power requirements and pinout routing (see Appendix E). The stacked approach would minimise the footprint of the prototype datalogger.

The interface board is also responsible for the low pass filtering of the back EMF from the DCBL motor and associated de-coupling from the LiPo batteries as the power supply to the datalogger. This was to enable the measurement of the DCBL motor's commutation frequency and conversion into an RPM measurement.

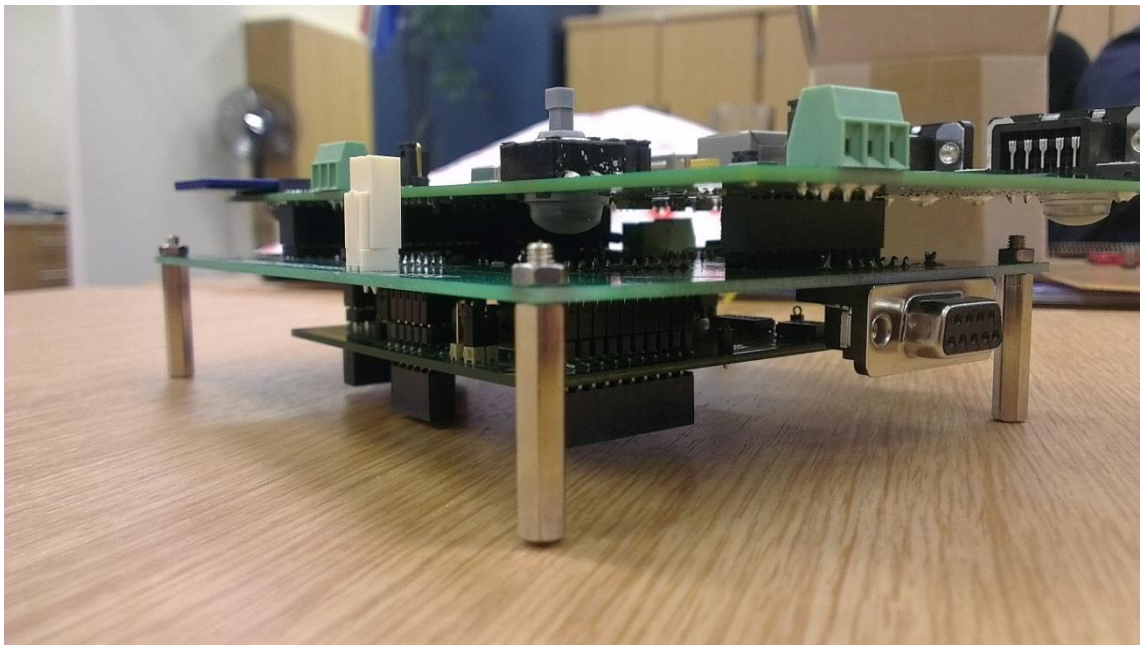


Figure 6-3: Compact prototype datalogger showing the 3 boards stacked on top of one another. The IF board is in the centre and the Atmel AT91SAM7X-EK board is located on top.

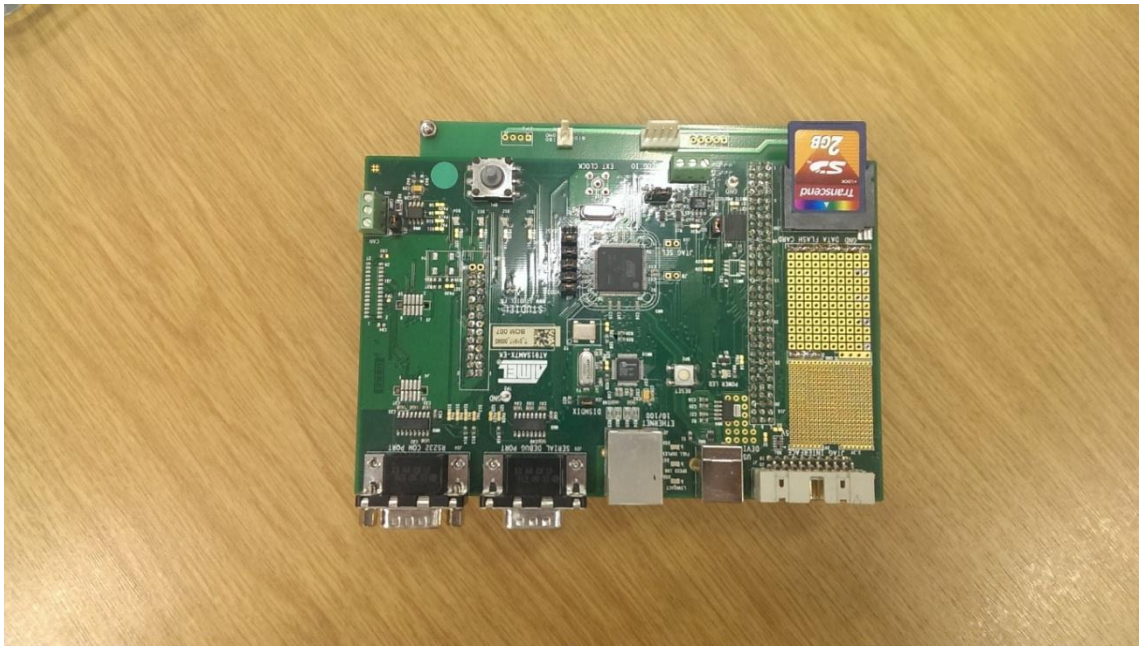


Figure 6-4: Prototype datalogger – top view. Note the Atmel AT91SAM7X-EK with 2 GB SD card attached.

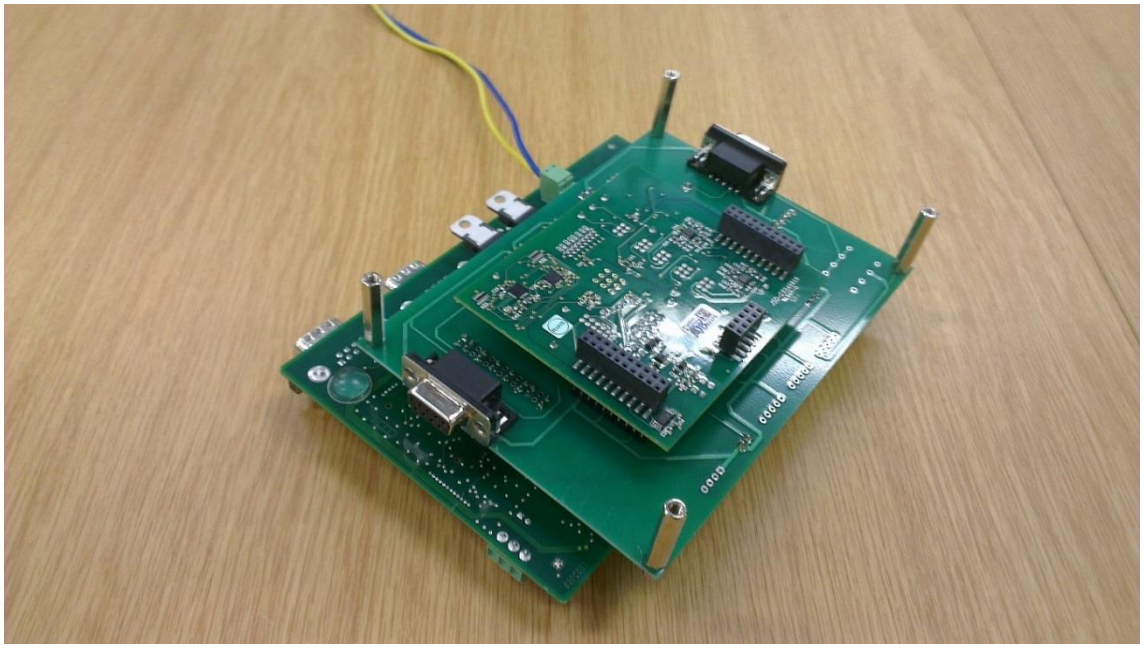


Figure 6-5: Prototype datalogger – bottom view. Note the ADS1278 ADC board on the bottom.

6.5 Prototype Datalogger Evaluation

The prototype datalogger proved to be adequate for collection of basic data but revealed some limitations in the chosen hardware.

While the microcontroller and the ADC could successfully be interfaced via the SCC interface, the data rates proved to be quite high. The high data rates coupled with the 1 Hz GPS fix rate cycle meant that only a limited amount of data (512 samples) could be read, averaged and written to the SD card in a cycle. This meant that not enough data could be collected to perform the post processing required to remove spectral artefacts associated with the environment and propulsion system.

Hence, the idea of recording the DCBL motor speed became an extra overhead in the logging chain that added no value and was therefore not implemented in the source code (see Appendix F). However, the provision for the filtering of the back-EMF and pin interfacing has been made on the interface (IF) board. Table 6-2 below shows an example of the output from the prototype datalogger. The GPS GGA message is written to file followed by a row of mean raw data (512 samples averaged) from the 8 ADC channels. These codes require a conversion to Volts and then Equation *(5.1)* can be applied to convert to nanoTesla. The data format is as follows:

Table 6-1: Digitised data format of raw data codes from the 4 outputs of each sensor.

Sensor 1 x data code	Sensor 1 y data code	Sensor 1 z data code	Sensor 1 temp. data code	Sensor 2 x data code	Sensor 2 y data code	Sensor 2 z data code	Sensor 2 temp. data code

Figure 6-6 below shows a picture of the oscilloscope output used to debug the microcontroller SCC interface to the ADC. Only channel 1 and 5 were enabled to simplify debugging. A Tektronix TDS 3024B oscilloscope was used to debug the interface.

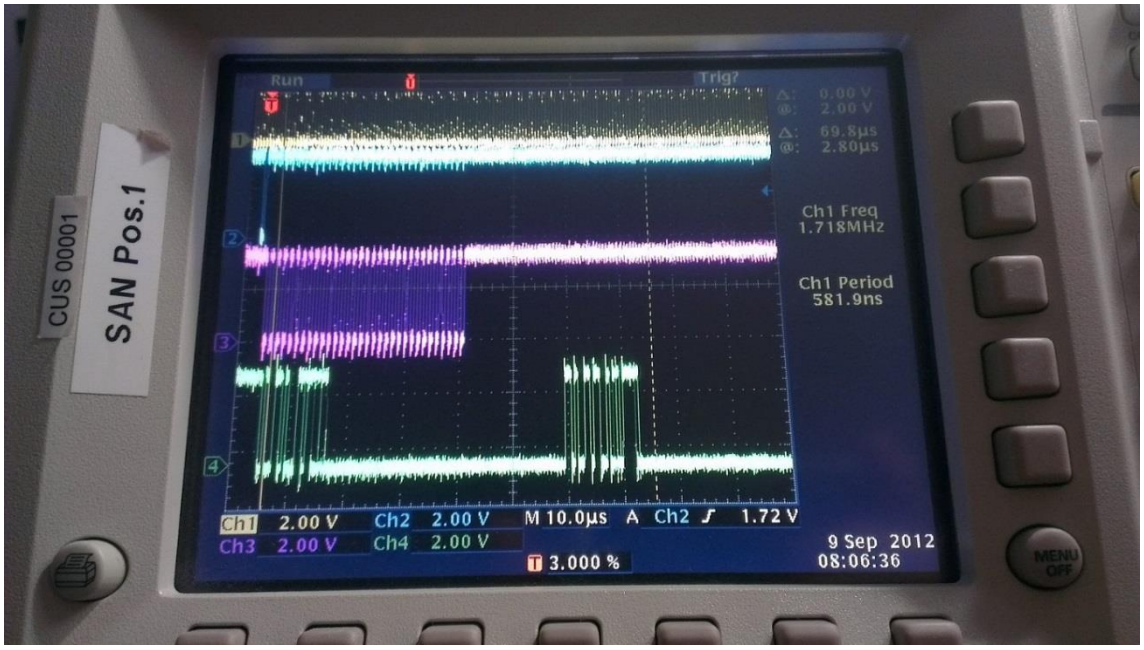


Figure 6-6: The oscilloscope output used to debug the microcontroller SCC interface to the ADS. The green trace shows the 192-bit data train from the ADC (only channel 1 and 5 enabled). The purple trace shows the dummy data (0xAAAAAAAA) being sent to the ADC and the yellow trace shows the 1.7 MHz clocking frequency.

Table 6-2: An extract of the data logged by the prototype datalogger. The GPS GGA message is logged once a second with averaged raw data codes (still requiring processing into a nanoTesla value) for each of the 8 channels output from the ADS1278.

$GPGGA,192914.000,3407.5995,S,01823.0987,E,1,4,4.05,65.4,M,32.5,M,,*7F

-2860239 445175 10806563826426-2962073 -520677 -1148248 3893791

$GPGGA,192915.000,3407.5982,S,01823.1011,E,1,4,4.05,64.6,M,32.5,M,,*7C

-2859396 445867 10814013826393-2961910 -520695 -1148193 3893779

$GPGGA,192916.000,3407.5957,S,01823.1080,E,1,4,4.05,62.7,M,32.5,M,,*78

-2860227 445237 10807703826379-2961929 -520750 -1148162 3893709

$GPGGA,192917.000,3407.5963,S,01823.1080,E,1,4,4.05,64.2,M,32.5,M,,*7D

-2859831 445526 10810643826412-2961941 -520689 -1148062 3893776

$GPGGA,192918.000,3407.5968,S,01823.1069,E,1,4,4.05,66.9,M,32.5,M,,*77

-2859381 445932 10816023826388-2961585 -520705 -1148024 3893786

$GPGGA,192919.000,3407.5969,S,01823.1075,E,1,4,4.06,68.5,M,32.5,M,,*7B

-2859469 445829 10815113826407-2961827 -520517 -1148031 3893748

$GPGGA,192920.000,3407.5964,S,01823.1073,E,1,4,4.05,69.3,M,32.5,M,,*7E

-2859702 445499 10809563826390-2961544 -520626 -1148179 3893722

$GPGGA,192921.000,3407.5953,S,01823.1083,E,1,4,4.05,68.2,M,32.5,M,,*74

-2860140 445136 10806923826389-2961936 -520605 -1148254 3893792

$GPGGA,192922.000,3407.5950,S,01823.1083,E,1,4,4.05,68.7,M,32.5,M,,*71

-2859829 445134 10806273826366-2961702 -520706 -1148266 3893756

$GPGGA,192923.000,3407.5948,S,01823.1079,E,1,4,4.05,68.6,M,32.5,M,,*7D

-2859350 445543 10811483826404-2961434 -520469 -1148135 3893725

$GPGGA,192924.000,3407.5944,S,01823.1076,E,1,4,4.05,66.2,M,32.5,M,,*73

-2859503 445934 10813163826389-2961985 -520563 -1148107 3893764

Chapter 7 – PTOL UAV Design, Construction and Evaluation

7.1 Introduction

During the conceptualisation phase of this project it was not envisaged that a UAV platform would be required in the scope of this MSc. dissertation. However, during the later phases it became apparent due to the nature of tight integration requirements between platform and payload it would be necessary to consider a UAV platform for this project.

For most of the project's timespan, only vertical-take-off-and-land (VTOL) UAVs were considered. The general idea being that a VTOL UAV platform would allow for the most generic real-world terrain scenarios to be covered and largely driven by the work conducted by Versteeg *et al.* [40] and McKay *et al.* [41].

7.2 Design

After reviewing various platforms for real-world scenarios it became apparent that point-take-off-and-land (PTOL) UAVs should be considered. These types of UAV were well suited to the current payload design and a simplified real-world operating scenarios that were envisioned. Also the mechanical lift created by a fixed wing craft would instantly provide an advantage in achieving the platform design parameters in Chapter 4 above.

Considering the work of Wagner *et al.* [5] for hand launchable UAVs, the parameters that become a design guideline for the UAV platform were:

1. Portability
   The UAV and associated system should be easily transportable by 1 or more people.

2. Launch and recovery:
   The UAV should not require a runway and should allow launch and recovery from a single point.

3. Endurance:
   The UAV should offer at least one hour flight endurance.

4. Stability:
   The UAV should offer maximum stability.

5. Durability:
   The UAV should be durable enough to stand up to flight trials.

6. Construction material:
   The UAV should be made from non-magnetic material where possible to minimise magnetic effects from the platform.

7.2.1 Choice of platform

Due to the time, cost and durability constraints the largest COTS RC platform was sought out. The design by Windrider [61] of their *Queen Bee* platform was a good candidate for a base platform. The construction material of the *Queen Bee* is extruded polypropylene (EPP) which has the advantages of being strong, durable, light and non-magnetic.

The *Queen Bee* design offers a 2.54 m (100 inch) wingspan and a wing area of approximately 0.85 m^2 [62]. The advantage of choosing a delta wing/flying wing design is that it offers a two channel control surface interface via the port and starboard elevons (elevators and ailerons combination). This was particularly beneficial during construction as only two control surface wiring looms had to be run to the servo motors. The *Ardupilot* autopilot provides for this control surface implementation [35] and performs the RC elevator and aileron signal mixing.

Figure 7-1: Windrider *Queen Bee* EPP delta wing after assembly. At this stage still awaiting the fuselage and control surface actuation mechanicals.

7.2.2 Fuselage Construction

While the *Queen Bee* provides a useable base platform, it does not allow for mounting a propulsion system or any associated payload and sensors. This meant a sturdy lightweight, non-magnetic fuselage needed to be constructed to house these components.

Using the RC hobbyist groups [37], [35], [63] as well as RC hobbyist as forum it was decided to construct a carbon composite fuselage to house and mount the propulsion system (DCBL motor and high capacity lithium ion polymer (LiPo) batteries) and associated payload. Supplementary photographs of the fuselage construction can be seen in Appendix C.

7.2.3 UAV Payload

The UAV payload is made up of the following elements:

1. Up to three Hyperion VX G3 4S (14.8 V) 5000 mAh LiPo batteries[3] [39]
2. One Hyperion ZS 3025 10 turn 775 KV DCBL motor (14 pole) [64]
3. One Castle Creations Phoenix ICE 100 A ESC [65]
4. Ardupilot Mega 2.5 autopilot with MediaTek MT3329 GPS and 3DR 433 MHz radio telemetry "Air" module [35]
5. AttoPilot Power Sense Kit [35]
6. Airspeed Kit [35]
7. Power distribution board for up to 3 LiPo batteries with current and voltage monitoring
8. Prototype datalogger with Fastrax UP501 GPS receiver module
9. Two LEMI-011b fluxgate magnetometers

Items 1 through 8 are housed within the fuselage.

[3] Only two batteries were implemented. This was due to the high individual cost of each of the batteries.

A comment on KV rating or "revolutions per minute (RPM) constant" for an electric motor must be made. The KV value refers to the RPM of an electric motor when 1 Volt (V) is applied. Sometimes the RPM constant is also written as Kv or kv but should not be confused with kV i.e. kilo-Volt [66]. This value is usually used to assess the torque level of a motor. A lower KV value indicates a high torque motor and vice versa. The value also gives insight into the construction of the motor including number of windings and magnetic materials used.

The choice of ESC and various other UAV components was made by using [63] and [37] as a guide.

7.2.4 Ground station

To facilitate the flight operations of the UAV a ground station was setup. This included:

1. A laptop running the APM Mission Planner Software [35] (See Figure 7-2 below)
2. A 3DR 433 MHz radio telemetry "air" module [35]
3. A JR XG8 2.4GHz radio control system [67]
4. Winch launch system (See Figure 7-3 and Figure 7-4 below)

Due to the awkward/dangerous positioning of the UAV's propeller along with the high take-off weight of the UAV it proved difficult to hand launch the UAV. After an unsuccessful attempt to hand launch was made a solution was required to facilitate a safe and successful launch of the UAV. A foot operated winch launch system was employed for this purpose (see Figure 7-5 below).

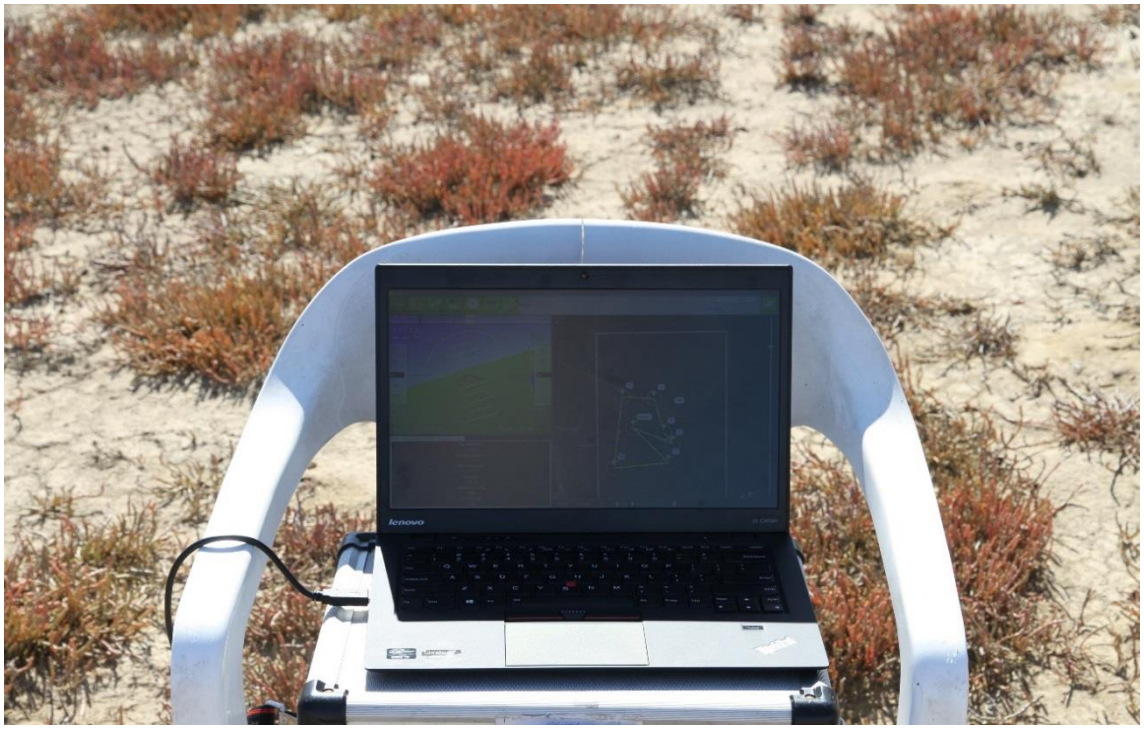


Figure 7-2: Ground station laptop running the APM Mission Planner Software.

Figure 7-3: Winch drum with foot operated switch and a 12 V automotive battery.

Figure 7-4: Winch pulley and winch line returning to the launch point.

Figure 7-5: Two person launch procedure with foot operated winch launch system. Note the parachute at the end of the winch line to act as a drag once the winch is de-activated. This allows for the successful decoupling of the winch line from the UAV launch hook.

7.3 System Costs

The costs of the various system components are listed below. The costs are split between the UAV and ground station combined and the costs of the prototype datalogger.

7.3.1 UAV Platform Costs

The UAV platform costs are tabulated below:

Table 7-1: UAV costs

Prices correct at time of purchase (2012)

Description	Unit Price	Quantity	Total
Windrider *Queen Bee* (with servos motors)	R 2 750,00	1	R 2 750,00
Fuselage Construction	R 2 000,00	1	R 2 000,00
Ardupilot with GPS	R 2 000,00	1	R 2 000,00
3DR Radio telemetry kit	R 850,00	1	R 850,00
Hyperion LiPo 4S Battery	R 1 500,00	2	R 3 000,00
Hyperion DCBL Motor	R 750,00	1	R 750,00
JR XG8 Radio System	R 4 740,00	1	R 4 740,00
Castle Creations 100 A ESC	R 1 500,00	1	R 1 500,00
Airspeed Kit	R 275,00	1	R 275,00
Atto Pilot Power Sense Kit	R 225,00	1	R 225,00
Winch Launch System (borrowed)	R 0,00	1	R 0,00
Consumables	R 800,00	1	R 800,00
			R 18 890,00

The initial budget set of R 15 000,00 (see Chapter 4) was exceeded by R 3 890,00.

7.3.2 Prototype Datalogger Costs

The prototype datalogger costs are tabulated below:

Table 7-2: Datalogger costs

Prices correct at time of purchase (2011)

Description	Unit Price	Quantity	Total
LEMI-011b Magnetometer (on loan from SANSA)	R 5 000,00	2	R 10 000,00
ADS1278 EVM-PDK (Free Sample)	R 0,00	1	R 0,00
Atmel AT91SAM7X-EK	R 2 810,00	1	R 2 810,00
Fastrax UP501 GPS Receiver Module	R 560,00	1	R 560,00
Interface Board	R 1 500,00	1	R 1 500,00
			R 14 870,00

The prototype datalogger came in under the initial budget set of R 15 000,00 although the ADS1278-EVM-PDK would normally have a price associated with it of R1 500,00 or more.

7.4 UAV Flight Trials and Evaluation

The flight trials of the UAV were conducted using the payload as described in 7.2.3 with the exception of the LEMI-011b magnetometers. The decision not to mount the magnetometers was the risk of damage in the event of an accident.

The take-off weight of the UAV with payload was around 5 kg and offered a flight time in excess of 30 min using 50% of the battery capacity (10 Ah). This provided a theoretical flight time of 1 hour as planned, but in practice the LiPo batteries should not be drained below 20% capacity as this could cause them damage. With the addition of a third battery, as designed, this would bring the flight time to well over an hour. The reason for not implementing the third battery was cost.

The flight trials were conducted at the Cape Radio Flyers Club Site at Rietvlei, Milnerton, Cape Town (GPS co-ordinates 33.852156 S; 18.49743 E) during the month of April 2013 (14th, 21st and 26th). The flight trial weather conditions were with a light westerly breeze, 25 °C and sunny. Launch and landings were into the light westerly breeze.

7.4.1 Ardupilot APM Evaluation

The autopilot function was tested for stabilised flight and waypoint navigation. The stabilised flight functioned as expected, keeping the UAV stabilised while allowing operator inputs for direction and speed changes.

Waypoint navigation proved to be problematic and required some tuning to the system, especially with regards to speed choices. For instance, if the next waypoint was far away the autopilot would try to get there as quick as possible which would cause the UAV to operate at maximum throttle. This was not conducive to extending battery life and hence endurance. The Ardupilot does allow for system tuning and setting of throttle inputs. Another work around to this would be to add more way points and have them closer together. These values were tuned but were not flight tested.

One other notable problem with the autopilot was that the altitude calculation proved to be temperamental in some instances. The altitude seemed to "jump" from the initial barometric sensor altitude calculation when first acquiring a GPS fix and creating a bad altitude offset. This was particularly evident when the first GPS fix took a long time to acquire. The reason for this is that the initial GPS fix is very inaccurate and these inaccuracies are exacerbated when using GPS as an altitude sensor. This was further investigated and found to be a bug with the Ardupilot APM firmware which has been fixed in the later versions (from V2.7 and higher) of the firmware. The version 2.7 of firmware was downloaded but unfortunately has not been flight tested due to inclement weather with the onset of the Cape winter.

7.4.2 Launch and Recovery Evaluation

The launch and landing of the UAV was done by an operator and in the case of launch the winch launch system worked very well. Landing was achieved by simply landing in an open area of scrub at the flight trial site. The landing did not create any damage although there was a risk the propeller and/or motor could be damaged in the event of an unlucky landing. This is something that should be addressed in future modifications to the platform.

7.5 Conclusion

Despite some setbacks the UAV design proved robust enough to undertake multiple launches and landings successfully. The decision to use COTS components and systems was justified as it aided in the developments of the UAV platform and offered reasonable performance for the costs incurred.

In conclusion the UAV platform performed very well and proved to be an adequate platform to perform further testing with the magnetometers onboard.

Figure 7-6: Initial flight trials – Mid-flight.

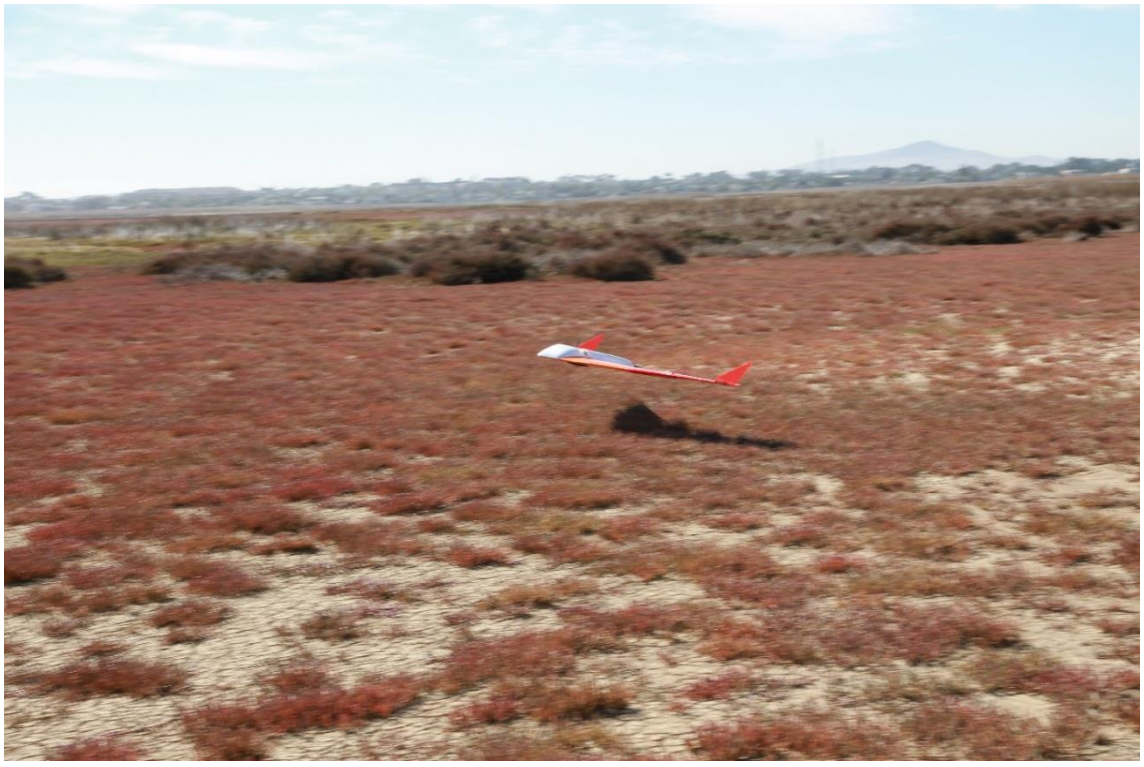


Figure 7-7: Initial flight trials – Coming into land.

Figure 7-8: Evaluation of flight trial data.

Chapter 8 – Propulsion System Characterisation

8.1 Introduction

To verify the effects of distance and speed of the DCBL motor on the magnetometers some tests were devised using two LEMI-011b fluxgate magnetometers and the initial prototype system outlined in Chapter 5 above. The tests were conducted with a test propulsion system that was similar to the UAV propulsion system outlined in Chapter 7 above.

The propulsion system comprised the following elements (see Figure 8-1 below):

1. E-Flite Power 25 870 KV DCBL Motor (14 pole)
2. Hobby wing Flyfun 60 A ESC
3. Turnigy 3S (11.1 V) 2200 mAh Battery
4. APC 8x8 propeller

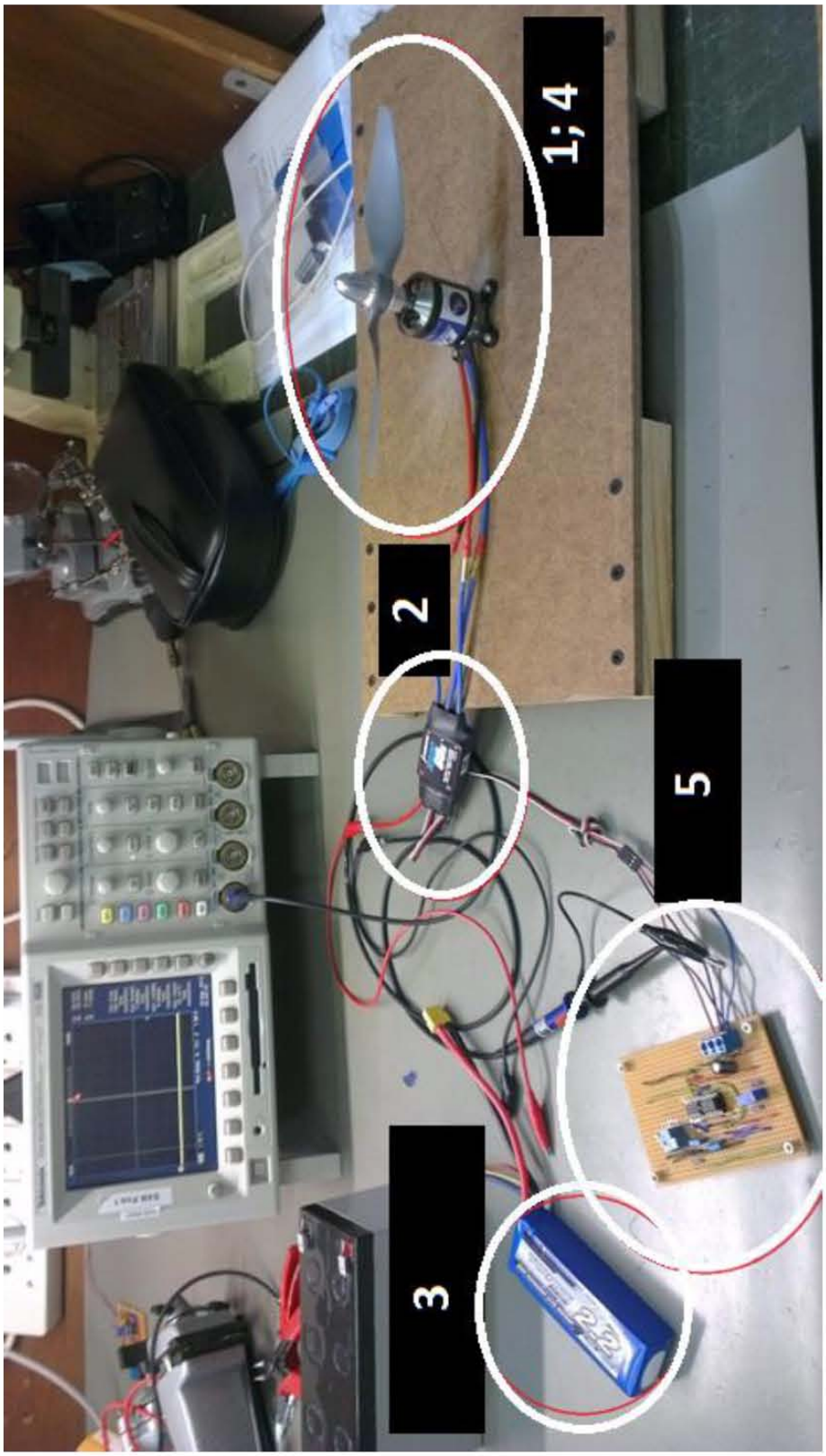


Figure 8-1: Initial propulsion system used for characterisation showing the 2200 mAh LiPo battery (3), 60 A ESC (2), DCBL motor (1) with propeller (4) mounted and the made up PWM controller (5) to control the 'throttle' to the ESC PWM input.

8.2 Initial Test Setup

The tests were set up to measure the magnetic signature of the DCBL motor at two different distances (500 mm and 1100 mm) and to show the impact of varying the distance of sensor to noise source. The LEMI-011b fluxgate magnetometers were used to measure these effects, and readings were taken for five different motor speeds (0 RPM, 2140 RPM, 4280 RPM, 6420RPM and 10700 RPM) at both distances and compared. The above motor speeds represent a 0 % to 100 % throttle opening.

To vary the motor speed a pulse width modulation (PWM) controller was made up using two 555-timers and a potentiometer (see Appendix D). The PWM controller was an input to the ESC which created the 'throttle control' for the motor.

A simple RC low pass filter was constructed (see Appendix D) and used to measure the motor commutation frequency using the back EMF of two of the motor phases. This was measured by a portable oscilloscope (Fluke 199C Scopemeter Color) and converted to a motor shaft speed in RPM.

Figure 8-2 below shows the test setup. Note the orthogonal axes of the magnetometers shown. The blue arrow shows the direction into the picture and represents the *x*-axis. The green arrow shows the direction of the *y*-axis. Note that the *z*-axis (shown in red) points down.

The magnetometer on the left of the picture is sensor 2. Sensor 1 is on the right. The *y*-axis for both sensors is aligned in the magnetic east-west direction. This means that the *y*-axis should "see" the least geomagnetic field strength.

The tests were carried out at SANSA Space Science, Hermanus, on the 21st November 2012.

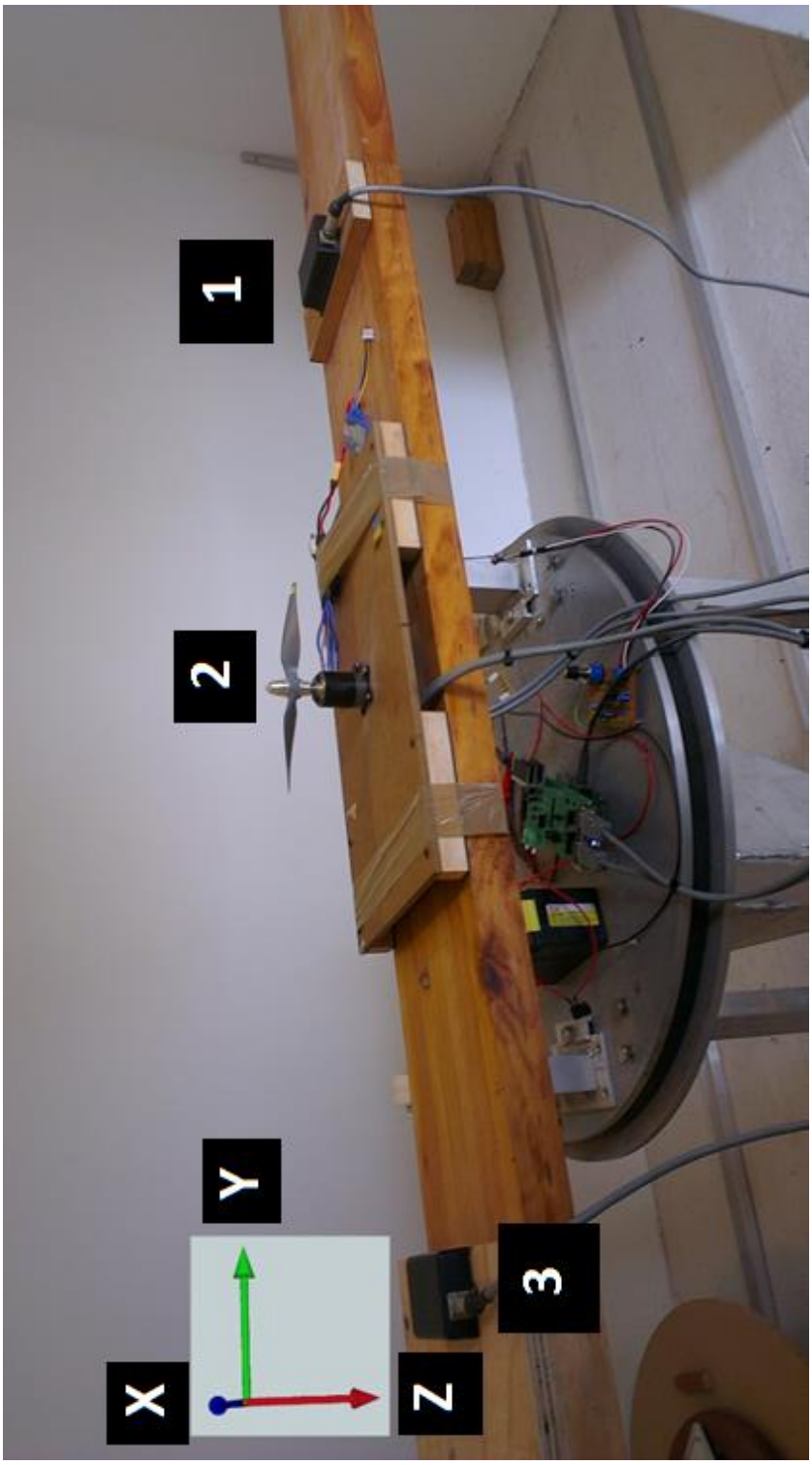


Figure 8-2: Initial characterisation tests of a DCBL motor propulsion system (2) with magnetometer (1) axes indicated. x–axis (shown in blue), y-axis(shown in green) and z-axis(shown in red). Note that the positive z-axis (red) points down (towards the ground) and sensor 2 (3) is on the left and sensor 1 (1) on the right in the picture. Magnetometers spaced 500 mm from the DCBL motor in this picture.

Figure 8-3: The Fluke 199C "Scopemeter Color" oscilloscope showing the filtered (using the 2nd order RC filter shown in Appendix D) commutation frequency (263.9 Hz) of the DCBL motor. This was measured between 2 phases on the ESC and converted to a DCBL motor shaft speed (2262 RPM).

8.3 Explanation of the data processing techniques and evaluation thereof

In the MATLAB code found in Appendix G a simple recursive notch filter was implemented using the iirnotch [68] function in the MATLAB 2009a DSP Toolbox. The recursive filter(s) is (are) 'dynamically' calculated based on the findpeaks algorithm [69] which detects peaks in the frequency domain [70], [71] based on a threshold calculated on the average level of the values from DC to the cut-off frequency (f_c) input [72] by the user (default is set to 15 Hz). The user also specifies the bandwidth (BW) [68] of the notch filter (default is set to 0.01 Hz).

The 'ringing' in the time domain data due to the impulse response of the notch filter(s) is (are) removed by 'chopping' off the first few samples of the calculated filtered noise data in the time domain. The amount of 'chopped' data is specified by the user (m) and is a value between 0 (0 %) and 1 (100%). The more filters there are, the more 'ringing' in the data. This is usually due to the higher sample rate being able to 'see' more frequencies (i.e. excitation frequency and subsequent harmonics of the magnetometers). To avoid this, the lower sample rate (10 kSPS) on the ADC was preferred.

To investigate the 'improvement in noise', the standard deviation in the noise data were used to offer a quantitative comparison and used to compare to the LEMI-11b specification. This was calculated by subtracting the mean of the data samples for each axis from the data. This put the three orthogonal axes measurements on the same 'zero axis'. These data were then processed with the MATLAB code described above and the processed/filtered result's standard deviation was compared to the pre-processed data.

The idea being that the entire system should be able to be compared to the standard performance specification of the LEMI-11b magnetometers when digitised and mounted on the UAV platform.

8.4 Initial Test Results

From the figures and plots below, a clear correlation of how the distance of sensor to noise source and speed of the DCBL motor affected the results. What was clearly noticed was that one could attribute a single spectral line to the motor speed and this varied with throttle input. This spectral line associated with the DCBL motor was clearly orientated in the *x* and *y* directions of the magnetometer, while the 50 Hz AC mains line was orientated in the *z* direction (perpendicular to the ground) and shows the noise source being the 50 Hz from the electrical power utility.

As mentioned in Chapter 5.2.1 above, the 50 Hz tone can be attributed to the electrical power utility and is ever present. It was also surprising how strong the noise source was in the data despite the LEMI-011b fluxgate magnetometers having a built-in low-pass 10 Hz cut-off filter. It was subsequently noted that magnetic signatures associated with common ore bodies are usually found in the DC frequency range. An explanation of how the data is processed to remove the AC tones is given in Chapter 8.3 above.

The motor shaft speed seen as a 36 Hz (2160 RPM) tone was compared to that of the commutation frequency measured on the oscilloscope by rearranging equation *(6.1)* above.

The DCBL motor has 14 poles therefore:

$$f = \frac{36 * 60 * 14}{120} = 252\ Hz \qquad (8.1)$$

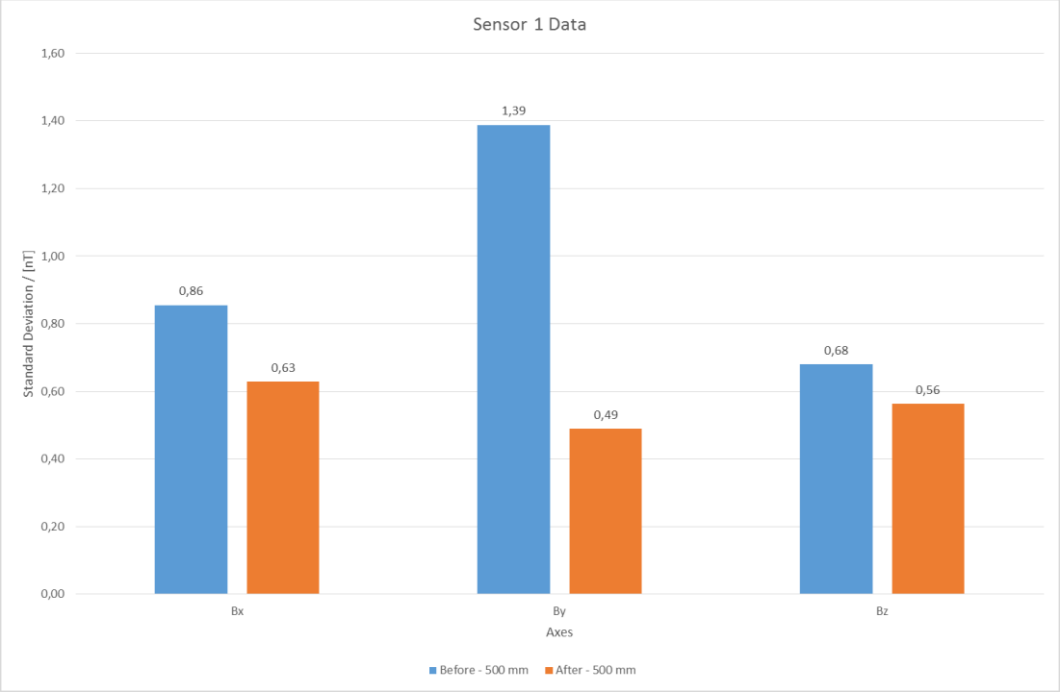


Figure 8-4: Summary of Sensor 1 data for the motor shaft speed at 36 Hz. The Standard deviation was used as a measure of improvement of the pre-processed ('before') and post-processed ('after') data.

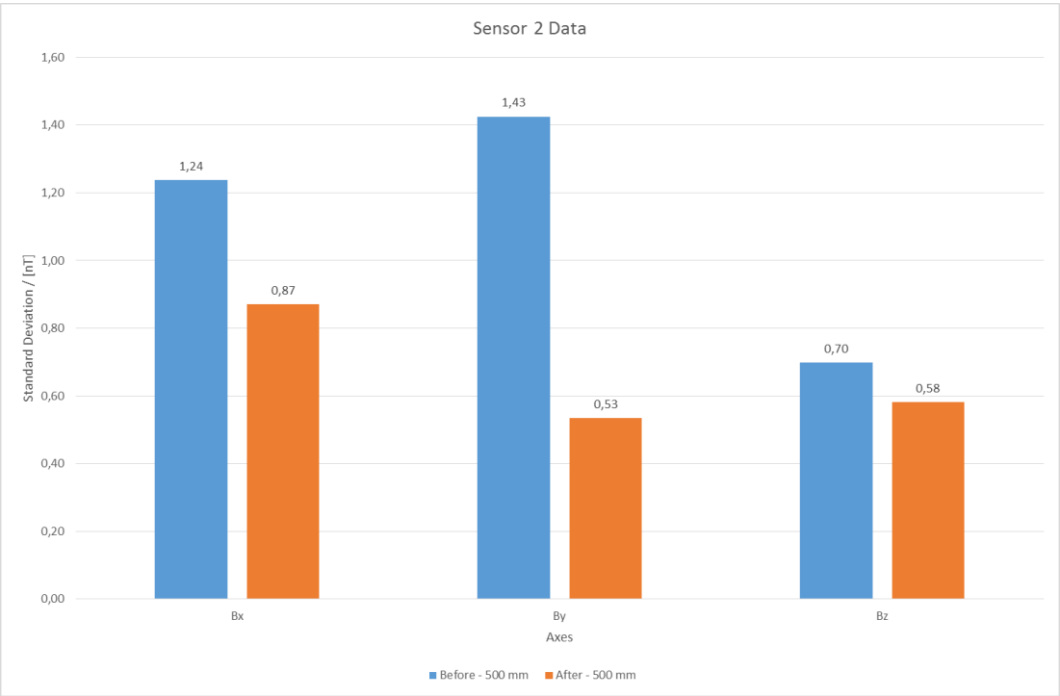


Figure 8-5: Summary of Sensor 2 data for the motor shaft speed at 36 Hz. The Standard deviation was used as a measure of improvement of the pre-processed ('before') and post-processed ('after') data.

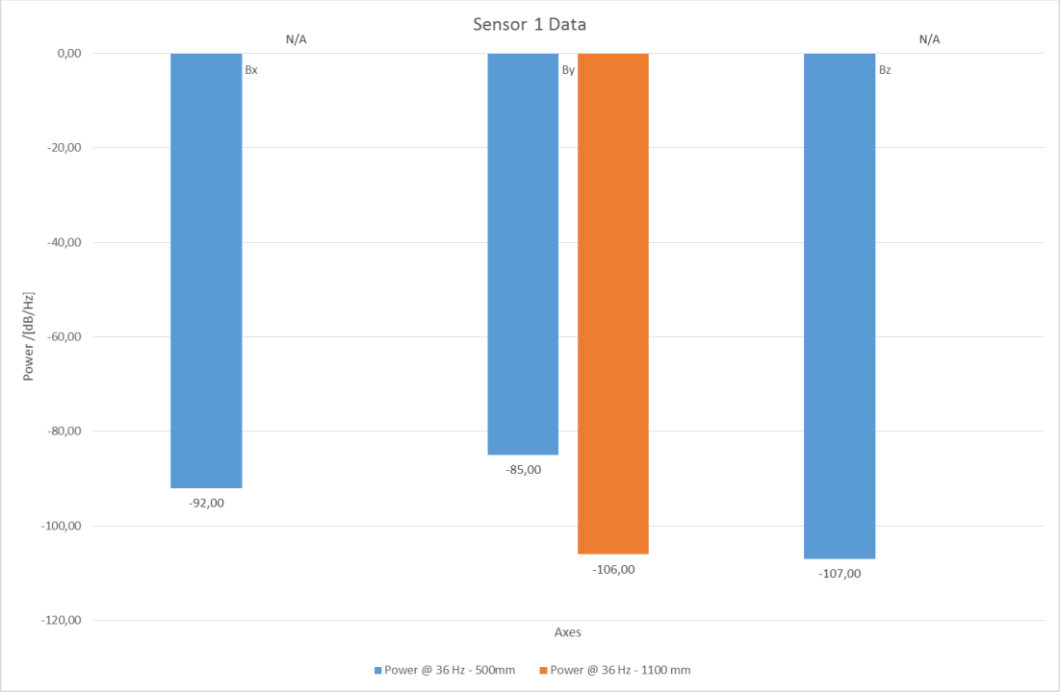


Figure 8-6: Summary of Sensor 1 data for the motor shaft speed at 36 Hz. The graph shows the power in dB/Hz of the 36 Hz tone at the two distances (500 mm and 1100 mm) for which the experiment was performed. The N/A values were not calculated as they were below the threshold set for the data but a 20 dB/Hz difference is shown for the y-axis by doubling the distance.

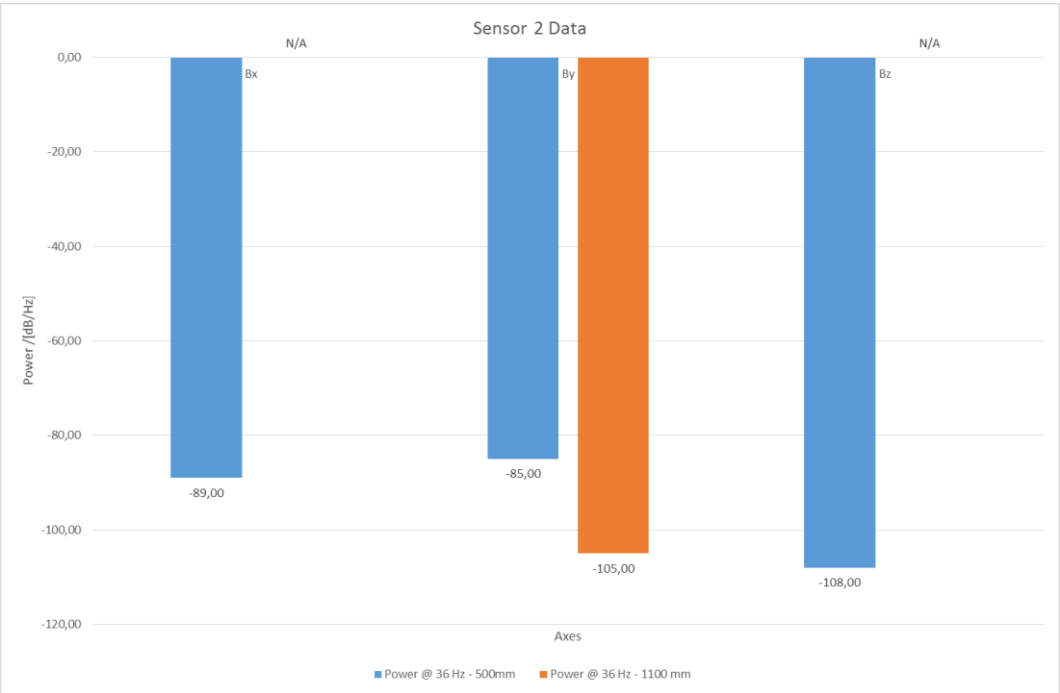


Figure 8-7: Summary of Sensor 2 data for the motor shaft speed at 36 Hz. The graph shows the power in dB/Hz of the 36 Hz tone at the two distances (500 mm and 1100 mm) for which the experiment was performed. The N/A values were not calculated as they were below the threshold set for the data but a 20 dB/Hz difference is shown for the y-axis by doubling the distance.

8.5 UAV Platform Test Setup

Buoyed by the initial test results it became clear that it would be possible to combine the fluxgate magnetometers on a platform with a similar propulsion system without negatively affecting the performance of the magnetometers provided the construction of the UAV employed the correct construction materials and techniques.

Once the platform was constructed and tested, similar propulsion system characterisation tests shown in section 8.2 above were carried out with the UAV and associated payload (see section 7.2.3 above).

Figure 8-8 below shows the test setup with the UAV securely[4] mounted to the test surface with the magnetometers mounted on the UAV. The starboard side of the UAV (wrapped in green) was used to mount sensor 2 and the port side (wrapped in red) was used to mount sensor 1.

Great care was taken to mount the two sensors so that the *y*-axis of each sensor was parallel to the DCBL motor shaft, since the motor shaft is mounted to create an offset thrust line to avoid torque roll of the UAV. This meant the sensors had to be mounted on an imaginary line perpendicular to the motor shaft, which created an asymmetry when mounted on the UAV.

Note the direction of the axes for the two sensors: the *x*-axis for both sensors points toward the respective wing tips of the UAV. The *y*-axis of the port side sensor (sensor 1) points in the forward direction of the UAV while the starboard sensor (sensor 2) points toward the rear of the UAV and both sensors have their *z*-axis pointing down. The axes are shown in Figure 8-9

[4] Note that the wing tips of the UAV had not been securely fastened. This led to slight flapping movements during the large throttle opening tests caused by turbulent wind flow in the experiment room. See Chapter 8.6.1 for more detail.

below, with the usual colour conventions (*x*-axis = blue, *y*-axis = green and *z*-axis = red). The magnetometers were each mounted 93 cm away from the DCBL motor shaft. This distance was chosen as it was as far away from the DCBL motor as practically possible.

The tests were carried out at SANSA Space Science, Hermanus, on the 14th of May 2013.

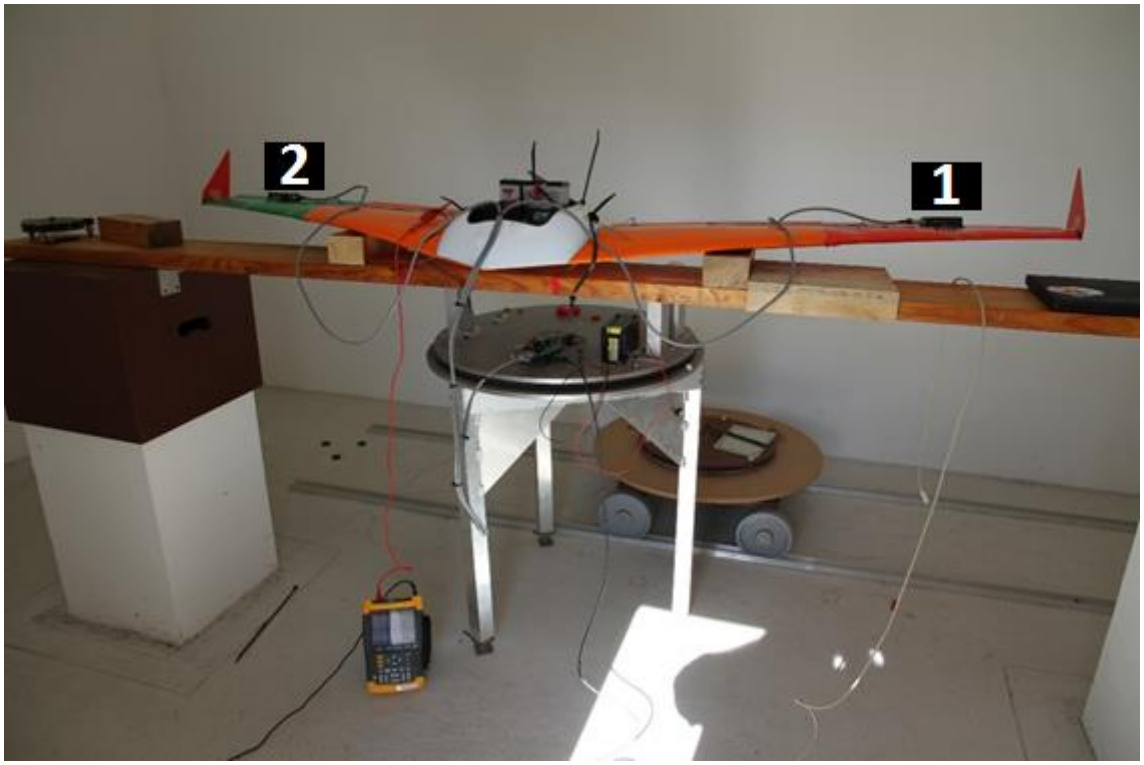


Figure 8-8: Propulsion system characterisation with UAV showing sensor 1 (1) mounted on the port side (red) and sensor 2 (2) on the starboard (green) side of the UAV.

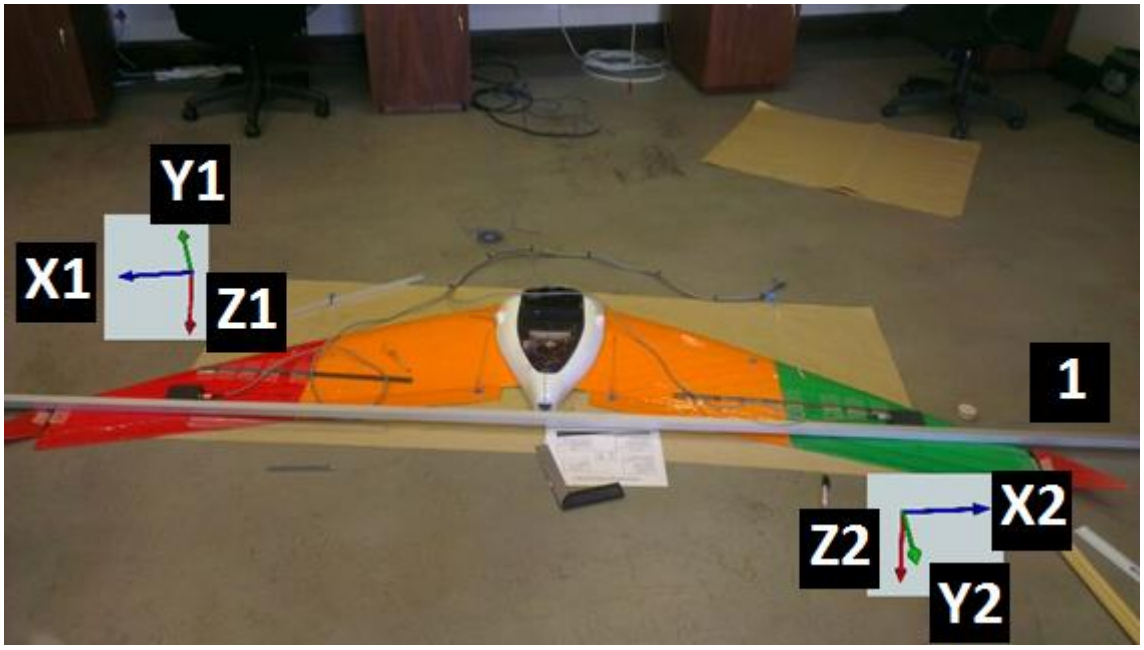


Figure 8-9: Mounting the sensors parallel to the DCBL motor shaft. Note the offset aluminium guide rail (1) to help achieve this. The guide rail was mounted perpendicular to the DCBL motor shaft. The orthogonal axes are shown using the usual colour convention for this paper for each sensor.

8.6 UAV Platform Test Results

A summarised view of the two sensor data sets is presented in Figure 8-10 and Figure 8-11 below. The standard deviation values are compared on the 'before' and 'after' data. The standard deviation on the noise pre and post notch filtering shows the improvement in noise using this filtering techniques to remove the AC tones. The high values seen in the large (75 % – 100 %) throttle openings are explained in chapter 8.7 below. A full explanation of how the data is processed to remove the AC tones is given in Chapter 8.3 above.

A comprehensive representation of the pre- and post-processed data for these experiments are given in Appendix I.

8.6.1 Explanation of the effects of the UAV 'wing tip movements' on the data.

A comment must be made on the high speed (large throttle openings) tests which caused serendipitous wing movements during the time of the data recordings. This movement was attributed to the turbulent airflow in the test room which caused the UAV wing tips, on which the magnetometers were mounted, to move about.

This caused a kind of 'slew' in the in the noise data (see the results in the 100 % throttle: Figure I - 37 through Figure I - 42 in Appendix I) on each of the orthogonal axes, which contributed to a large standard deviation (see Figure 8-10 and Figure 8-11 below) in the noise, before and after the data were processed.

As this is a real concern for practical application of the system, one would have to deal with this to successfully remove this kind of phenomena. The suggestion would be to try and measure this movement of the magnetometer in real-time (perhaps some kind of IMU) and compensate for this using other transformation techniques on the data. If this were possible, it would allow the two magnetometer's data to be used as a possible gradiometer to get gradient information relating to the localised magnetic field which would help in the intended purpose of mineral exploration.

It must be said that if the tri-axial data from the magnetometer is used to calculate the TMI, this value can be used without fear of the data being corrupted. This is a practical way of 'removing' these ill-effects.

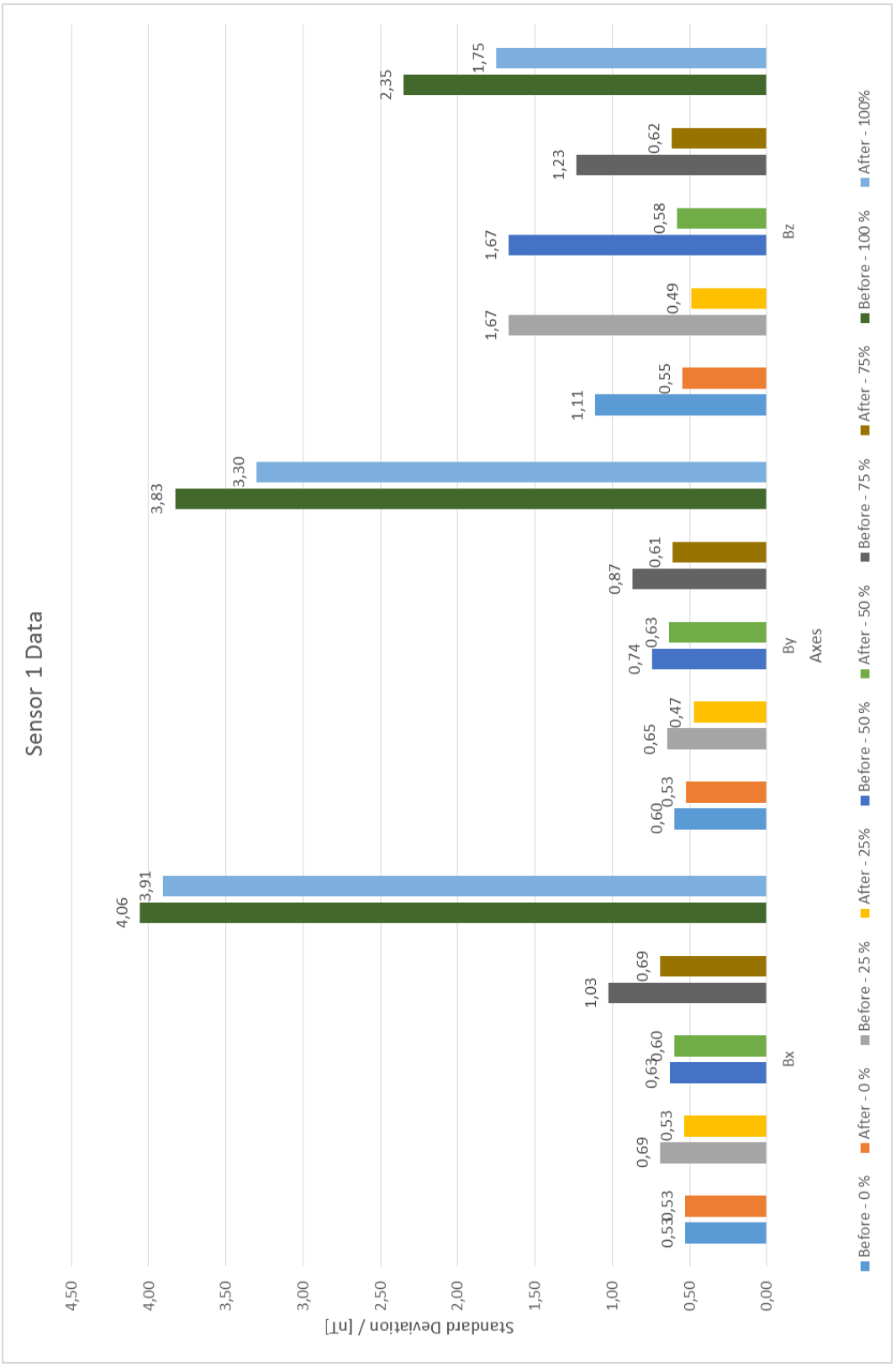


Figure 8-10: Summary of Sensor 1 data for throttle openings from 0 % to 100 %. Standard deviation was used as a measure of improvement of the pre-processed ('before') and post-processed ('after') data.

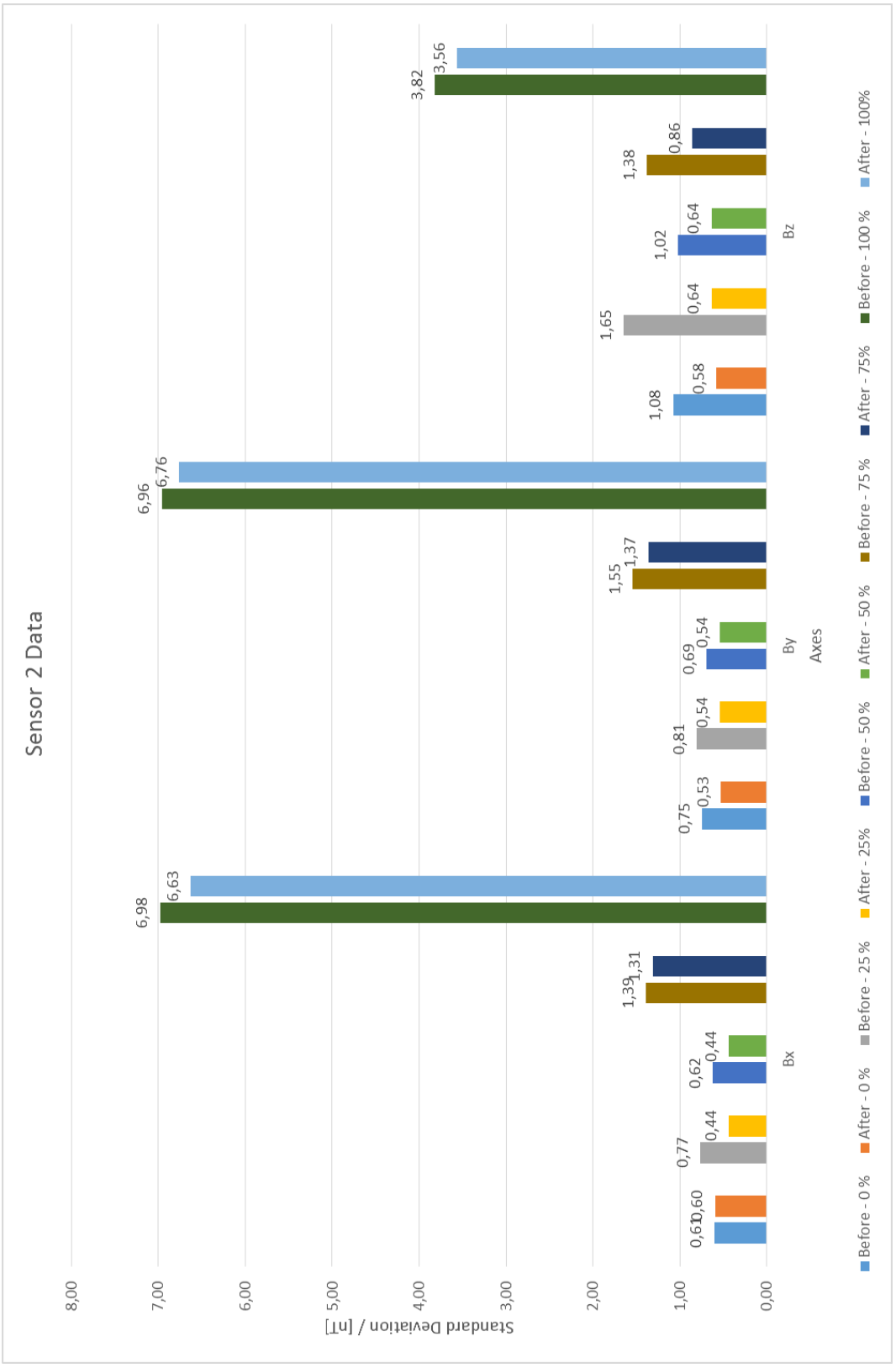


Figure 8-11: Summary of Sensor 2 data for throttle openings from 0 % to 100 %. Standard deviation was used as a measure of improvement of the pre-processed ('before') and post-processed ('after') data.

8.7 Conclusion

The high speed digitisation of magnetometer data with the ADS1278 ADC proved to be successful. The high sample rates allowed for accurate detection of AC and DC noise sources from the UAV propulsion system.

Using digital notch filters to remove the various AC components allowed a reduction in the basic noise levels caused by the propulsion system at various throttle openings.

The serendipitous wing tip movements of the UAV due to turbulent airflow during the large (75% to 100%) throttle opening tests showed the effects of the movements on the individual orthogonal axes of the magnetometers. It was interesting to note that the TMI values remained unaffected. It would also be interesting to make use of these individual orthogonal axes and perform compensation to these during dynamic flight trials. Perhaps the use of an inertial measurement unit (IMU) mounted on each magnetometer could be used to as an input to perform such compensation.

The concept of using two magnetometers symmetrically situated about a noise source for noise cancellation proved to be unfeasible with the current configuration. The main reason for this is that the fluxgate magnetometers are not 'synchronised'. The reason for the magnetometers not being 'synchronised' is due to the fact that the excitation circuitry of the two magnetometers are independent of one another, so the noise cannot easily be correlated between the two sensors. Some modifications to the sensors to allowing for a central excitation circuit would be required to achieve this. Perhaps one could look into a centralised excitation circuit for the two magnetometers which could be used to improve the noise correlation and perhaps achieve noise cancellation in this manner.

Chapter 9 – Magnetic Dipole Detection

9.1 Introduction

Various tests were set up to assess the performance of the magnetometer based payload in the detection of a magnetic dipole target. These experiments were conducted at SANSA Space Science on the 17th of May 2013. The aim of these experiments was to verify the effect of the increasing distance on the magnetic dipole's field strength. The relationship is expected to be proportional to $1/R^3$ where distance is represented as R.

9.2 Test Setup

The test setup entailed the marking out a 120 point 1 m x 1 m grid. The 15 m (wide) x 8 m (deep) grid was marked out using white sports field powder (see Figure 9-2 below).

The 8 m deep grid markings were used to simulate a height above the target for the UAV and magnetometers. The 15 m wide points would provide for a simulated distance port (left) and starboard (right) of the UAV. The 0 m line (see Figure 9-1 below for the marking out of the 0 m line) would represent the middle of the UAV, while the -7 m parallel line would represent the extreme port (left) side of the grid. The +7 m parallel line would represent the extreme starboard (right) side of the grid.

Figure 9-1: Setting up the experiment. Marking out the simulated height markings in 1 m intervals from 1 m to 8 m 'below' the UAV. This is the 0 m or 'centre line' being marked out.

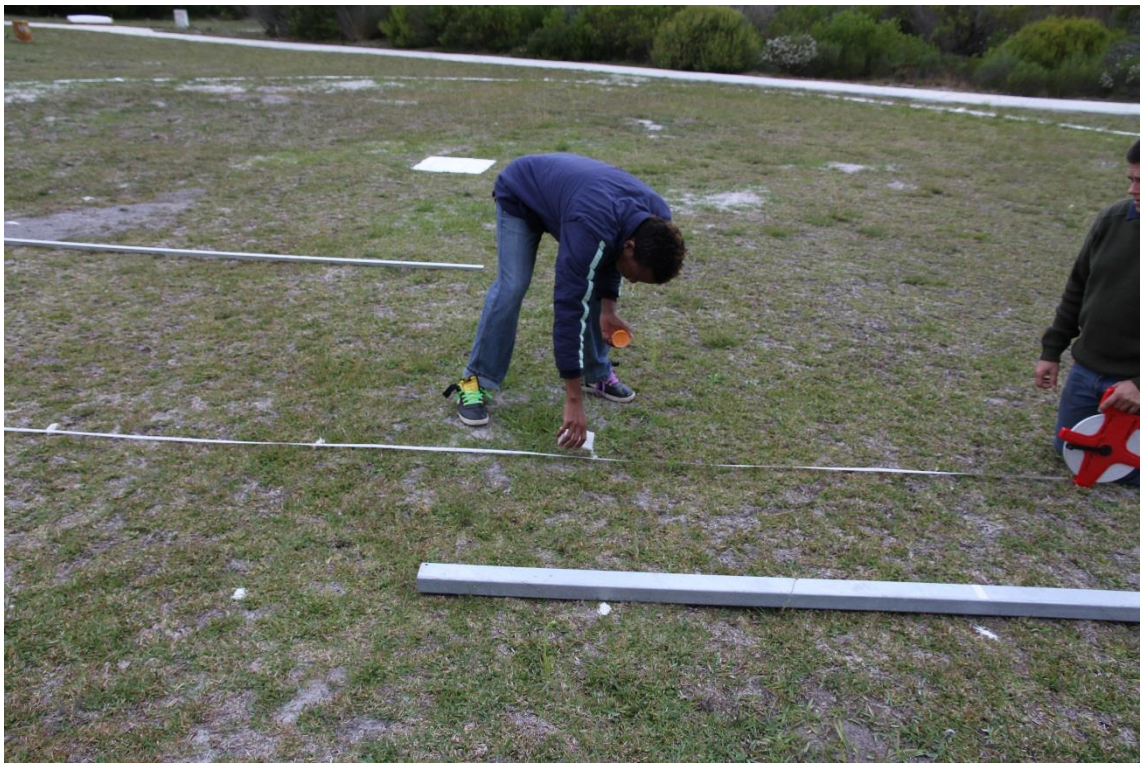


Figure 9-2: Marking out the 1 m grid points in the port to starboard orientation.

Once the grid was marked out, a rectangular soft metal pole 3 m in length was magnetised. The pole was magnetised by placing it inside a 61 turn cylindrical coil to which a 35 Ampere current was applied (see Figure 9-4 below). The pole was left to magnetise for approximately 20 minutes inside the coil. Figure 9-3 below shows the magnetisation process of the rectangular metal pole.

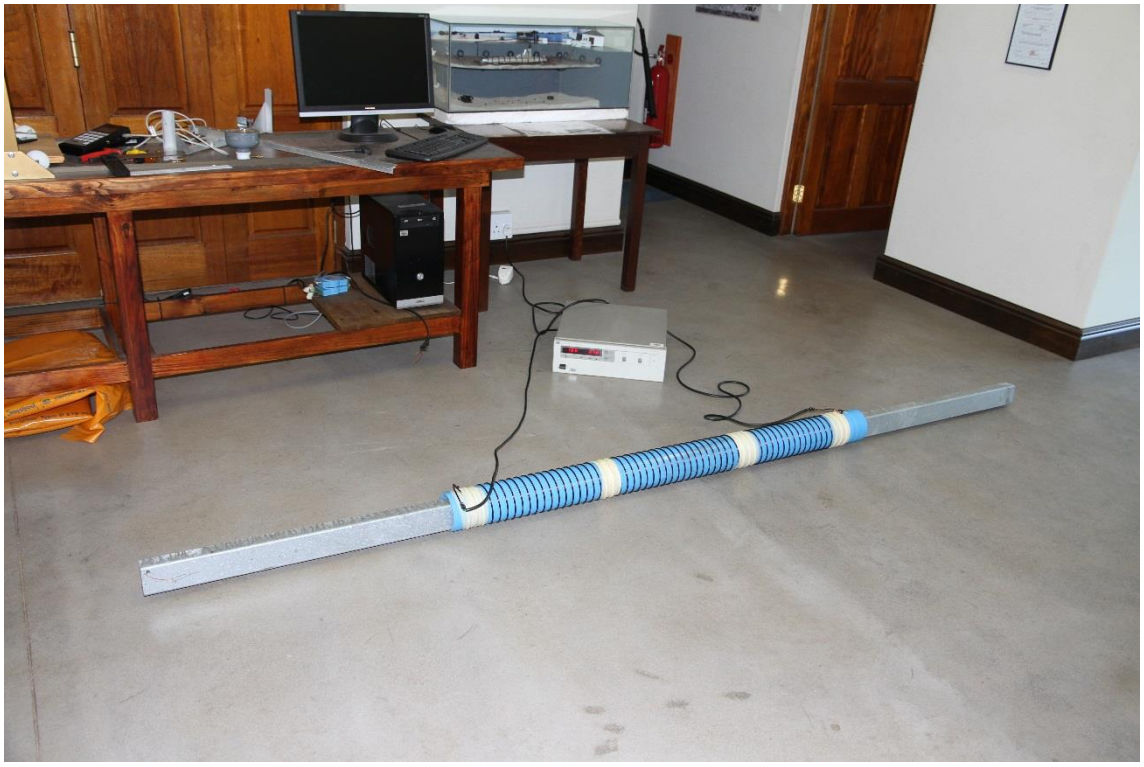


Figure 9-3: The magnetisation of the 3 m rectangular metal pole.

Figure 9-4: The HP 6021B DC power supply used to energise the 61 turn coil.

Figure 9-5: Characterisation of the 3 m magnetised rectangular metal pole. Note the position of the magnetometer 73 cm above the magnetised rectangular pole.

Once the 3 m rectangular metal pole was sufficiently magnetised, the pole was characterised for its magnetic dipole nature. Figure 9-5 above shows the process used to characterise the 3 m magnetised rectangular metal pole.

9.3 Test Results

The results for the characterisation of the magnetised rectangular pole and the magnetic dipole detection tests are presented below.

9.3.1 Characterisation Results of the Magnetised Metal Pole

The characterisation of the magnetic dipole due to the magnetisation process of the 3 m metal pole is shown below. Note the magnetometer placed 73 cm above the magnetised pole and digitisation equipment for the magnetometer output in Figure 9-5 above. The output was recorded for five points along the metal pole. The distances for which measurements were taken along the magnetised pole length are:

1. 0 mm
2. 760 mm
3. 1500 mm
4. 2260 mm
5. 3000 mm

Individual plots are shown in Appendix J. The plots show the normalised (geomagnetic field removed) data of the magnetised pole. Note the classic (asymmetrical) dipole shape along some of the magnetic axes as well as the TMI plot.

9.3.2 Magnetic Dipole Detection Results

Once the 3 m magnetised rectangular metal pole had been characterised, the magnetic dipole detection test, using the two LEMI-011b magnetometers and digitisation equipment outlined in Chapter 5, was carried out. A 1 m cubic non-magnetic structure was constructed to mount the UAV firmly in place while the tests were conducted.

It must be noted that the UAV platform was mounted in the orientation shown in Figure 9-6 and Figure 9-7 below to simulate the UAV 'flying' over the target i.e. the target was 'below' the UAV in these static tests. This was done to easily compare the static results to a dynamic[5] flight run (should one be done).

Figure 9-6: Measurement equipment setup showing test equipment and non-magnetic cubic support structure.

[5] It must be noted that due to time, weather and funding limitations, dynamic flight trials were sadly not conducted.

Figure 9-7: Measurement equipment setup showing test equipment and non-magnetic cubic support structure. Image taken from a distance with the recording laptop in the background.

The 3 m magnetised pole was then placed with its centre on each of the 120 grid points and a measurement taken. This was done for 2 heights. One set of measurements was done with the 3 m pole on the ground representing a 600 mm offset 'behind' the UAV while the second set of measurements were conducted with the 3 m pole held parallel to the ground at the same height as the 2 LEMI-011b magnetometers (see Figure 9-9 below).

Figure 9-8: The 3m pole being held at the same height as the LEMI-011b magnetometers with the centre of the pole over the extreme port (left) side of the grid. Example grid co-ordinates for the measurements shown in the figure are -7 m; +2 m, i.e. extreme left (-7 m) and 2 m 'away'.

Figure 9-9: Measuring the height of the port side LEMI-011b magnetometer. The figure shows the orientation of the axes for the port side magnetometer and the height above the ground.

The full set of results for the two sets of measurements are presented in Appendix K. The data plots show the data collected with the ambient geomagnetic field removed. The plots were achieved by manipulating the recorded data using the MATLAB code found in Appendix H and then plotted in Microsoft Excel 2013.

9.3.2.1 Test results for the 3 m pole 600 mm above the ground

Figure 9-10 and Figure 9-11 below show the results for the TMI plots for sensor 1 and sensor 2 for the experiment where the magnetised metal pole was held in the horizontal (600 mm above the ground) to the sensors. The full set of results is presented it Appendix K.

Note the curve fitting results in Figure 9-12 and Figure 9-13 below confirming the characteristic fall-off of the magnetic dipole strength following a $1/R^3$.

Figure 9-14 and Figure 9-15 below show the results for the TMI plots for sensor 1 and sensor 2 for the experiment where the magnetised metal pole was placed on the ground (600 mm below the horizontal to the sensors). The full set of results is presented it Appendix K.

Note the curve fitting results in Figure 9-12 and Figure 9-13 (first experiment) and Figure 9-16 and Figure 9-17 below (second experiment) confirming the characteristic fall-off of the magnetic dipole strength following a $1/R^3$.

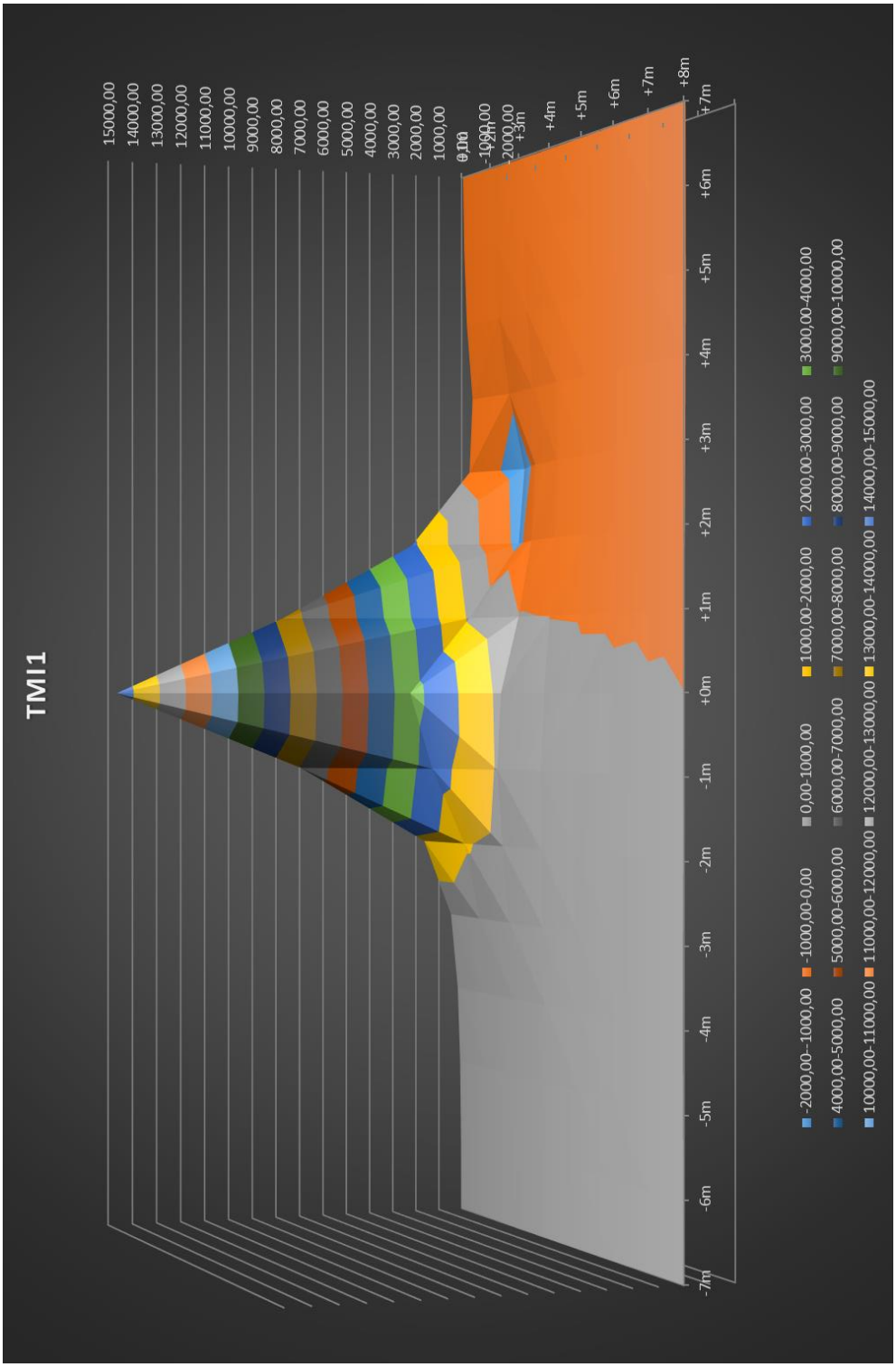


Figure 9-10: TMI data for sensor 1 with the 3 m magnetised pole held at the same height (600 mm above the ground) as the LEMI-011b magnetometers. Please note the vertical axis displays the values in nT.

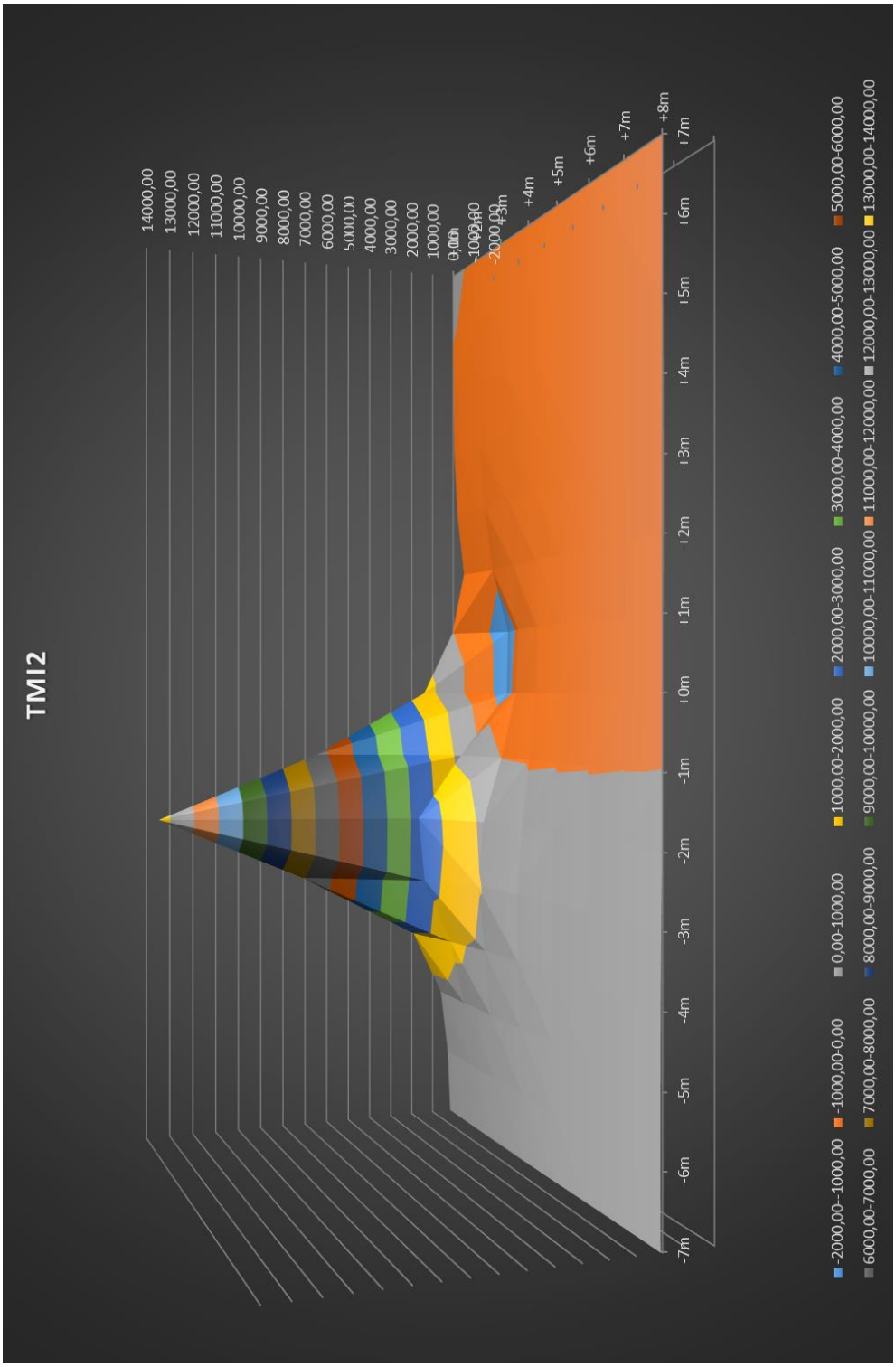


Figure 9-11: TMI data for sensor 2 with the 3 m magnetised pole held at the same height (600 mm above the ground) as the LEMI-011b magnetometers. Please note the vertical axis displays the values in nT.

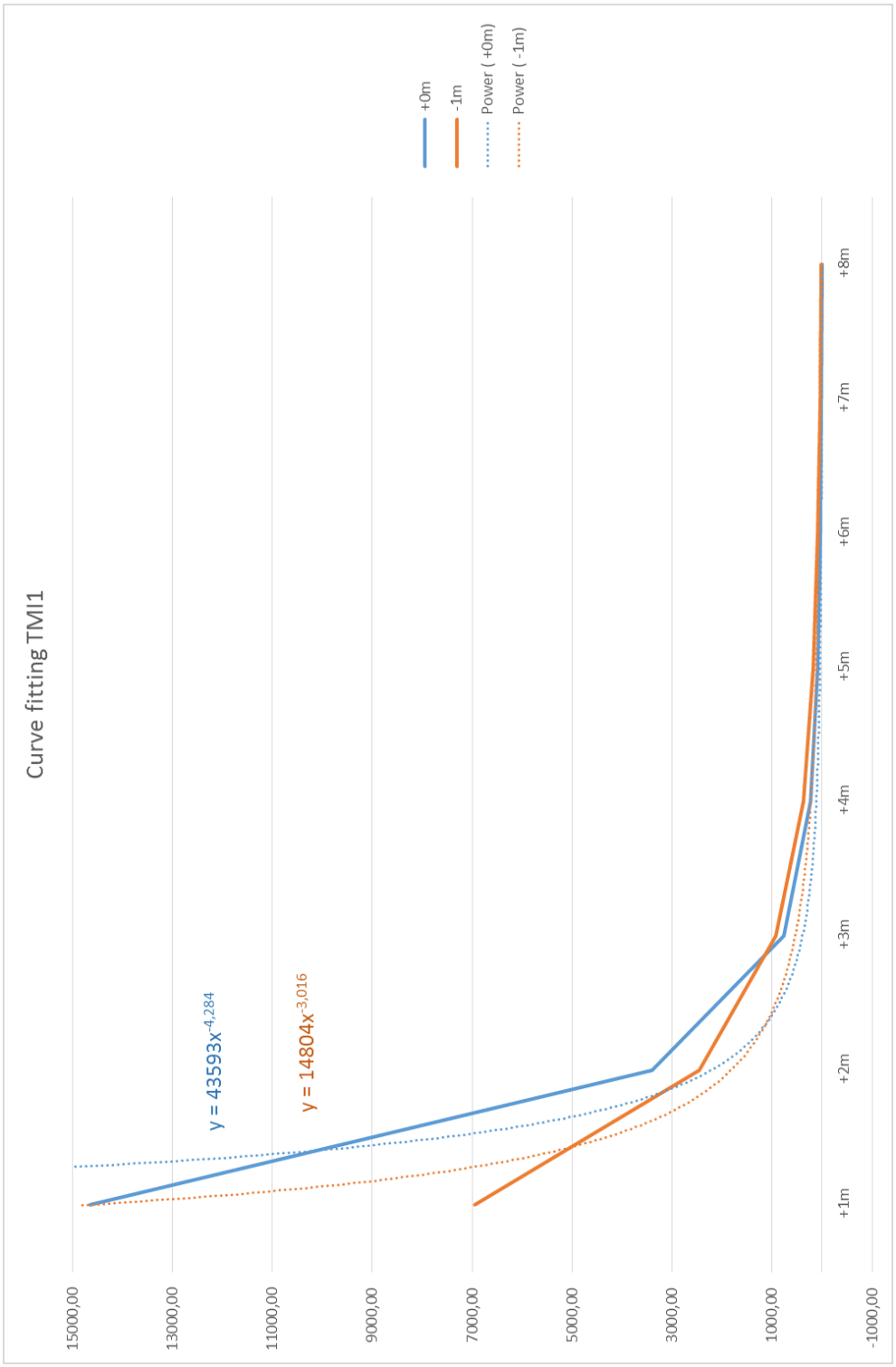


Figure 9-12: Curve fitting of the TMI data for the magnetised pole held 600 mm above the ground for sensor 1. The fitted curves were used to approximate the characteristic fall off of $1/R^3$ associated with magnetic dipoles. Please note the vertical axis displays the values in nT.

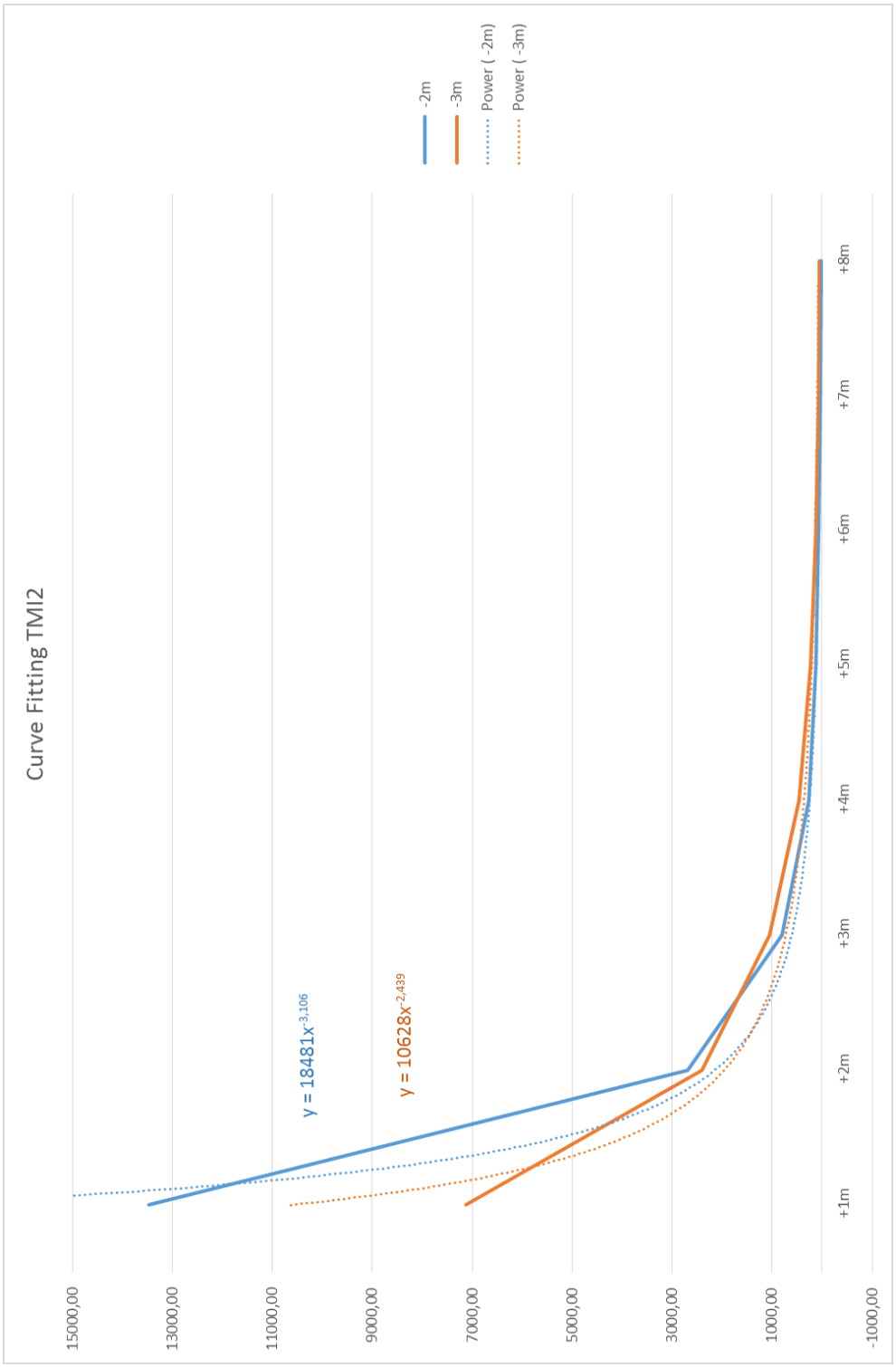


Figure 9-13: Curve fitting of the TMI data for the magnetised pole held 600 mm above the ground for sensor 2. The fitted curves were used to approximate the characteristic fall off of $1/R^3$ associated with magnetic dipoles. Please note the vertical axis displays the values in nT.

9.3.2.2 Test results for the 3 m pole on the ground

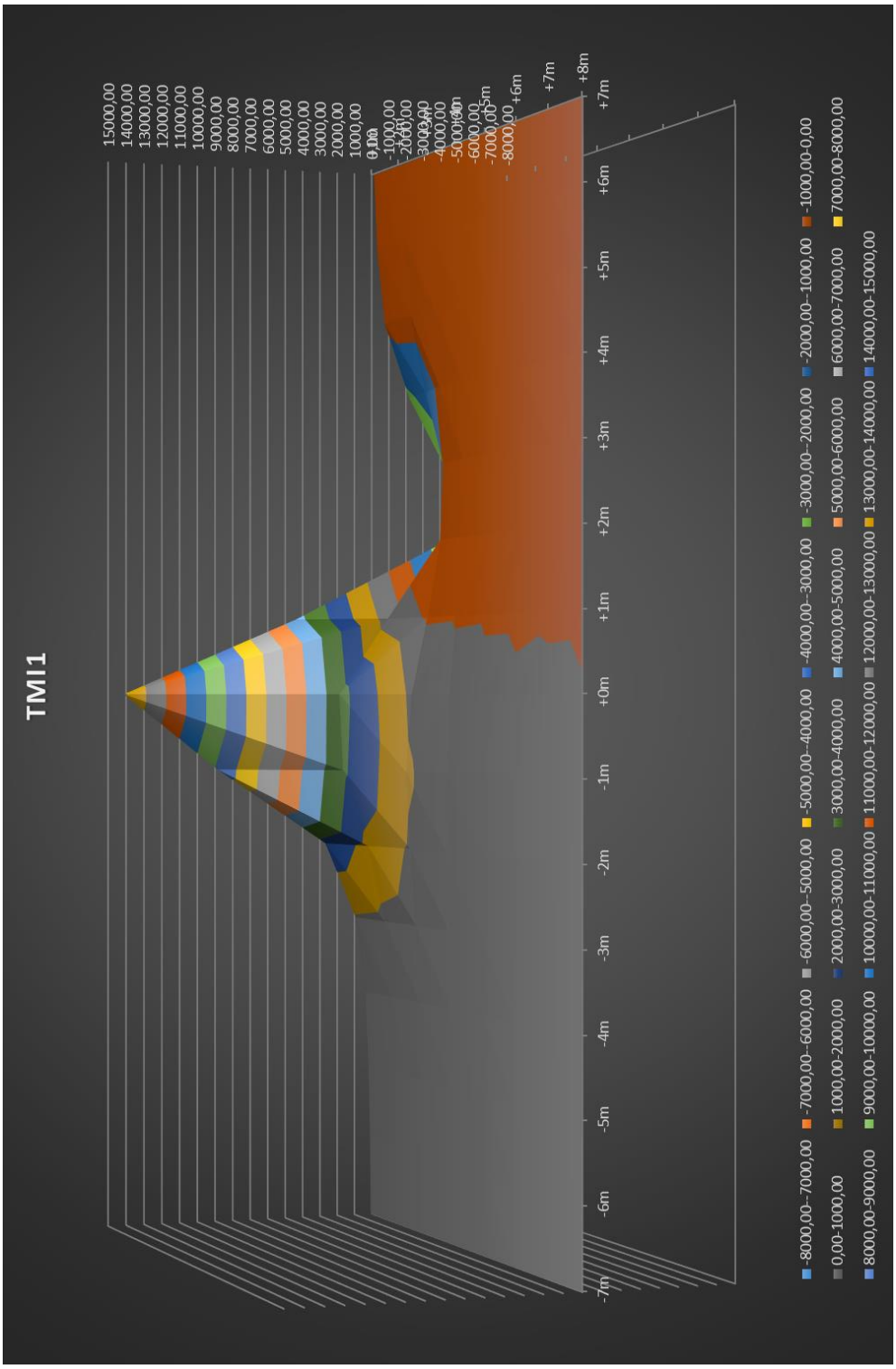


Figure 9-14: TMI data for sensor 1 with the 3 m magnetised pole 600 mm below (on the ground) the LEMI-011b magnetometers. Please note the orientation of the TMI data presented is different to that of the individual axial components. Please note the vertical axis displays the values in nT.

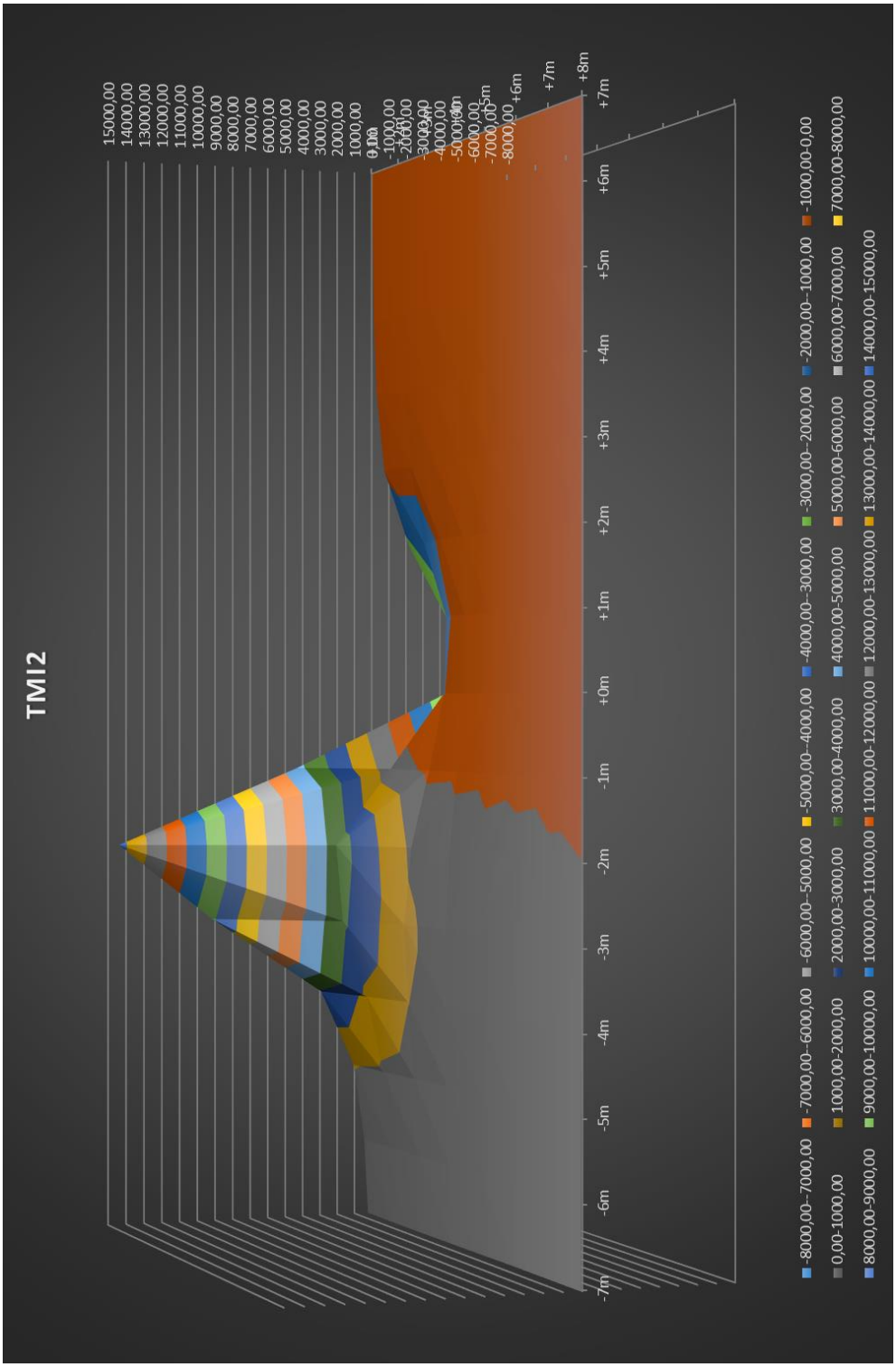


Figure 9-15: TMI data for sensor 2 with the 3 m magnetised pole 600 mm below (on the ground) the LEMI-011b magnetometers. Please note the orientation of the TMI data presented is different to that of the individual axial components. Please note the vertical axis displays the values in nT.

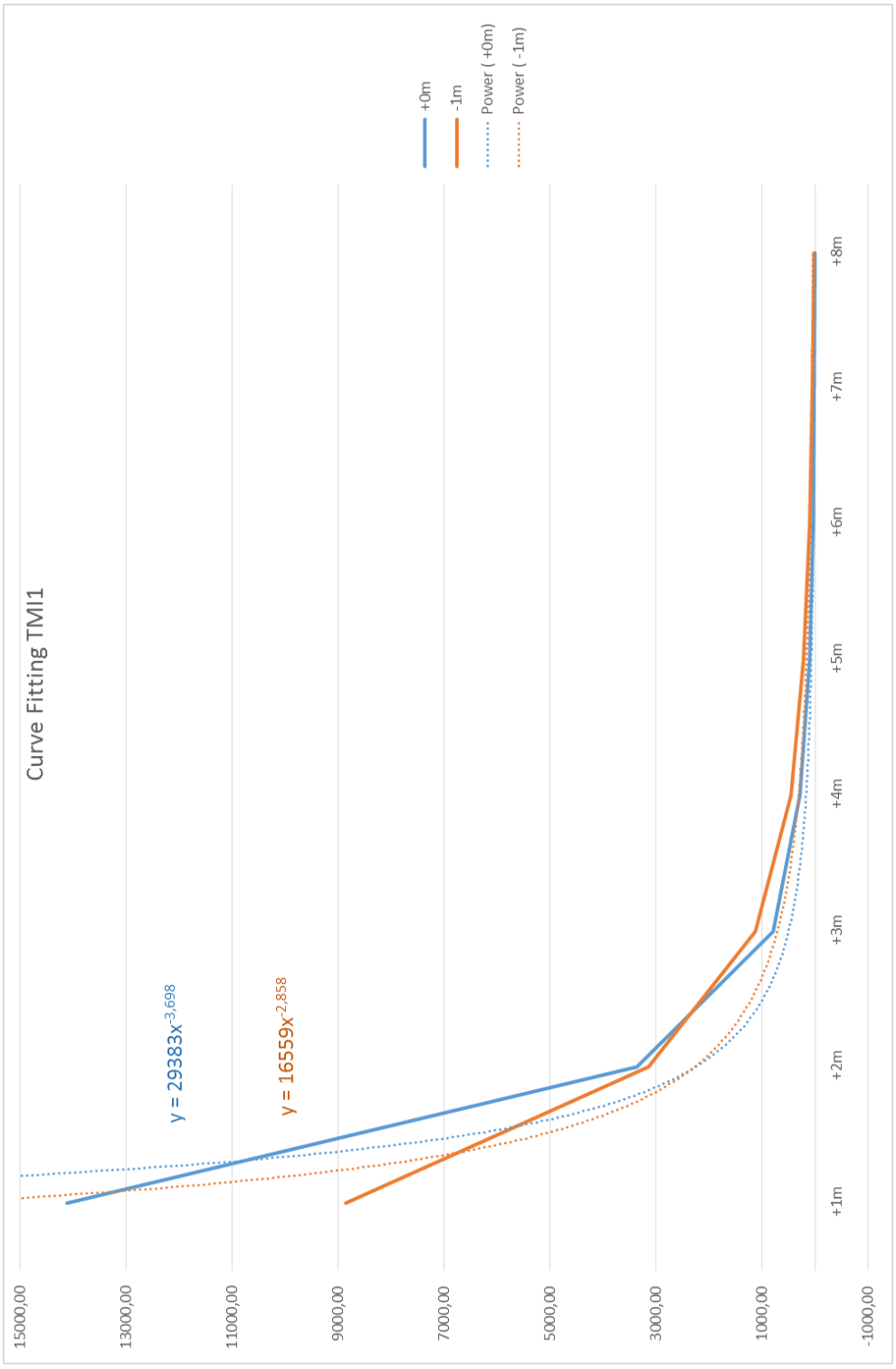


Figure 9-16: Curve fitting of the TMI data for the magnetised pole placed on the ground for sensor 1. The fitted curves were used to approximate the characteristic fall off of $1/R^3$ associated with magnetic dipoles. Please note the vertical axis displays the values in nT.

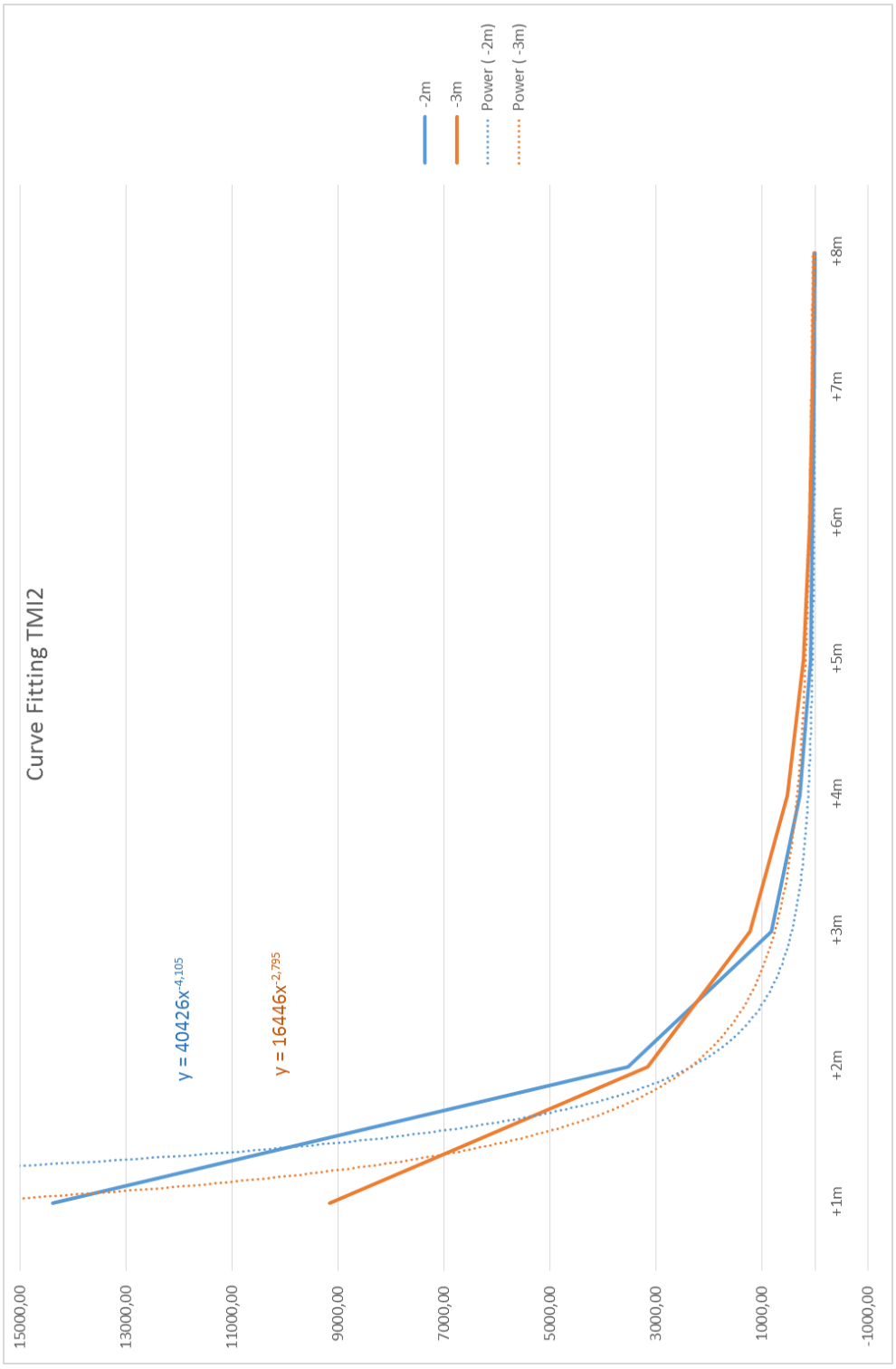


Figure 9-17: Curve fitting of the TMI data for the magnetised pole placed on the ground for sensor 2. The fitted curves were used to approximate the characteristic fall off of $1/R^3$ associated with magnetic dipoles. Please note the vertical axis displays the values in nT.

9.4 Conclusion

The results show that a magnetic dipole with the characteristics shown in 9.3.1 could be detected at various distances up to 8 m away from the magnetometers.

Note that the asymmetric nature of the 3 m rectangular magnetised pole was visible in the two sets of data. By offsetting the 3 m magnetised rectangular pole by 600 mm, by placing it on the ground, effectively provided a more 'symmetric' view of the dipole target.

The prominent distances of the TMI data were plotted for both sets of data collected and a curve fitted using Microsoft Excel 2013. The curve fittings illustrates the characteristic fall-off of the magnetic dipole strength following a $1/R^3$ trend associated with magnetic dipoles. These results were as expected and yet interesting, especially noting the asymmetry of the dipole target.

Chapter 10– Conclusion and Recommendations for Future Work

10.1 Conclusion

This dissertation presents the analysis, design and practical implementation of a magnetometer based payload for a point-take-off-and-land (PTOL) UAV. This includes:

1. The development of a dual tri-axial fluxgate magnetometer datalogger payload and the evaluation thereof.

2. The construction and evaluation of a PTOL UAV.

3. The characterisation of the magnetic signature for the PTOL UAV's propulsion system along with subsequent magnetic noise mitigation techniques.

4. Finally, the payload is used to detect a characterised magnetic dipole target to ascertain the performance of the sensors and associated systems.

The conclusion of these points are presented below.

10.1.1 Prototype Datalogger

The prototype datalogger proved to be adequate for collection of basic data but revealed some limitations in the chosen hardware.

While the microcontroller and the ADC could successfully be interfaced via the SCC interface, the data rates proved to be quite high. The high data rates coupled with the 1 Hz GPS fix rate cycle meant that only a limited amount of data (512 samples) could be read, averaged and written to the SD card in a cycle. This meant that not enough data could be collected to perform the post processing required to remove spectral artefacts associated with the environment and propulsion system.

10.1.2 PTOL UAV

Despite some setbacks the UAV design proved robust enough to undertake multiple launches and landings successfully. The decision to use COTS components and systems was justified as it aided in the developments of the UAV platform and offered reasonable performance for the costs incurred.

In conclusion the UAV platform performed very well and proved to be an adequate platform to perform further testing with the magnetometers onboard.

10.1.3 Propulsion System Characterisation

The high speed digitisation of magnetometer data with the ADS1278 ADC proved to be successful. The high sample rates allowed for accurate detection of AC and DC noise sources from the UAV propulsion system.

Using digital notch filters to remove the various AC components allowed a reduction in the basic noise levels caused by the propulsion system at various throttle openings.

The serendipitous wing tip movements of the UAV due to turbulent airflow during the large (75% to 100%) throttle opening tests showed the effects of the movements on the individual orthogonal axes of the magnetometers. It was interesting to note that the TMI values remained unaffected. It would also be interesting to make use of these individual orthogonal axes and perform compensation to these during dynamic flight trials. Perhaps the use of an inertial measurement unit (IMU) mounted on each magnetometer could be used to as an input to perform such compensation.

The concept of using two magnetometers symmetrically situated about a noise source for noise cancellation proved to be unfeasible with the current configuration. The main reason for this is that the fluxgate magnetometers are not 'synchronised'. The reason for the magnetometers not being 'synchronised' is due to the fact that the excitation circuitry of the two magnetometers are independent of one another, so the noise cannot easily be correlated between the two sensors. Some modifications to the sensors allowing for a central excitation circuit would be required to achieve this. Perhaps one could look into a centralised excitation circuit for the two magnetometers which could be used to improve the noise correlation and perhaps achieve noise cancellation in this manner.

10.1.4 Magnetic Dipole Detection

The results show that a magnetic dipole with the characteristics shown in 9.3.1 could be detected at various distances up to 8 m away from the magnetometers.

Note that the asymmetric nature of the 3 m rectangular magnetised pole was visible in the two sets of data. By offsetting the 3 m magnetised rectangular pole by 600 mm, by placing it on the ground, effectively provided a more 'symmetric' view of the dipole target.

The prominent distances of the TMI data were plotted for both sets of data collected and a curve fitted using Microsoft Excel 2013. The curve fittings show the characteristic fall-off of the magnetic dipole strength following a $1/R^3$ trend associated with magnetic dipoles. These results were as expected and yet interesting, especially noting the asymmetry of the dipole target.

10.2 Recommendations for Future Work

Based on the work carried out in this dissertation various conclusions were reached as well as a number of limitations found. The following recommendations are based on the experience gained during this process and are in no way complete but it is felt that these recommendations would greatly benefit this field of work in the future:

1. Synchronisation of the excitation circuitry of the fluxgate magnetometers to investigate the use of two magnetometers as a noise cancellation mechanism.

2. Improve the interface to the ADS1278 ADC to exploit the full potential of the digitisation of the fluxgate magnetometer data. Perhaps use the continuous data mode via the SPI interface to perform this.

3. Add roll, pitch and yaw sensors to the magnetometers to allow for single axis data use and correction. Perhaps the use of an IMU mounted directly on the magnetometers could go a long way in achieving an input for this.

4. Addition of the ground station with magnetometer and differential GPS to improve geospatial accuracy of the recorded data and an ambient geomagnetic reference for the duration of the trials/data gathering.

5. Perform real time signal processing to remove noise and send data via downlink for real time flight testing.

6. Improve the data acquisition rate of the datalogger to 100 Hz or more. This will allow closer data samples during flight trials.

7. Perform dynamic tests of the datalogger in a flight trial application.

Bibliography

[1] Wikipedia, "Geophysics," Wikipedia, 26 July 2013. [Online]. Available: http://en.wikipedia.org/wiki/Geophysics. [Accessed 18 August 2013].

[2] Wikipedia, "Unmanned aerial vehicle," Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/Unmanned_aerial_vehicle. [Accessed 30 July 2013].

[3] S. G. Gupta, M. M. Ghonge, P. M. Jawandhiya, "Review of Unmanned Aircraft System (UAS)," *International Journal of Advanced Research in Computer Engineering & Technology,* vol. 2, no. 4, p. 1646, April 2013.

[4] H. Chao, Y. Cao, Y. Chen, "Autopilots for Small Unmanned Aerial Vehicles: A Survey," *International Journal of Control, Automation, and Systems,* vol. 8, no. 1, pp. 36 - 44, 2010.

[5] N. Wagner, S. Boland, B. Taylor, et al., "Powertrain Design for Hand-Launchable Long Endurance Unmanned Aerial Vehicles," Colorado State University, Fort Collins, 2011.

[6] R. Beard, D. Kingston, M. Quigley, et al., "Autonomous Vehicle Technologies For Small Fixed-Wing UAVs," *Journal of Aerospace Computing, Information and Communication,* vol. 2, no. 1, pp. 92 - 108, January 2005.

[7] P. Kearey, M. Brooks, I. Hill, An Introduction to Geophysical Exploration. 3rd Edition, Oxford: Blackwell Science, 2002.

[8] Wikipedia, "Exploration geophysics," Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/Exploration_geophysics. [Accessed 20 June 2013].

[9] The University of California - Berkeley, "THE BERKELEY COURSE IN APPLIED GEOPHYSICS," The University of California - Berkley, [Online]. Available: http://appliedgeophysics.lbl.gov/intro/index.html. [Accessed 20 June 2013].

[10] W.E.S. Urquhart, “General Introduction to Airborne Magnetic Surveys,” GeoExplo Ltda., March 2013. [Online]. Available: http://www.geoexplo.com/airborne_survey_workshop.html. [Accessed 20 June 2013].

[11] National Geophysical Data Center, “National Geophysical Data Center,” [Online]. Available: http://www.ngdc.noaa.gov/geomag/faqgeom.shtml. [Accessed 08 July 2013].

[12] Wikipedia, “Earth's magnetic field,” Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/Earth%27s_magnetic_field. [Accessed 07 July 2013].

[13] Wikipedia, “Permeability (electromagnetism),” Wikipedia, 20 January 2014. [Online]. Available: http://en.wikipedia.org/wiki/Permeability_%28electromagnetism%29. [Accessed 20 January 2014].

[14] R. Hansen, “Magnetic Survey Design,” EDCON-PRJ, 22 January 2007. [Online]. Available: http://www.prj.com/SurveyDesign-final.htm. [Accessed 20 June 2013].

[15] GEM System Advanced Magnetometers, “GEM System Advanced Magnetometers,” GEM System Advanced Magnetometers, 2013. [Online]. Available: http://www.gemsys.ca/. [Accessed 21 July 2013].

[16] Fugro, “Fugro Airborne Surveys,” Fugro, [Online]. Available: http://www.fugroairborne.com/services/geophysicalservices/bysurvey/aeromagnetic/. [Accessed 21 July 2013].

[17] J. E. Lenz, “A Review of Magnetic Sensors,” *Proceedings of the IEEE,* vol. 78, no. 6, pp. 973 - 989, June 1990.

[18] Wikipedia, “Maxwell's equations,” Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Maxwell%27s_equations. [Accessed 11 July 2013].

[19] I. Hrvoic, G. Hollyer, “Brief Review of Quantum Magnetometers,” GEM Systems, Richmond Hill, Ontario, Canada.

[20] Wikipedia, “Magnetometer,” Wikipedia, 17 December 2013. [Online]. Available: http://en.wikipedia.org/wiki/Magnetometer. [Accessed 20 December 2013].

[21] Wikipedia, “Larmor precession,” Wikipedia, 16 December 2013. [Online]. Available: http://en.wikipedia.org/wiki/Larmor_precession. [Accessed 2 January 2014].

[22] Wikipedia, “Optical pumping,” Wikipedia, 26 February 2013. [Online]. Available: http://en.wikipedia.org/wiki/Optical_pumping. [Accessed 2 January 2014].

[23] Wikipedia, “Proton magnetometer,” Wikipedia, 27 December 2013. [Online]. Available: http://en.wikipedia.org/wiki/Proton_magnetometer. [Accessed 2 January 2014].

[24] Wikipedia, “Gyromagnetic ratio,” Wikipedia, 4 September 2013. [Online]. Available: http://en.wikipedia.org/wiki/Gyromagnetic_ratio. [Accessed 3 January 2014].

[25] Wikipedia, “Nuclear Overhauser effect,” Wikipedia, 12 December 2013. [Online]. Available: http://en.wikipedia.org/wiki/Nuclear_Overhauser_effect. [Accessed 2 January 2014].

[26] Wikipedia, “Magnetometer - Overhauser effect magnetometer,” Wikipedia, 30 December 2013. [Online]. Available: http://en.wikipedia.org/wiki/Magnetometer#Overhauser_effect_magnetometer. [Accessed 2 January 2014].

[27] V. Korepanov, R. Berkman, L. Rakhlin, et al., “Advanced field magnetometers comparative study,” *Measurement,* vol. 29, no. 2, pp. 137 - 146, 2001.

[28] P. Ripka, “Advances in fluxgate sensors,” *Sensors and Actuators A: Physical,* vol. 106, no. 1, pp. 8 - 14, 2003.

[29] F. Primdahl, “The fluxgate magnetometer,” *Journal of Physics,* vol. 12, no. 4, p. 241, 1979.

[30] D. I. Gordon, R. E. Brown, “Recent Advances in Fluxgate Magnetometry,” *IEEE Transactions on Magnetics,* vol. 8, no. 1, pp. 76 - 82, March 1972.

[31] S. Foner, “Review of Magnetometry,” *IEEE Transactions on Magnetics,* vol. 17, no. 6, pp. 3358 - 3363, 1981.

[32] Wikipedia, "PTOL," Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/PTOL. [Accessed 21 July 2013].

[33] C. Eck, B. Imbach, "Aerial Magnetic Sensing With An UAV Helicopter," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* vol. 38, no. 1, 2011.

[34] K. Muraoka, N. Okada, D. Kubo, "Quad Tilt Wing VTOL UAV: Aerodynamic Characteristics and Prototype Flight Test," in *AIAA Unmanned… Unlimited Conf., American Institute of Aeronautics and Astronautics*, Seattle, Washington, 6-9 April 2009.

[35] APM Multiplatform Autopilot, "APM Autopilot Suite," 2013. [Online]. Available: http://ardupilot.com. [Accessed 20 June 2013].

[36] P. Yedamale, "Brushless DC (BLDC) Motor Fundamentals - AN885," MicroChip Technology Inc., 2003.

[37] AdamOne, "Welcome to Model Aircraft," February 2003. [Online]. Available: http://adamone.rchomepage.com/index.html. [Accessed 14 July 2013].

[38] Battery University, "Battery University: What's the Best Battery?," Cadex Electronics Inc., [Online]. Available: http://batteryuniversity.com/learn/article/whats_the_best_battery. [Accessed 14 July 2013].

[39] Hyperion, "Hyperion G3 LiPo - Specifications," 20 April 2010. [Online]. Available: http://media.hyperion.hk/dn/g3lipo/G3-Specs-EN.pdf. [Accessed 14 July 2013].

[40] R. Versteeg, M. McKay, M. Anderson et al., "Feasibility Study for an Autonomous UAV - Magnetometer System," Idaho National Laboratory, Idaho, 2007.

[41] M. D. McKay, M. O. Anderson, R. Versteeg, "Development of Autonomous Magnetometer Rotorcraft for Wide Area Assessment," Idaho National Laboratory, Idaho, August 2011.

[42] C. Eck, “Aerial Magnetic Scanning with an Unmanned Helicopter,” *UAS: The Global Perspective,* 9th Edition June 2011.

[43] Hochschule Luzern: Felix Wuersten, “Troika der Lüfte,” January 2012. [Online]. Available: http://www.aeroscout.ch/downloads/magazin_troika_der_luefte.pdf. [Accessed 14 July 2013].

[44] B. Zhang, Z. Guo, Y Qiao, “A Simplified Aeromagnetic Compensation Model For Low Magnetism UAV Platform,” in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Vancouver, Canada, 2011.

[45] D. Clarke, “Magnetic Signature of Brushless Electric Motors,” Australian DoD Defence Science and Technology Organisation, Fishermans Bend, Victoria, Australia, April 2006.

[46] National Space Agency of Ukraine, “Flux-gate miniature magnetometer LEMI-011B Nr.: B099 - Technical Description and User Manual,” Lviv Centre of Institute of Space Research, Lviv, Ukraine, 2009.

[47] D. Gouws - HMO, “LEMI-011 Low Cost Fluxgate Magnetometer Specifications,” Hemanus Magnetic Observatory, Hermanus.

[48] Geometrics, “G-823A CESIUM MAGNETOMETER Specifications,” Geometrics, San Jose, California, 2005.

[49] GEM System Advanced Magnetometers, “Ultra Light Magnetometer for UAV Specifications,” GEM System Advanced Magnetometers, Markham, Ontario, 2012.

[50] G. Bartington, C. E. Chapman, “A High-stability Fluxgate Magnetic Gradiometer for Shallow Geophysical Survey Applications,” *Archaeological Prospection,* vol. 11, p. 19–34, 2004.

[51] Texas Instruments, “ADS1278 - Octal Simultaneous Sampling, 24-Bit Analog-to-Digital Converters,” July 2009. [Online]. Available: http://www.ti.com. [Accessed 20 June 2013].

[52] Texas Instruments, “ADS1278EVM-PDK - User’s Guide,” April 2010. [Online]. Available: http://www.ti.com. [Accessed 20 June 2013].

[53] Texas Instruments, “ADS1x7xEVM-PDK - User’s Guide,” February 2012. [Online]. Available: http://www.ti.com. [Accessed 20 June 2013].

[54] Atmel Corporation, “AT91SAM7X-EK Evaluation Board for AT91SAM7X and AT91SAM7XC - User Guide,” 22 March 2007. [Online]. Available: http://www.atmel.com. [Accessed 20 June 2013].

[55] Atmel Corporation, “AT91 ARM® Thumb®-based Microcontrollers - AT91SAM7X256/AT91SAM7X128 - Preliminary,” 2 February 2006. [Online]. Available: http://www.atmel.com. [Accessed 20 June 2013].

[56] M. J. Butcher Consulting, “µTasker User Guide – First Steps for New Users,” 2009. [Online]. Available: http://www.utasker.com/docs/uTasker/uTaskerV1.4_user_guide.PDF. [Accessed 20 June 2013].

[57] M. J. Butcher Consulting, “AT91SAM7X Tutorial – Ethernet and the Simulator,” 2009. [Online]. Available: http://www.utasker.com/docs/SAM7X/uTaskerV1-4_SAM7X.PDF. [Accessed 20 June 2013].

[58] Fastrax Ltd., “TECHNICAL DESCRIPTION Fastrax UP501 GPS Receiver,” 29 October 2010. [Online]. Available: http://www.fastraxgps.com. [Accessed 2012].

[59] Fastrax Ltd., “NMEA Manual for Fastrax IT500 Series GPS receivers,” 2 June 2009. [Online]. Available: http://www.fastraxgps.com. [Accessed 2012].

[60] P. Jawayon, J. Srisertpol, “Speed Estimation Of 3-Phase BLDC Motor Using Genetic Algorithm,” *International Journal of Engineering Science and Innovative Technology,* vol. 2, no. 1, pp. 254 - 263, January 2013.

[61] Windrider, “Windrider - Queen Bee,” [Online]. Available: http://www.windrider.com.hk/index.asp. [Accessed 03 August 2013].

[62] D. A. Caughey, Introduction to Aircraft Stability and Control Course Notes for M&AE 5070, Ithaca, New York: Sibley School of Mechanical & Aerospace Engineering Cornell University, 2011.

[63] stefanv.com, "An Electronic Speed Control Primer," stefanv.com, 30 September 1997. [Online]. Available: http://www.stefanv.com/electronics/escprimer.html. [Accessed 13 July 2013].

[64] Hyperion, "Hyperion ZS 3025 10-Turn 775KV," [Online]. Available: http://www.hyperion-world.com/products/product/HP-ZS3025-10. [Accessed 14 July 2013].

[65] Castle Creations , "Castle Creations Phoenix Ice Controllers," Castle Creations, [Online]. Available: http://www.castlecreations.com/products/phoenix_ice.html. [Accessed 03 August 2013].

[66] P. Pine, "Meaning of KV for electric motors," Electric Flight in Australia, [Online]. Available: http://www.flyelectric.com/ans.kv.html. [Accessed 17 August 2013].

[67] Japan Remote Control CO., Ltd, "Japan Remote Control CO., Ltd XG8 Radio Control," Japan Remote Control CO., Ltd, [Online]. Available: http://www.jrpropo.co.jp/english/propo/dmss/xg8.html. [Accessed 03 August 2013].

[68] The MathWorks, Inc., "DSP System Toolbox->Filter Design and Analysis->Filter Design->iirnotch," The MathWorks, Inc., 2013. [Online]. Available: http://www.mathworks.com/help/dsp/ref/iirnotch.html. [Accessed 23 December 2013].

[69] The MathWorks, Inc., "Signal Processing Toolbox->Spectral Analysis->Nonparametric Methods->Findpeaks," The MathWorks, Inc., 2013. [Online]. Available: http://www.mathworks.com/help/signal/ref/findpeaks.html. [Accessed 23 December 2013].

[70] The MathWorks, Inc., "MATLAB->Mathematics->Fourier Analysis and Filtering->fft," The MathWorks, Inc., 2013. [Online]. Available: http://www.mathworks.com/help/matlab/ref/fft.html. [Accessed 23 December 2013].

[71] The MathWorks, Inc., “Signal Processing Toolbox->Spectral Analysis->Nonparametric Methods->Spectral Analysis,” The MathWorks, Inc., 2013. [Online]. Available: http://www.mathworks.com/help/signal/ug/spectral-analysis.html#f12-20709. [Accessed 23 December 2013].

[72] The MathWorks, Inc, “MATLAB->GUI Building->Component Selection->Predefined Dialog Boxes->inputdlg,” The MathWorks, Inc, 2013. [Online]. Available: http://www.mathworks.com/help/matlab/ref/inputdlg.html. [Accessed 23 December 2013].

[73] M. J. Butcher Consulting, “µTasker – utFAT,” 2012. [Online]. Available: http://www.utasker.com/docs/uTasker/uTasker_utFAT.PDF. [Accessed 23 December 2013].

[74] M. J. Butcher Consulting, “µTasker - ATMEL SAM7X SSC Driver,” 2010. [Online]. Available: http://www.utasker.com/docs/uTasker/uTaskerSSC.PDF. [Accessed 23 December 2013].

[75] M. J. Butcher Consulting, “µTasker – UART User’s Guide,” 2013. [Online]. Available: http://www.utasker.com/docs/uTasker/uTaskerUART.PDF. [Accessed 23 December 2013].

[76] The MathWorks, Inc., “MATLAB->Graphics->Graphics Objects->Graphics Object Identification->gcf,” The MathWorks, Inc., 2013. [Online]. Available: http://www.mathworks.com/help/matlab/ref/gcf.html. [Accessed 23 December 2013].

[77] The MathWorks, Inc., “Documentation Center,” The MathWorks, Inc., 2013. [Online]. Available: http://www.mathworks.com/help/matlab/index.html. [Accessed 23 December 2013].

[78] N. Morrison, Introduction to Fourier Analysis, New York: John Wiley & Sons, Inc., 1994.

Appendices

Appendix A – Glossary of Abbreviations

A table of abbreviations applicable to this dissertation are given in the table below:

Table A - 1: Abbreviations applicable to this dissertation.

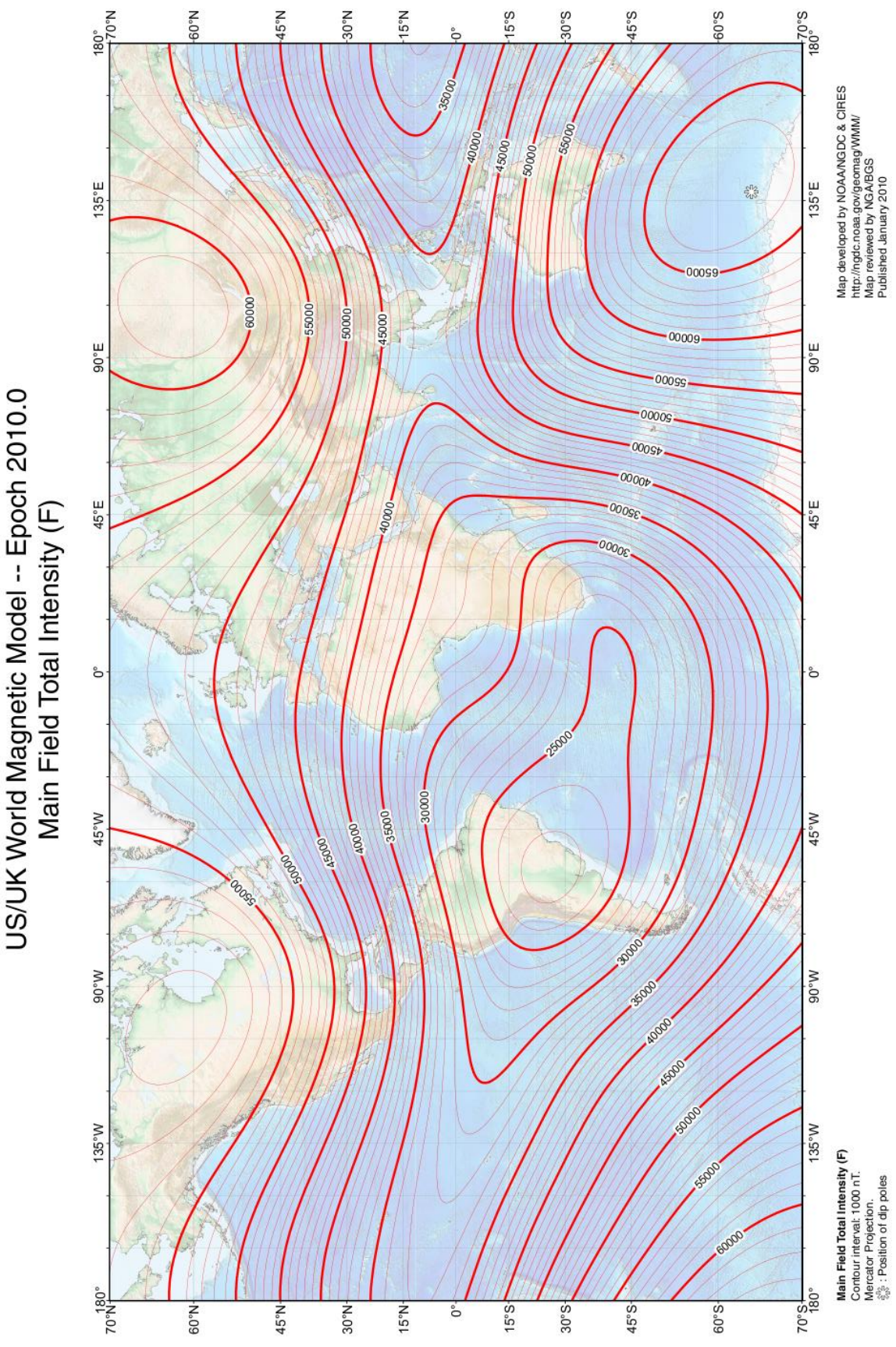
Abbreviation	Meaning/Definition
A	Ampere
AC	Alternating Current
Ah	Ampere-hour
AUV	Autonomous underwater vehicle
CAD	Canadian Dollar(s)
COTS	Commercial off-the-shelf
dB	Decibel
DC	Direct Current
DCBL	DC Brushless
EM	Electromagnetic
EMI	EM Interference
ESC	Electronic speed controller
FFT	Fast Fourier Transform
g	Gram (1E-03 kilogram)
G/Gs	Gauss (1E+05 nT)
GPS	Global Positioning System
Hz	Hertz
kg	kilogram
kHz	kilohertz (1E+03 Hertz)
km	kilometre (1E+03 Metre)
kSPS	kilo samples per second
LCISR	Lviv Centre of Institute of Space Research

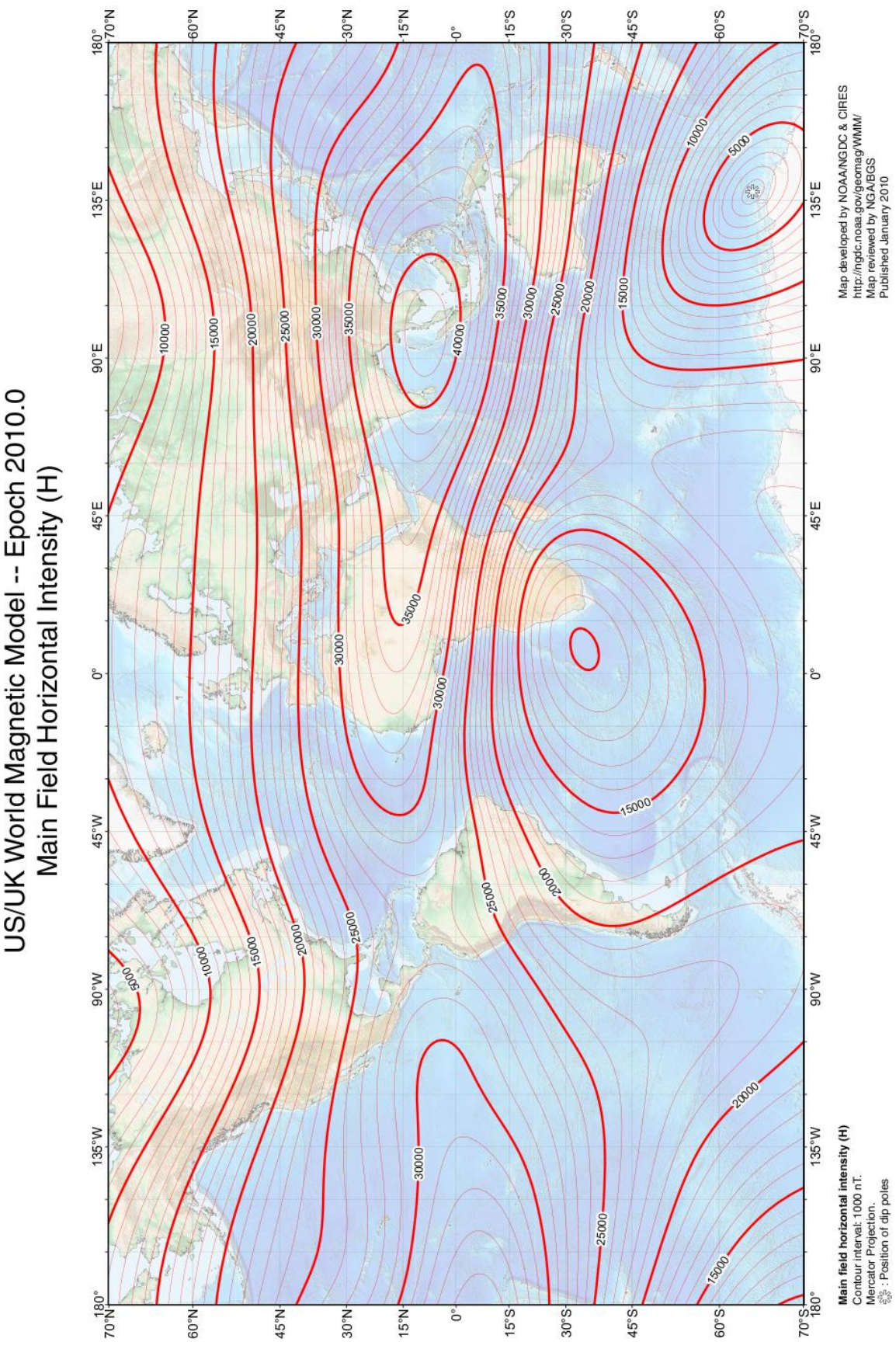
Appendix A

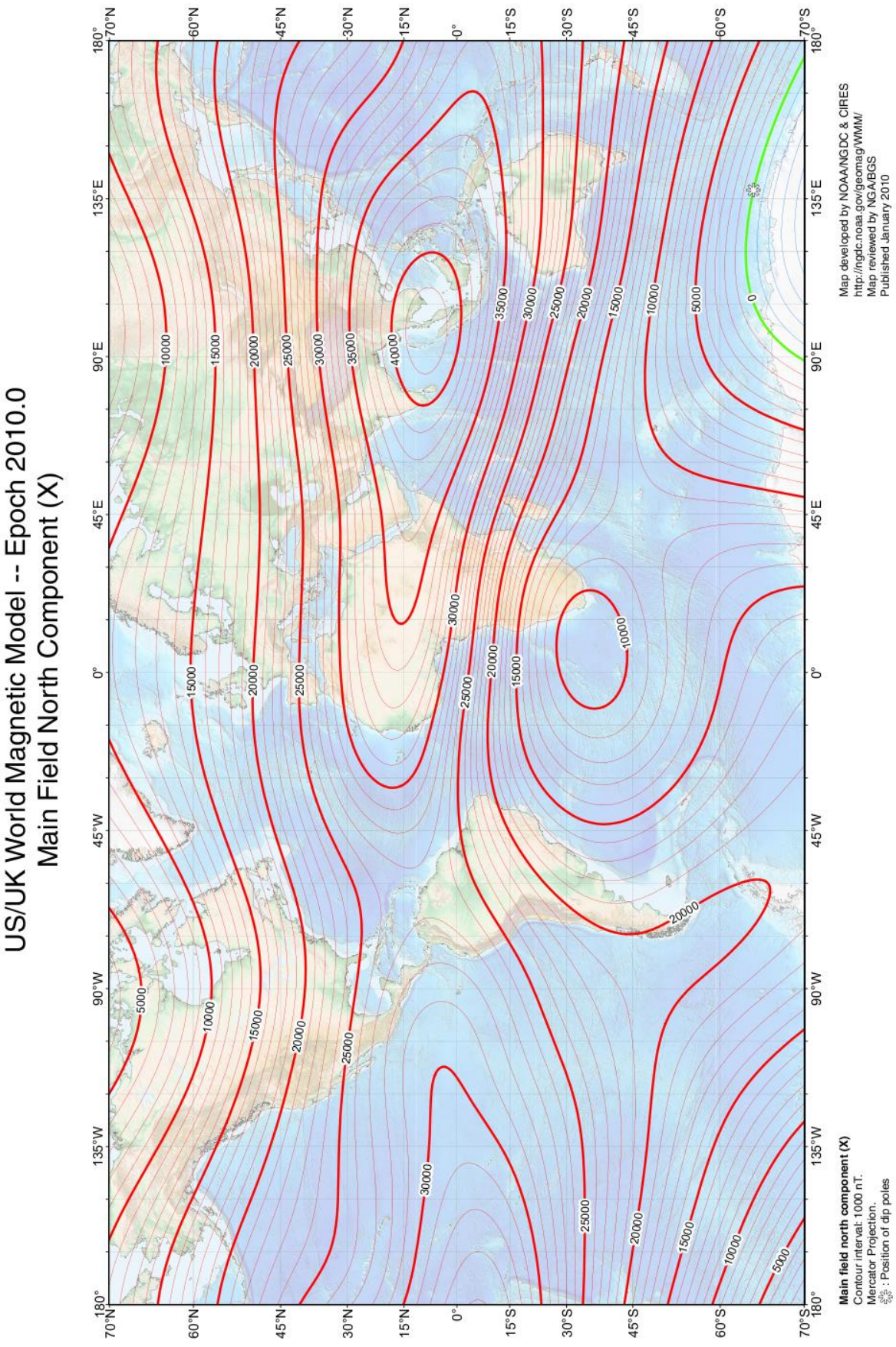
Abbreviation	Meaning/Definition
LSB	Least significant bit
m	Metre
MHz	Megahertz (1E+06 Hertz)
nT	Nanotesla (1E-09 Tesla)
PPM	Proton precession magnetometer
PTOL	Point take off and land
PWM	Pulse Width Modulation
RPM	Revolutions per minute
SCC	Synchronous Serial Controller
SPI	Serial Peripheral Interface
SPS	Samples per second
SQUID	Superconducting quantum interference device
T	Tesla
TMI	Total Magnetic Intensity
UART	Universal Asynchronous Receiver Transmitter
UAV	Unmanned autonomous vehicle
USD	United States Dollar(s)
V	Volt
VTOL	Vertical table off and land
W	Watt
Wh	Watt-hour
ZAR	South African Rand(s)

Appendix B – World Magnetic Model Data, Epoch 2010.0

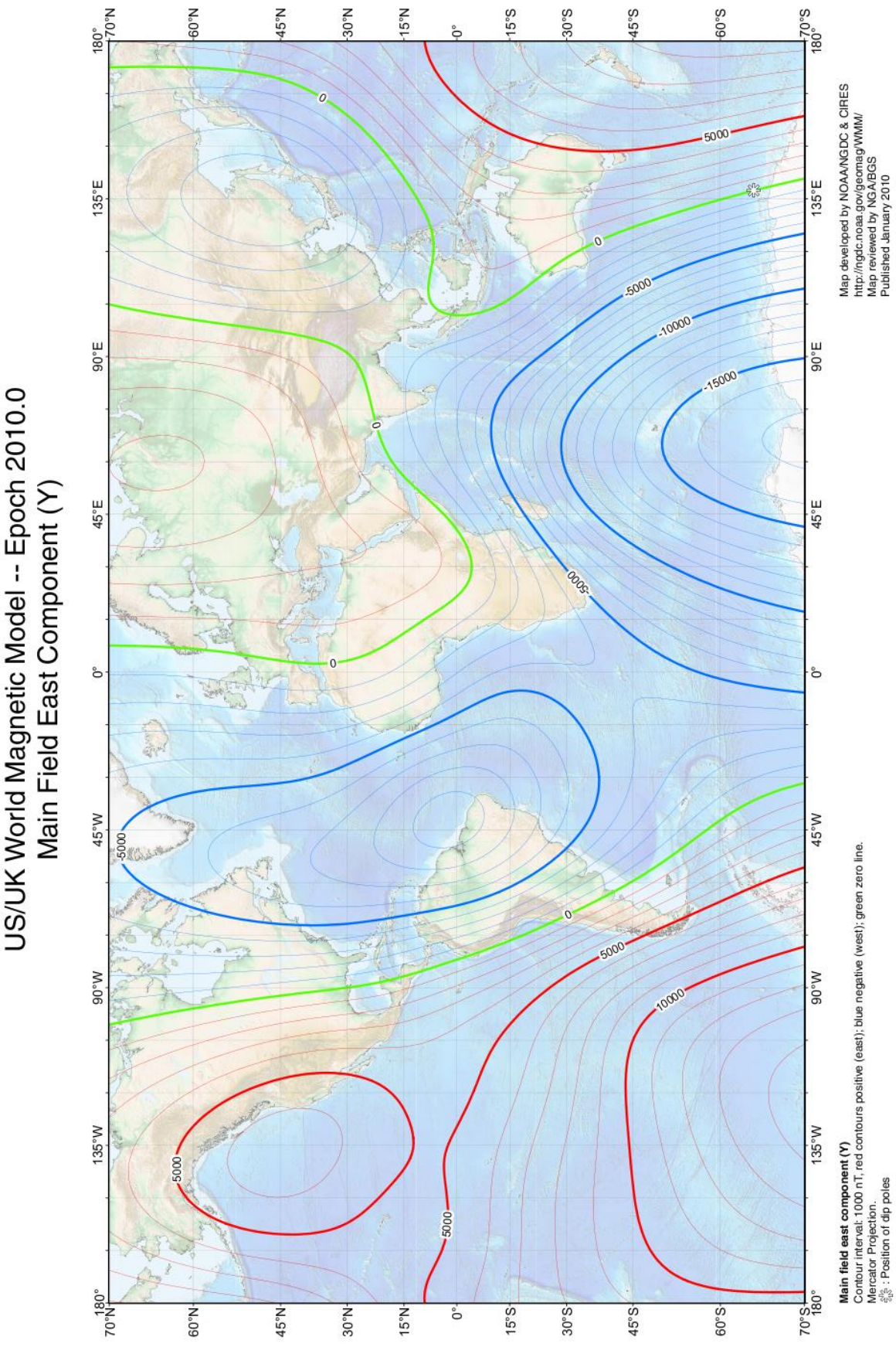
Figures in Appendix A are taken from the National Geophysical Data Center [11].


US/UK World Magnetic Model -- Epoch 2010.0
Main Field Total Intensity (F)
Map developed by NOAA/NGDC & CIRES
http://ngdc.noaa.gov/geomag/WMM/
Map reviewed by NGA/BGS
Published January 2010
Main Field Total Intensity (F)
Contour interval: 1000 nT.
Mercator Projection.
: Position of dip poles

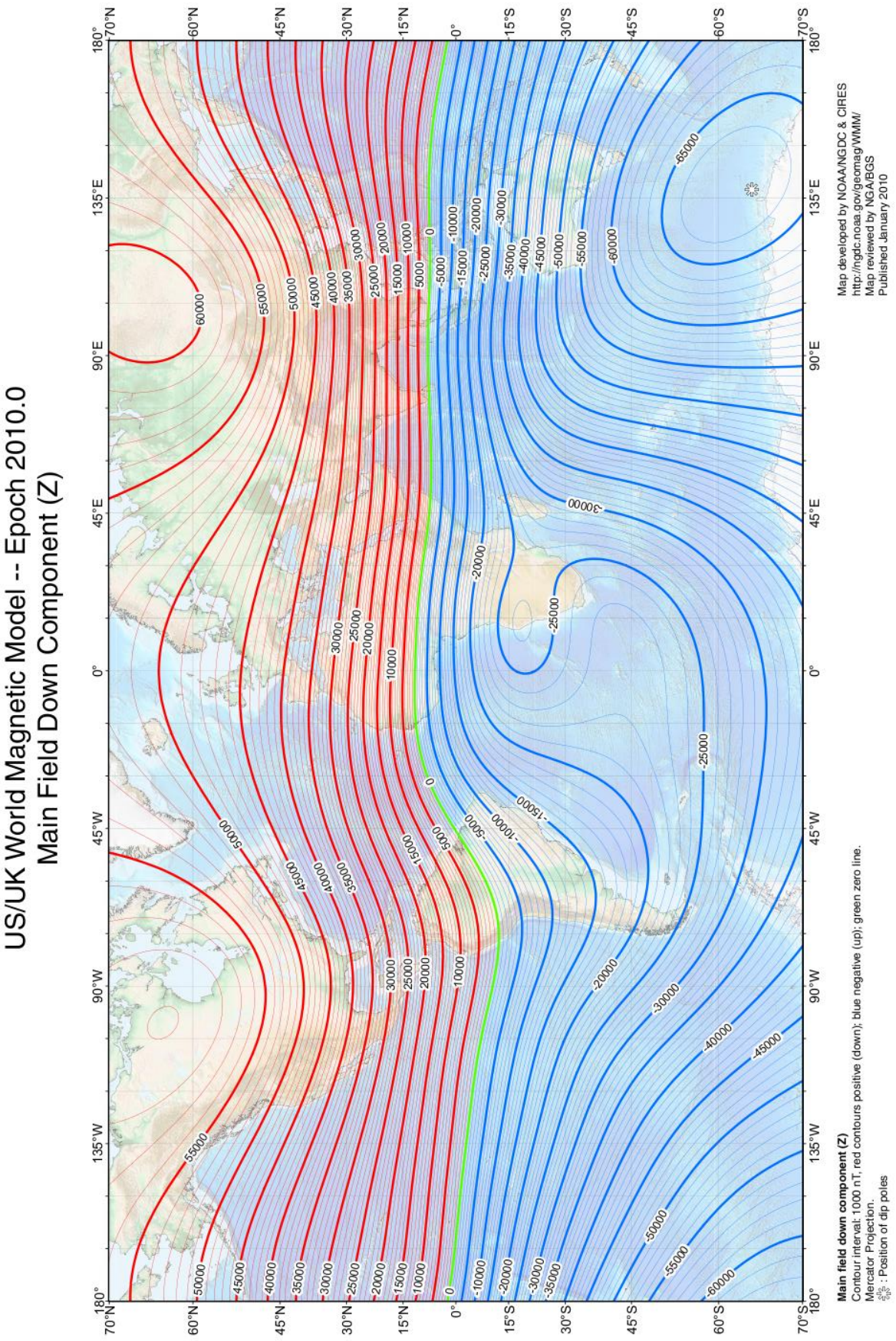

US/UK World Magnetic Model -- Epoch 2010.0
Main Field Horizontal Intensity (H)
Map developed by NOAA/NGDC & CIRES
http://ngdc.noaa.gov/geomag/WMM/
Map reviewed by NGA/BGS
Published January 2010
Main field horizontal intensity (H)
Contour interval: 1000 nT.
Mercator Projection.
: Position of dip poles

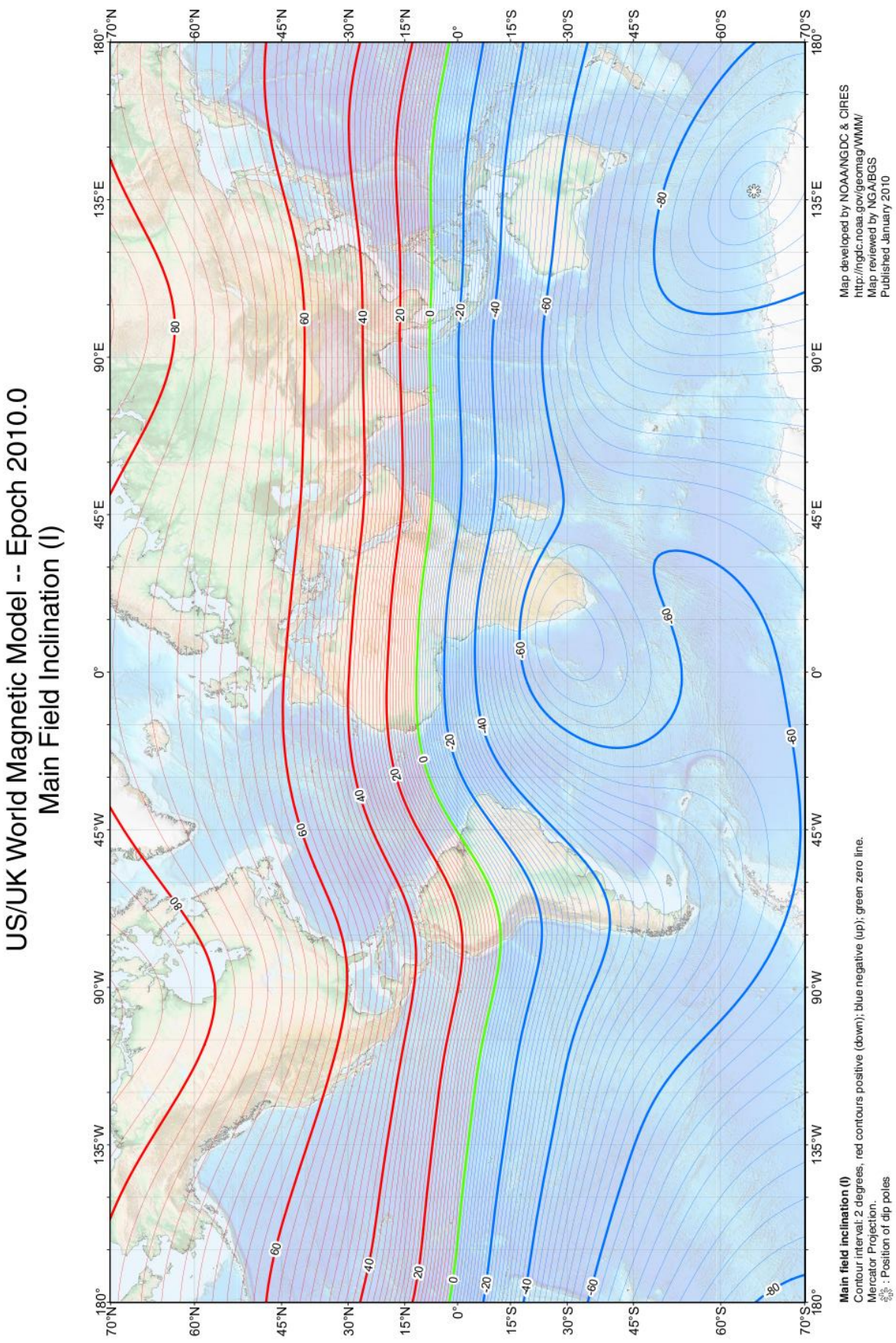

US/UK World Magnetic Model -- Epoch 2010.0
Main Field North Component (X)
Map developed by NOAA/NGDC & CIRES
http://ngdc.noaa.gov/geomag/WMM/
Map reviewed by NGA/BGS
Published January 2010
Main field north component (X)
Contour interval: 1000 nT.
Mercator Projection.
: Position of dip poles

Appendix B

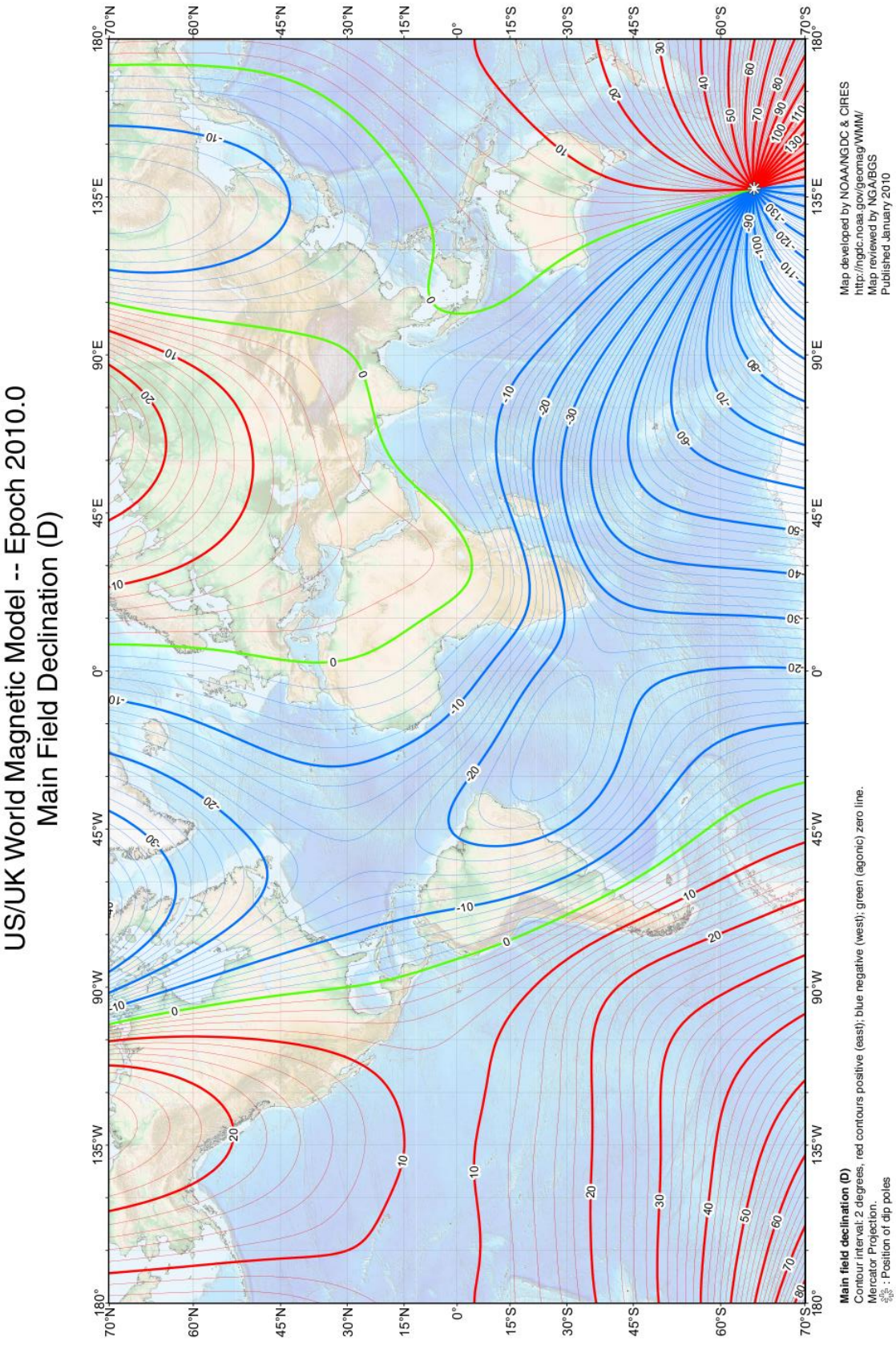

US/UK World Magnetic Model -- Epoch 2010.0
Main Field East Component (Y)
Main field east component (Y)
Contour interval: 1000 nT, red contours positive (east); blue negative (west); green zero line.
Mercator Projection.
: Position of dip poles
Map developed by NOAA/NGDC & CIRES
http://ngdc.noaa.gov/geomag/WMM/
Map reviewed by NGA/BGS
Published January 2010

Appendix B


US/UK World Magnetic Model -- Epoch 2010.0
Main Field Down Component (Z)
Main field down component (Z)
Contour interval: 1000 nT, red contours positive (down); blue negative (up); green zero line.
Mercator Projection.
: Position of dip poles
Map developed by NOAA/NGDC & CIRES
http://ngdc.noaa.gov/geomag/WMM/
Map reviewed by NGA/BGS
Published January 2010


US/UK World Magnetic Model -- Epoch 2010.0
Main Field Inclination (I)
Main field inclination (I)
Contour interval: 2 degrees, red contours positive (down); blue negative (up); green zero line.
Mercator Projection.
: Position of dip poles
Map developed by NOAA/NGDC & CIRES
http://ngdc.noaa.gov/geomag/WMM/
Map reviewed by NGA/BGS
Published January 2010

Appendix B


US/UK World Magnetic Model -- Epoch 2010.0
Main Field Declination (D)
Map developed by NOAA/NGDC & CIRES
http://ngdc.noaa.gov/geomag/WMM/
Map reviewed by NGA/BGS
Published January 2010
Main field declination (D)
Contour interval: 2 degrees, red contours positive (east); blue negative (west); green (agonic) zero line.
Mercator Projection.
: Position of dip poles

Appendix C – UAV Construction

The following pictures show the construction of various elements the UAV platform.

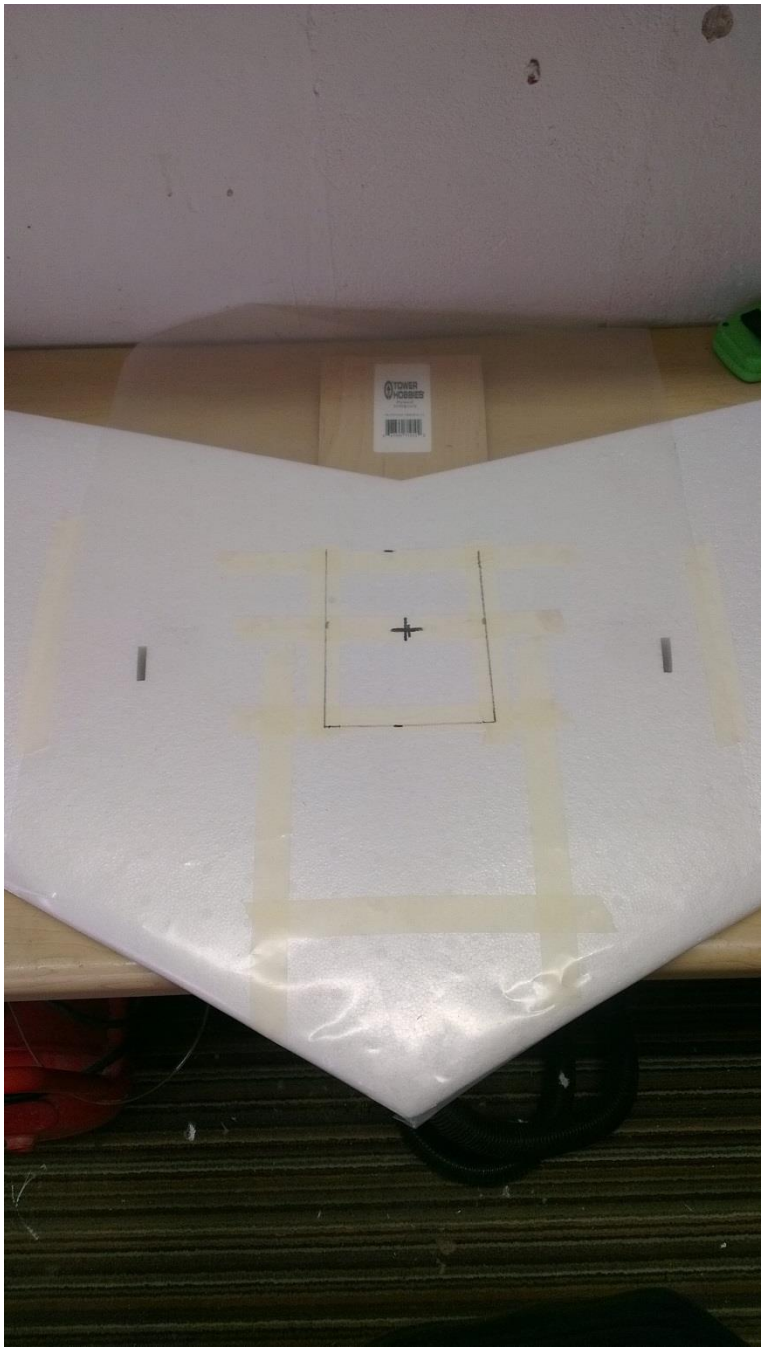


Figure C - 1: Marking out the area to be cut out and removed to mount the sunken fuselage. Note the "+" pencil mark showing the centre of gravity for the *Queen Bee*.

Figure C - 2: Laying out the carbon fibre sheets for cutting using a template. The baseplate sheet is shown here.

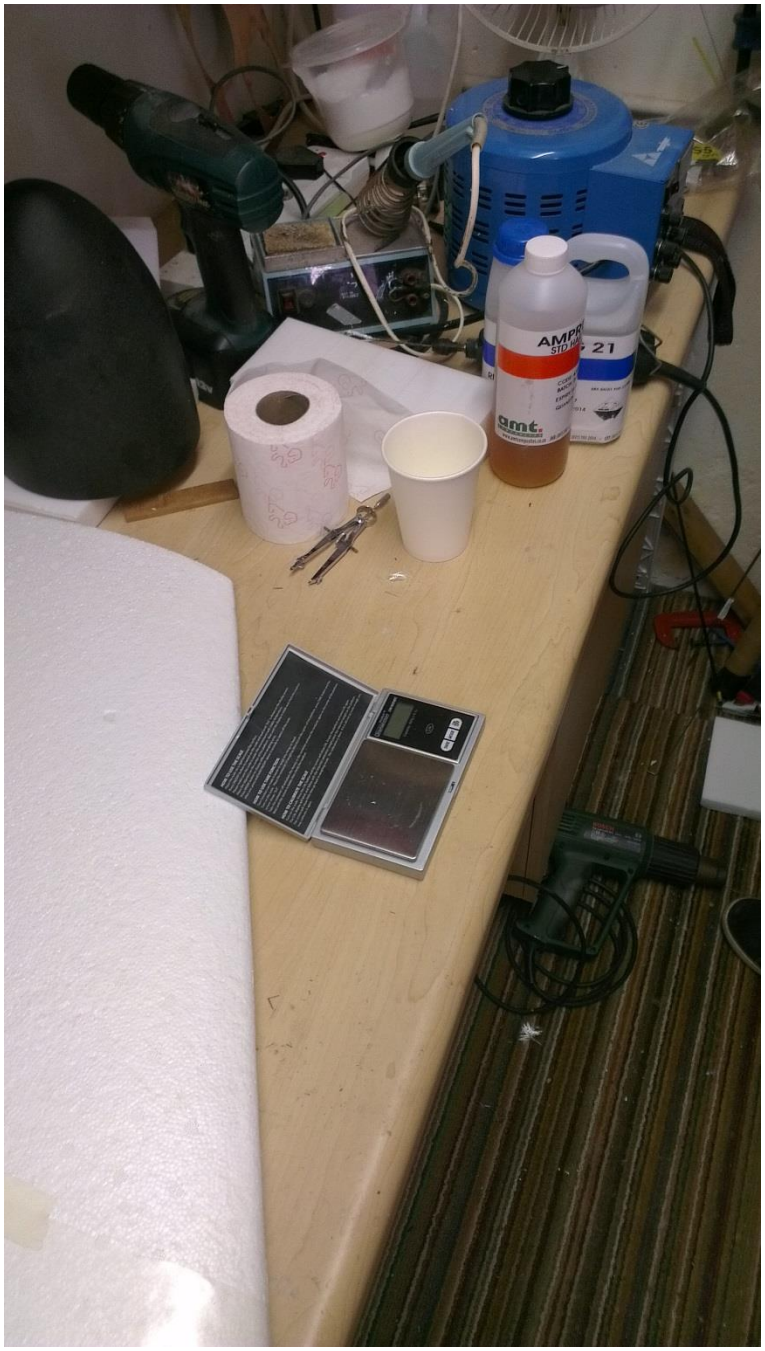


Figure C - 3: The scale used to measure the resin components. This is done to ensure the correct mix of the resin to allow for sufficient bonding of the various cloth elements.

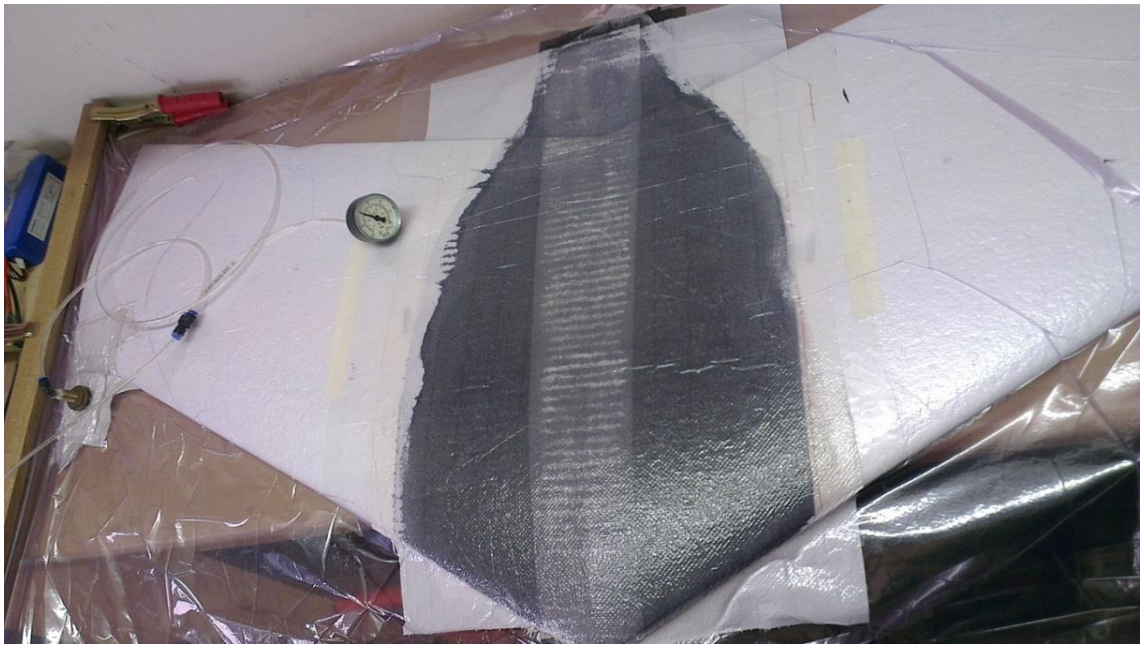


Figure C - 4: Construction of the moulded fuselage baseplate moulded to the shape of the EPP foam *Queen Bee* shape. Note the vacuum bag used in conjunction with heat (not shown) for curing the carbon fibre resin construction.

Figure C - 5: Cutting the foam for shaping of the fuselage over the baseplate. This foam is used to create the shape of the fuselage by hand to which the carbon fibre composite was then shaped and set. The fuselage was then bonded to the baseplate shown in Figure C - 2 above.

Figure C - 6: Initial full throttle propeller tests with a *Watt's Up* power meter. Current draw at full throttle on the LiPo 4S batteries was tested.

Appendix D – PWM Throttle Controller and Low Pass Filter

To aid with the initial propulsion system characterisation a PWM throttle controller was constructed using two 555-timer ICs and a potentiometer. A picture (see Figure D - 1 below) of the controller along with the circuit diagram (Figure D - 2) is presented below.

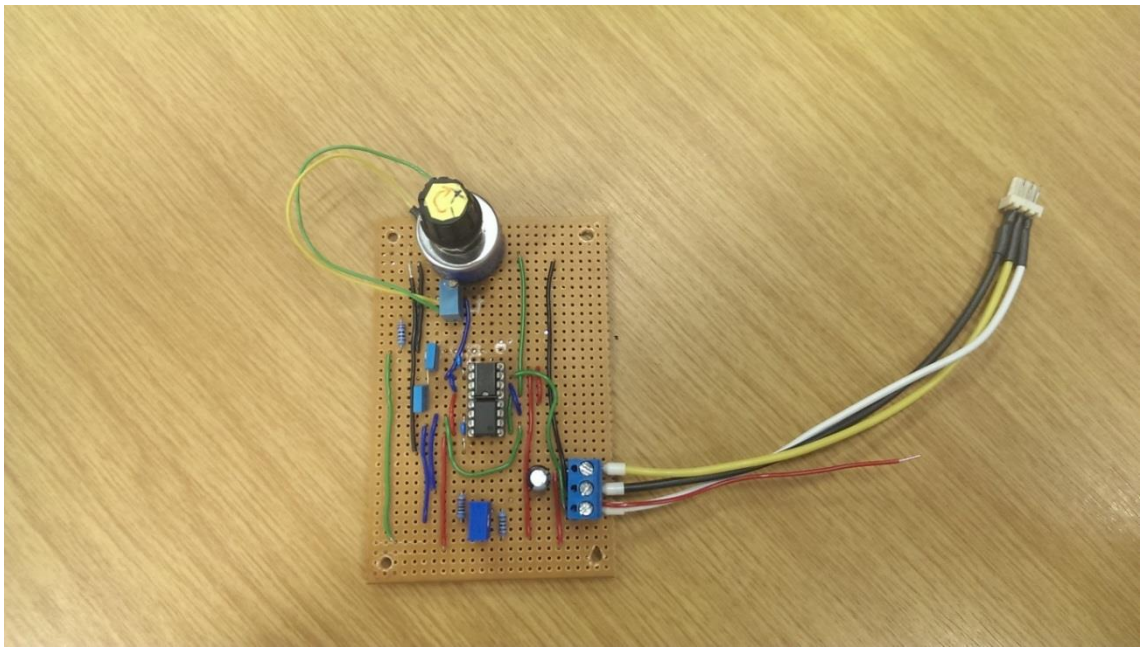


Figure D - 1: Picture of the PWM throttle controller, constructed using two 555-timer ICs and a potentiometer. This was later replaced by the radio controlled handset PWM throttle control.

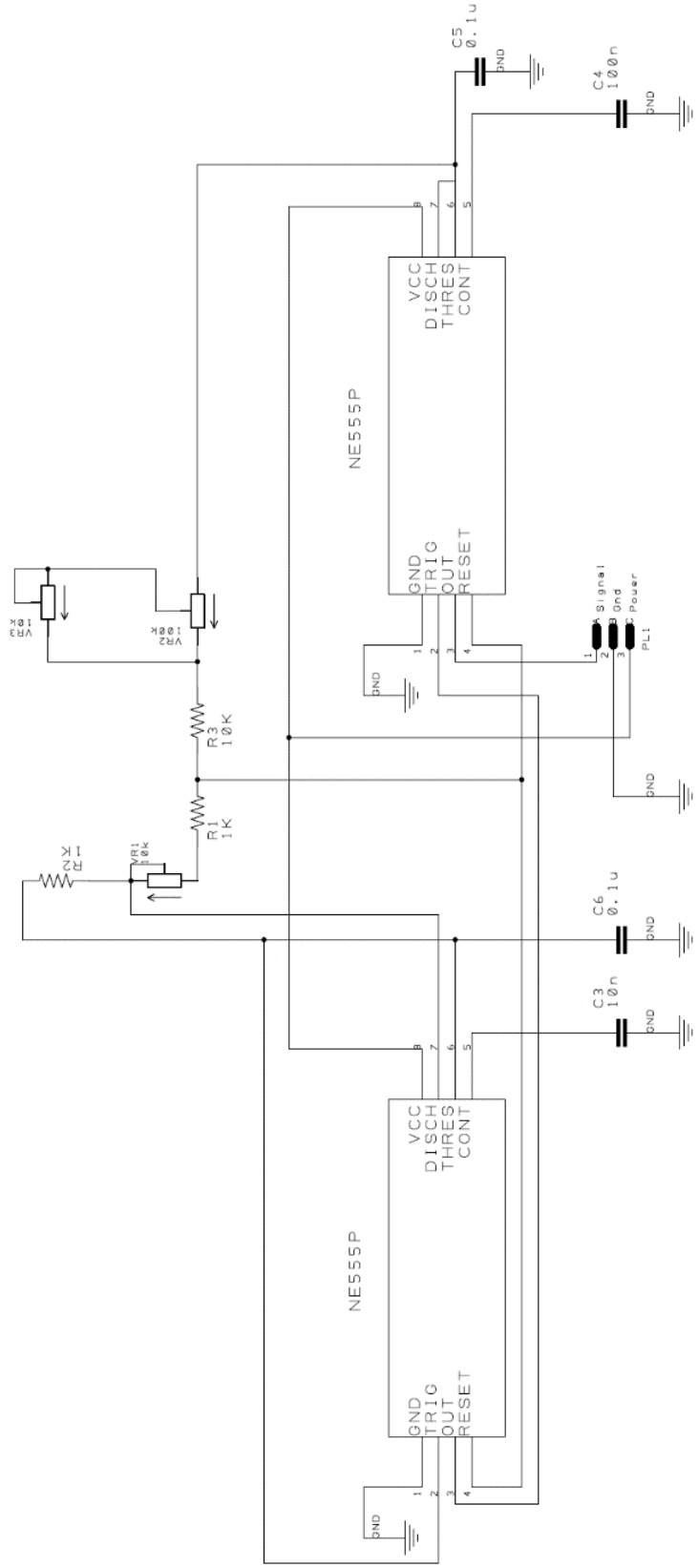


Figure D - 2: PWM throttle controller circuit diagram.

To successfully filter out noise on the back-EMF signal a simple second order low pass RC filter was constructed. This was placed in line with two of the phase wires of the DCBL motor and the oscilloscope. This allowed for the determination of the commutation frequency of the DCBL motor.

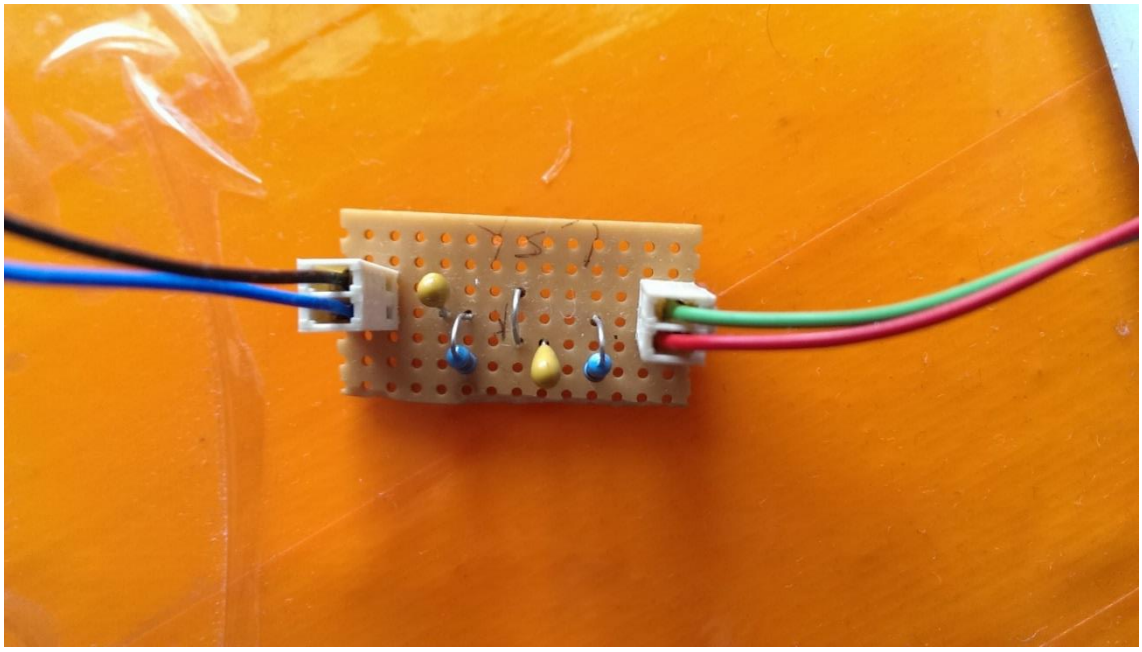


Figure D - 3: 2nd order low pass RC filter. Red and green wires are the filter input and the black and blue lines are the output (attached to the oscilloscope see Figure 8-3 above). This filter was used to filter out noise on the DCBL back-EMF between two phases on an oscilloscope. The cut off frequency for the filter is approximately 1.59 kHz.

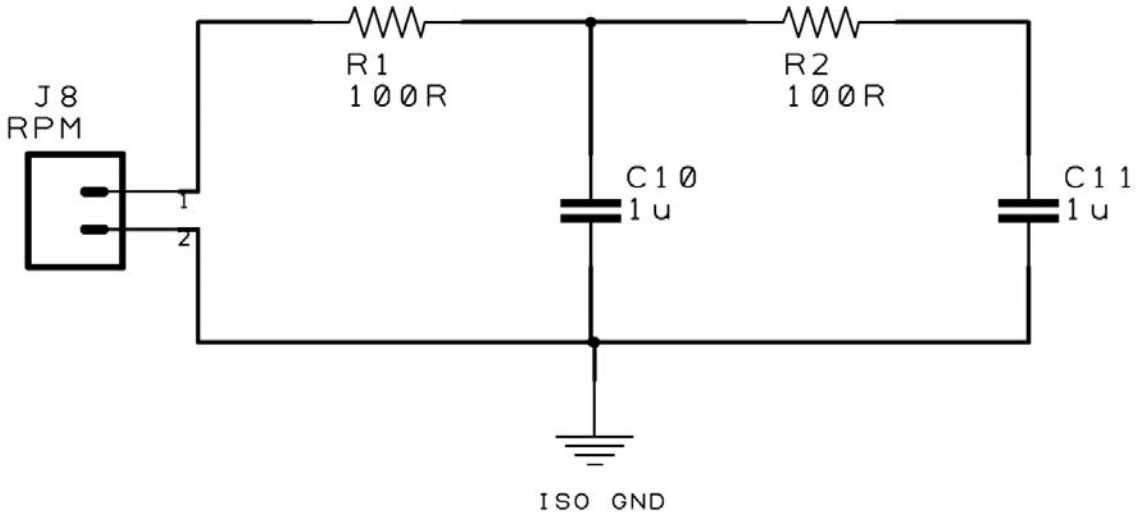


Figure D - 4: 2nd order low pass RC filter circuit diagram. f_c = 1.59 kHz.

Appendix E – Interface (IF) Board Construction

The PCB top, bottom and combined layers (Figure E - 1, Figure E - 2, Figure E - 3) with the IF board circuit diagram (Figure E - 4) are presented below.

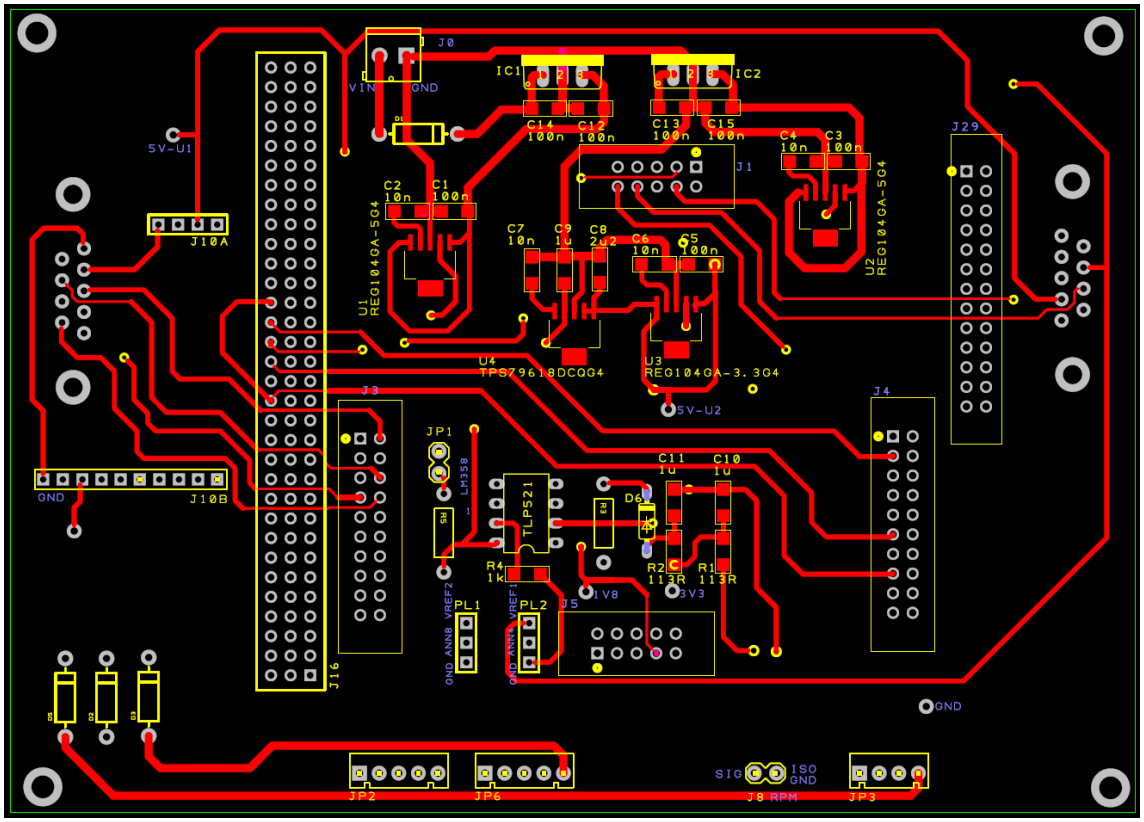


Figure E - 1: Top layer of the IF board PCB.

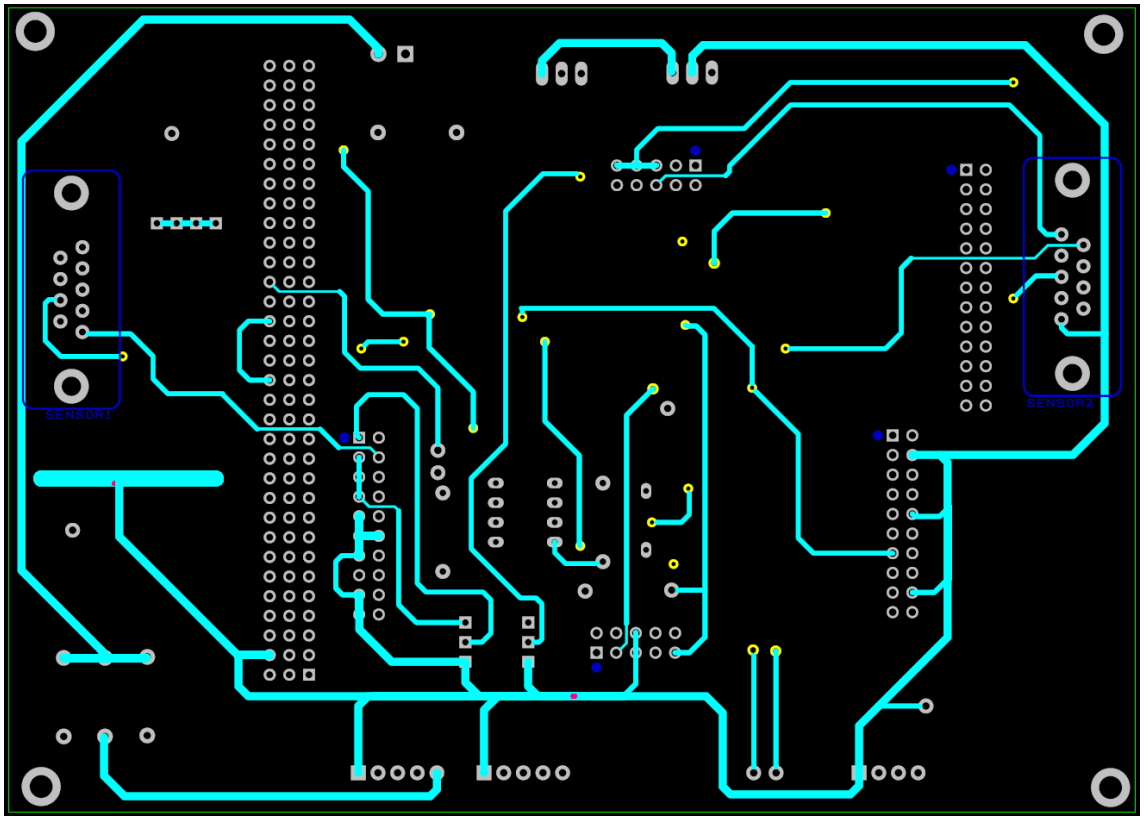


Figure E - 2: Bottom layer of the IF board PCB.

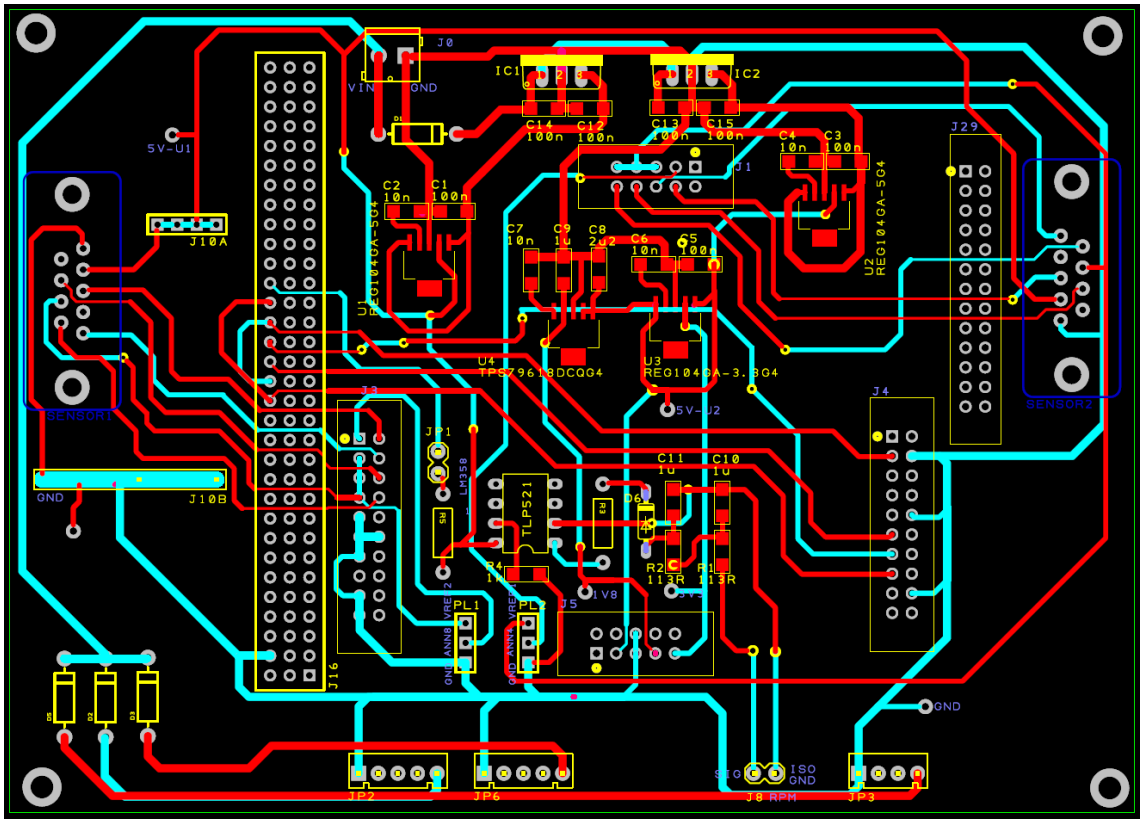


Figure E - 3: Both layers of the IF board PCB combined.

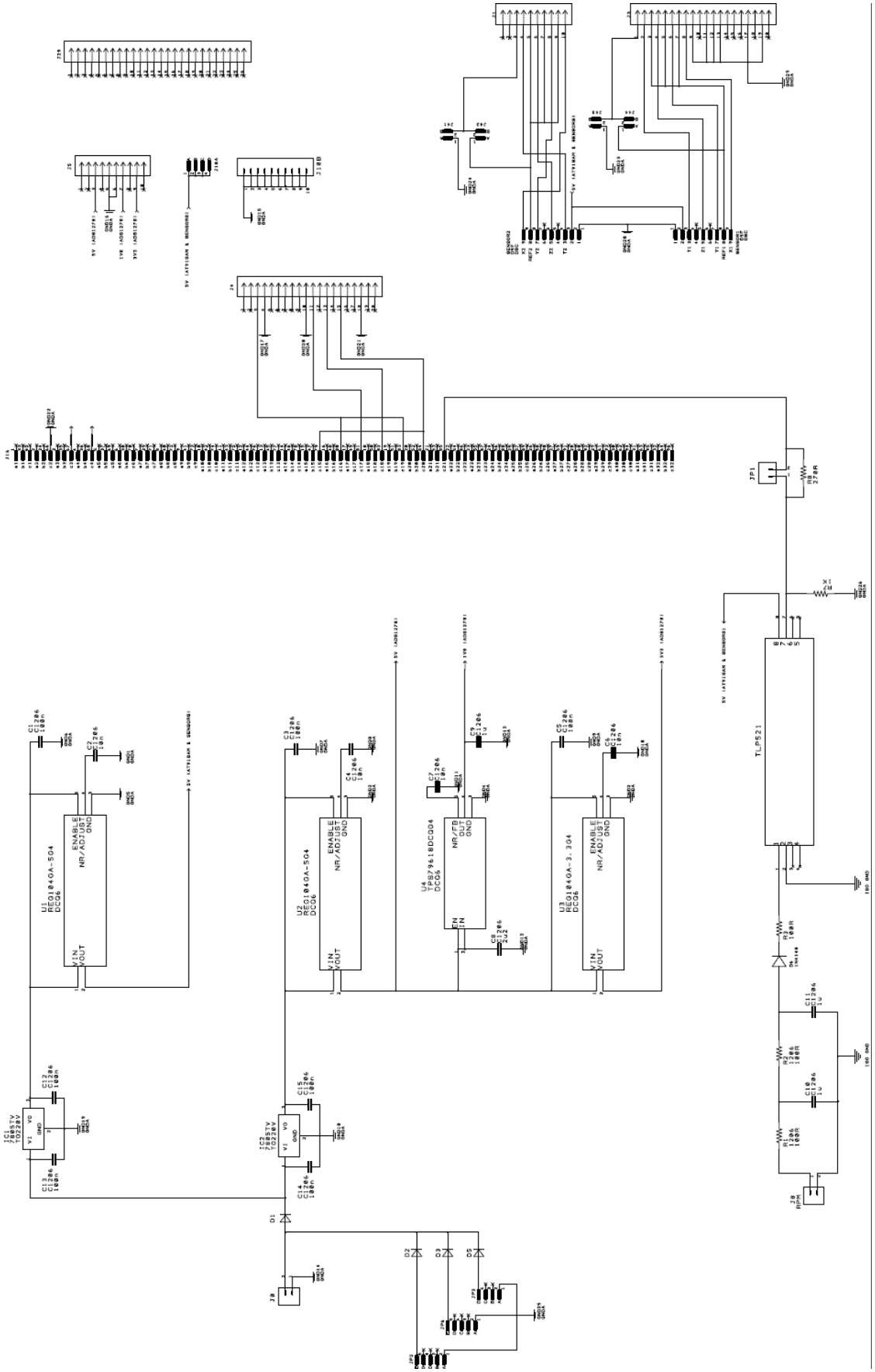


Figure E - 4: IF board circuit diagram.

Appendix F – µTasker Datalogger Source Code Extract

The following code extract was used to perform the datalogger function shown in Figure 6-2 above. The code was run on the Atmel AT91SAM7X-EK platform. The code is used within the µTasker operating system framework [57], [56], [73], [74], [75] for the development board. Microsoft Visual Studio 2005 was used as the integrated development environment (IDE).

```
/* ===================================================================== */
/*                           include files                               */
/* ===================================================================== */

#include "config.h"

#define BLINK_LED2      PB20
#define BLINK_LED3      PB21
#define BLINK_LED4      PB22

#define MAX_SAMPLES        512
#define MAX_CHANNELS       8

#define TOGGLE_LED2()    _TOGGLE_PORT(B, BLINK_LED2)               // {26} blink
the LED, if set as output
#define TOGGLE_LED3()    _TOGGLE_PORT(B, BLINK_LED3)               // {26} blink
the LED, if set as output
#define TOGGLE_LED4()    _TOGGLE_PORT(B, BLINK_LED4)               // {26} blink
the LED, if set as output

//Supported GPS NMEA Sentences
#define NMEA_SEN_GLL       "$GPGLL"                                        //
GPGLL interval - Geographic Position - Latitude longitude
#define NMEA_SEN_RMC       "$GPRMC"                                        //
GPRMC interval - Recomended Minimum Specific GNSS Sentence
#define NMEA_SEN_VTG       "$GPVTG"                                        //
GPVTG interval - Course Over Ground and Ground Speed
#define NMEA_SEN_GGA       "$GPGGA"                                        //
GPGGA interval - GPS Fix Data
#define NMEA_SEN_GSA       "$GPGSA"                                        //
GPGSA interval - GNSS DOPS and Active Satellites
#define NMEA_SEN_GSV       "$GPGSV"                                        //
GPGSV interval - GNSS Satellites in View
#define NMEA_SEN_GRS       "$GPGRS"                                        //
GPGRS interval - GNSS Range Residuals
#define NMEA_SEN_GST       "$GPGST"                                        //
GPGST interval - GNSS Pseudorange Erros Statistics
#define NMEA_SEN_ZDA       "$GPZDA"                                        //
GPZDA interval - Time & Date
```

```
//Supported PMTK NMEA Sentences
#define NMEA_SEN_MALM      "$PMTKALM"                                          //
PMTKALM interval - GPS almanac information
#define NMEA_SEN_MEPH      "$PMTKEPH"                                          //
PMTKEPH interval - GPS ephmeris information
#define NMEA_SEN_MDGP      "$PMTKDGP"                                          //
PMTKDGP interval - GPS differential correction information
#define NMEA_SEN_MDBG      "$PMTKDBG"                                          //
PMTKDBG interval - MTK debug information
#define NMEA_SEN_MCHN      "$PMTKCHN"                                          //
PMTKCHN interval - GPS channel status

static UTDIRECTORY     *ptr_utMyDirectory = 0;
static UTFILE              utMyFile = {0};

QUEUE_HANDLE               qhMySSCHandle = {0};                          // My SSC
handle
QUEUE_HANDLE               qhMyUARTHandle = {0};                         // My UART
handle

static const int           DummyMsg[1] = {0xaaaaaaaa};

static signed int          raw_data[MAX_CHANNELS] = {0};
static double              mean_raw_data[MAX_CHANNELS] = {0};

static double              mag_data[MAX_CHANNELS] = {0};
static double              mean_mag_data[MAX_CHANNELS] = {0};

static const float         S = 36.5;                                           //
[V/nT] Sensitivity

static const signed int    B0[] = {-123, -54, -136, 0, 4, -93, -148, 0};  // [nT]
Zero offest

static const float         Vmax = 2.25;                                  // [V]
Maximum voltage range
static const float         Vmin = -2.25;                                 // [V]
Minimum voltage range

static const signed int Cmax = 0x007fffff;                               // Maximum
code (+2^23 - 1)
static const signed int Cmin = 0xff800000;                               // Minimum
code (-2^23)
```

```
void DisplayHexData()
{
	fnDebugMsg("------------------------------------------------------------------
----------------\n\r");
	fnDebugMsg("Last Raw Data Received:\n\r");

	fnDebugMsg("X1= ");
	fnDebugHex(raw_data[0], (WITH_LEADIN | WITH_SPACE | 4));
	fnDebugMsg("\n\r");

	fnDebugMsg("Y1= ");
	fnDebugHex(raw_data[1], (WITH_LEADIN | WITH_SPACE | 4));
	fnDebugMsg("\n\r");

	fnDebugMsg("Z1= ");
	fnDebugHex(raw_data[2], (WITH_LEADIN | WITH_SPACE | 4));
	fnDebugMsg("\n\r");

	fnDebugMsg("T1= ");
	fnDebugHex(raw_data[3], (WITH_LEADIN | WITH_SPACE | 4));
	fnDebugMsg("\n\r");

	fnDebugMsg("X2= ");
	fnDebugHex(raw_data[4], (WITH_LEADIN | WITH_SPACE | 4));
	fnDebugMsg("\n\r");

	fnDebugMsg("Y2= ");
	fnDebugHex(raw_data[5], (WITH_LEADIN | WITH_SPACE | 4));
	fnDebugMsg("\n\r");

	fnDebugMsg("Z2= ");
	fnDebugHex(raw_data[6], (WITH_LEADIN | WITH_SPACE | 4));
	fnDebugMsg("\n\r");

	fnDebugMsg("T2= ");
	fnDebugHex(raw_data[7], (WITH_LEADIN | WITH_SPACE | 4));
	fnDebugMsg("\n\r");

	fnDebugMsg("------------------------------------------------------------------
----------------\n\r");

	fnDebugMsg("Mean Data:\n\r");

	fnDebugMsg("X1= ");
	fnDebugHex((signed int) mean_raw_data[0], (WITH_LEADIN | WITH_SPACE | 4));
	fnDebugMsg("\n\r");

	fnDebugMsg("Y1= ");
	fnDebugHex((signed int) mean_raw_data[1], (WITH_LEADIN | WITH_SPACE | 4));
	fnDebugMsg("\n\r");

	fnDebugMsg("Z1= ");
	fnDebugHex((signed int) mean_raw_data[2], (WITH_LEADIN | WITH_SPACE | 4));
	fnDebugMsg("\n\r");

	fnDebugMsg("T1= ");
	fnDebugHex((signed int) mean_raw_data[3], (WITH_LEADIN | WITH_SPACE | 4));
	fnDebugMsg("\n\r");

	fnDebugMsg("X2= ");
	fnDebugHex((signed int) mean_raw_data[4], (WITH_LEADIN | WITH_SPACE | 4));
	fnDebugMsg("\n\r");

	fnDebugMsg("Y2= ");
	fnDebugHex((signed int) mean_raw_data[5], (WITH_LEADIN | WITH_SPACE | 4));
	fnDebugMsg("\n\r");

	fnDebugMsg("Z2= ");
```

```
	fnDebugHex((signed int) mean_raw_data[6], (WITH_LEADIN | WITH_SPACE | 4));
	fnDebugMsg("\n\r");

	fnDebugMsg("T2= ");
	fnDebugHex((signed int) mean_raw_data[7], (WITH_LEADIN | WITH_SPACE | 4));
	fnDebugMsg("\n\r");

	fnDebugMsg("------------------------------------------------------------------
----------------\n\r\n\r");
}

void DisplayMagData()
{
	fnDebugMsg("------------------------------------------------------------------
----------------\n\r");
	fnDebugMsg("Last Mag Data Received:\n\r");

	fnDebugMsg("X1= ");
	fnDebugDec(mag_data[0], (DISPLAY_NEGATIVE));
	fnDebugMsg(" nT\n\r");

	fnDebugMsg("Y1= ");
	fnDebugDec(mag_data[1], (DISPLAY_NEGATIVE));
	fnDebugMsg(" nT\n\r");

	fnDebugMsg("Z1= ");
	fnDebugDec(mag_data[2], (DISPLAY_NEGATIVE));
	fnDebugMsg(" nT\n\r");

	fnDebugMsg("X2= ");
	fnDebugDec(mag_data[4], (DISPLAY_NEGATIVE));
	fnDebugMsg(" nT\n\r");

	fnDebugMsg("Y2= ");
	fnDebugDec(mag_data[5], (DISPLAY_NEGATIVE));
	fnDebugMsg(" nT\n\r");

	fnDebugMsg("Z2= ");
	fnDebugDec(mag_data[6], (DISPLAY_NEGATIVE));
	fnDebugMsg(" nT\n\r");

	fnDebugMsg("------------------------------------------------------------------
----------------\n\r");

	fnDebugMsg("Mean Mag Data:\n\r");

	fnDebugMsg("X1= ");
	fnDebugDec((signed int) mean_mag_data[0], (DISPLAY_NEGATIVE));
	fnDebugMsg(" nT\n\r");

	fnDebugMsg("Y1= ");
	fnDebugDec((signed int) mean_mag_data[1], (DISPLAY_NEGATIVE));
	fnDebugMsg(" nT\n\r");

	fnDebugMsg("Z1= ");
	fnDebugDec((signed int) mean_mag_data[2], (DISPLAY_NEGATIVE));
	fnDebugMsg(" nT\n\r");

	fnDebugMsg("X2= ");
	fnDebugDec((signed int) mean_mag_data[4], (DISPLAY_NEGATIVE));
	fnDebugMsg(" nT\n\r");

	fnDebugMsg("Y2= ");
	fnDebugDec((signed int) mean_mag_data[5], (DISPLAY_NEGATIVE));
	fnDebugMsg(" nT\n\r");

	fnDebugMsg("Z2= ");
	fnDebugDec((signed int) mean_mag_data[6], (DISPLAY_NEGATIVE));
```

```
        fnDebugMsg(" nT\n\r");

        fnDebugMsg("-------------------------------------------------------------------
----------------\n\r\n\r");
}

void DisplayDecData()
{
        fnDebugMsg("-------------------------------------------------------------------
----------------\n\r");
        fnDebugMsg("Last Raw Data Received:\n\r");

        fnDebugMsg("X1= ");
        fnDebugDec(raw_data[0], (DISPLAY_NEGATIVE));
        fnDebugMsg("\n\r");

        fnDebugMsg("Y1= ");
        fnDebugDec(raw_data[1], (DISPLAY_NEGATIVE));
        fnDebugMsg("\n\r");

        fnDebugMsg("Z1= ");
        fnDebugDec(raw_data[2], (DISPLAY_NEGATIVE));
        fnDebugMsg("\n\r");

        fnDebugMsg("T1= ");
        fnDebugDec(raw_data[3], (DISPLAY_NEGATIVE));
        fnDebugMsg("\n\r");

        fnDebugMsg("X2= ");
        fnDebugDec(raw_data[4], (DISPLAY_NEGATIVE));
        fnDebugMsg("\n\r");

        fnDebugMsg("Y2= ");
        fnDebugDec(raw_data[5], (DISPLAY_NEGATIVE));
        fnDebugMsg("\n\r");

        fnDebugMsg("Z2= ");
        fnDebugDec(raw_data[6], (DISPLAY_NEGATIVE));
        fnDebugMsg("\n\r");

        fnDebugMsg("T2= ");
        fnDebugDec(raw_data[7], (DISPLAY_NEGATIVE));
        fnDebugMsg("\n\r");

        fnDebugMsg("-------------------------------------------------------------------
----------------\n\r");

        fnDebugMsg("Mean Data:\n\r");

        fnDebugMsg("X1= ");
        fnDebugDec((signed int) mean_raw_data[0], (DISPLAY_NEGATIVE));
        fnDebugMsg("\n\r");

        fnDebugMsg("Y1= ");
        fnDebugDec((signed int) mean_raw_data[1], (DISPLAY_NEGATIVE));
        fnDebugMsg("\n\r");

        fnDebugMsg("Z1= ");
        fnDebugDec((signed int) mean_raw_data[2], (DISPLAY_NEGATIVE));
        fnDebugMsg("\n\r");

        fnDebugMsg("T1= ");
        fnDebugDec((signed int) mean_raw_data[3], (DISPLAY_NEGATIVE));
        fnDebugMsg("\n\r");

        fnDebugMsg("X2= ");
        fnDebugDec((signed int) mean_raw_data[4], (DISPLAY_NEGATIVE));
        fnDebugMsg("\n\r");
```

```
	fnDebugMsg("Y2= ");
	fnDebugDec((signed int) mean_raw_data[5], (DISPLAY_NEGATIVE));
	fnDebugMsg("\n\r");

	fnDebugMsg("Z2= ");
	fnDebugDec((signed int) mean_raw_data[6], (DISPLAY_NEGATIVE));
	fnDebugMsg("\n\r");

	fnDebugMsg("T2= ");
	fnDebugDec((signed int) mean_raw_data[7], (DISPLAY_NEGATIVE));
	fnDebugMsg("\n\r");

	fnDebugMsg("------------------------------------------------------------------
----------------\n\r\n\r");
}

UTFILE fnOpenFile(void)
{
	if (ptr_utMyDirectory == 0) {
		ptr_utMyDirectory = utAllocateDirectory(DISK_D, 256);
	}

	ptr_utMyDirectory->usDirectoryFlags &= ~UTDIR_VALID;

	if (ptr_utMyDirectory->usDirectoryFlags & UTDIR_VALID)
	{
		fnDebugMsg("Directory is valid\r\n");
	}

	if (utOpenDirectory(0, ptr_utMyDirectory) != UTFAT_SUCCESS)
	{	// open the root directory
		//Error
		TOGGLE_LED3();
	}
	else
	{
		// Success
		utMyFile.ptr_utDirObject = ptr_utMyDirectory;
	}

	if (utOpenFile("data.txt", &utMyFile, UTFAT_OPEN_FOR_WRITE | UTFAT_CREATE)
!= UTFAT_PATH_IS_FILE)
	{	// open a file for writing and create if not existing
		//Error
		TOGGLE_LED3();
	}

	return utMyFile;
}

void ResetData()
{
	int i = 0;

	for (i = 0; i < MAX_CHANNELS; i++)
	{
		raw_data[i] = 0;
		mean_raw_data[i] = 0;

		mag_data[i] = 0;
		mean_mag_data[i] = 0;
	}
}
```

```
void WriteGPSTimeStampToFile(char *GPSTimeStamp, int numchars)
{
	utSeek(&utMyFile, 0, UTFAT_SEEK_END);

	if (utWriteFile(&utMyFile, GPSTimeStamp, numchars) != UTFAT_SUCCESS)
	{
		//Error
		TOGGLE_LED3();
	}
}

void WriteDecMeanRawDataToFile()
{
	int i = 0;
	unsigned short numchars = 0;

	char ucDataMsg[80] = {0};

	for(i = 0; i < MAX_CHANNELS; i++)
	{
		char ucTmpMsg[10] = {0};
		fnBufferDec(mean_raw_data[i], (DISPLAY_NEGATIVE), &ucTmpMsg);
		strcat(&ucDataMsg, ucTmpMsg);

		if(i < MAX_CHANNELS - 1)
		{
			strcat(&ucDataMsg, "\t");
		}
	}
	strcat(&ucDataMsg, "\r\n");

	i = 0;

	while (ucDataMsg[i] != '\0')
	{
		i++;
		numchars++;
	}

	utSeek(&utMyFile, 0, UTFAT_SEEK_END);

	if (utWriteFile(&utMyFile, &ucDataMsg, numchars) != UTFAT_SUCCESS)
	{
		// Error
		TOGGLE_LED3();
	}
}
```

```
void WriteDecRawDataToFile()
{
	int i = 0;
	unsigned short numchars = 0;

	char ucDataMsg[80] = {0};

	for(i = 0; i < MAX_CHANNELS; i++)
	{
		char ucTmpMsg[10] = {0};
		fnBufferDec(raw_data[i], (DISPLAY_NEGATIVE), &ucTmpMsg);
		strcat(&ucDataMsg, ucTmpMsg);

		if(i < MAX_CHANNELS - 1)
		{
			strcat(&ucDataMsg, "\t");
		}
	}
	strcat(&ucDataMsg, "\r\n");

	i = 0;

	while (ucDataMsg[i] != '\0')
	{
		i++;
		numchars++;
	}

	utSeek(&utMyFile, 0, UTFAT_SEEK_END);

	if (utWriteFile(&utMyFile, &ucDataMsg, numchars) != UTFAT_SUCCESS)
	{
		// Error
		TOGGLE_LED3();
	}
}
```

```
extern void fnMyUARTTask(TTASKTABLE *ptrTaskTable)
{
	int i, num = 0;
	static int count = 0;

	static unsigned char GPSQuality = '0';

	static unsigned char ucNMEAMsg[256] = {0};
	static unsigned char ucGPSMsgType[7] = {0};
	unsigned char ucInputChar;

	while (fnRead(qhMyUARTHandle, &ucInputChar, 1) != 0)
	{
		ucNMEAMsg[count++] = ucInputChar;

		if (ucInputChar == '\n')
		{
			ucNMEAMsg[count] = '\0';	// Terminate the NMEA string message
after the newline character

			for (i = 0; i < 6; i++)
			{
				ucGPSMsgType[i] = ucNMEAMsg[i];
			}

			if (!strcmp(NMEA_SEN_GGA, ucGPSMsgType))
			{
				i = 6;
				while((ucNMEAMsg[i] != '\n') && (num < 5))
				{
					if(ucNMEAMsg[i] == ',')
					{
						num++;
					}
					i++;
				}

				GPSQuality = ucNMEAMsg[i+1];

				if (GPSQuality != '0')
				{
					utMyFile = fnOpenFile();
					WriteGPSTimeStampToFile(&ucNMEAMsg, count);
					utCloseFile(&utMyFile);

					TOGGLE_LED4();

					fnWrite(qhMySSCHandle, (unsigned char *)DummyMsg,
1);	//transmit dummy message to ADS - this will trigger data from the ADS
				}
				else
				{
					//fnDebugMsg("GPS Quality Indicator = 0!!\r\n");
					fnDebugMsg(&ucNMEAMsg);
				}
			}
			count = 0;
		}
	}
}
```

```
extern void fnMySSCTask(TTASKTABLE *ptrTaskTable)
{
	int i = 0;

	static int count = 1;

	unsigned int usADS_data[6] = {0};

	signed int tmp_data[MAX_CHANNELS] = {0};

	while (fnRead(qhMySSCHandle, &usADS_data, 6) != 0)
	{
		if (count > MAX_SAMPLES)
		{
			utMyFile = fnOpenFile();
			WriteDecMeanRawDataToFile();
			utCloseFile(&utMyFile);

			count = 1;
			ResetData();

			TOGGLE_LED4();
			return;
		}

		//X1
		tmp_data[0] = ((usADS_data[0] >> 8) & 0x00ffffff);

		//Y1
		tmp_data[1] = ((usADS_data[0]  << 16) & 0x00ff0000) + ((usADS_data[1]
>> 16) & 0x0000ffff);

		//Z1
		tmp_data[2] = ((usADS_data[1]  << 8) & 0x00ffff00) + ((usADS_data[2]
>> 24) & 0x000000ff);

		//T1
		tmp_data[3] = (usADS_data[2]  & 0x00ffffff);

		//X2
		tmp_data[4] = ((usADS_data[3] >> 8) & 0x00ffffff);

		//Y2
		tmp_data[5] = ((usADS_data[3]  << 16) & 0x00ff0000) + ((usADS_data[4]
>> 16) & 0x0000ffff);

		//Z2
		tmp_data[6] = ((usADS_data[4]  << 8) & 0x00ffff00) + ((usADS_data[5]
>> 24) & 0x000000ff);

		//T2
		tmp_data[7] = (usADS_data[5]  & 0x00ffffff);

		if ((tmp_data[0] || tmp_data[1] || tmp_data[2] || tmp_data[3] ||
tmp_data[4] || tmp_data[5] || tmp_data[6] || tmp_data[7]) == 0)
		{
			// No data
			TOGGLE_LED3();
			return;
		}

		for (i = 0; i < MAX_CHANNELS; i++)
		{
			if (tmp_data[i] > 0x007fffff)
			{
				tmp_data[i] |= 0xff000000;

	// The 24 bit word is negative so make the integer (32bit word) negative.
```

```
                        mag_data[i] = ((((double)((double) tmp_data[i] /
(double) Cmin) * Vmin) * 1000000)/S) - B0[i];  // Convert ADC code into nT
                    }
                    else
                    {
                        mag_data[i] = ((((double)((double) tmp_data[i] /
(double) Cmax) * Vmax) * 1000000)/S) - B0[i];  // Convert ADC code into nT
                    }

                    raw_data[i] = tmp_data[i];

                    mean_raw_data[i] += tmp_data[i];
                    mean_mag_data[i] += mag_data[i];

                    if (count > 1)
                    {
                        mean_raw_data[i] /= 2;
                        mean_mag_data[i] /= 2;
                    }
                }
                count++;
                fnWrite(qhMySSCHandle, (unsigned char *)DummyMsg, 1); //transmit dummy
message to ADS - this will trigger data from the ADS
        }
}

QUEUE_HANDLE fnOpenMyUART(void)
{
        QUEUE_HANDLE qhMyUARTHandle; // UART handle to be obtained during open

        TTYTABLE tInterfaceParameters; // table for passing information to driver

        tInterfaceParameters.Channel = 0; // set UART channel for serial use i.e.
COM1
        tInterfaceParameters.ucSpeed = SERIAL_BAUD_9600; // baud rate 9'600
        tInterfaceParameters.Rx_tx_sizes.RxQueueSize = 256; // input buffer size
        tInterfaceParameters.Rx_tx_sizes.TxQueueSize = 512; // output buffer size
        tInterfaceParameters.Task_to_wake = TASK_MY_UART_TASK; // wake task on rx

#ifdef SUPPORT_FLOW_HIGH_LOW
        tInterfaceParameters.ucFlowHighWater = 80; // set the flow control high in %
        tInterfaceParameters.ucFlowLowWater = 20; // set the flow control low in %
#endif
        tInterfaceParameters.Config = (CHAR_8 + NO_PARITY + ONE_STOP + USE_XON_OFF +
CHAR_MODE);

#ifdef SERIAL_SUPPORT_DMA
        tInterfaceParameters.ucDMAConfig = UART_TX_DMA; // activate DMA on
transmission
#endif

        if ((qhMyUARTHandle = fnOpen( TYPE_TTY, FOR_I_O, &tInterfaceParameters )) !=
0) {
                // open the channel with defined configurations (initially inactive)
                fnDriver(qhMyUARTHandle, ( TX_ON | RX_ON ), 0 ); // enable rx and tx
        }
        return qhMyUARTHandle; // return the serial port handle for this UART
}

QUEUE_HANDLE fnOpenMySSC(void)
{
        QUEUE_HANDLE qhMySSCHandle; // SSC handle to be obtained during open

        SSCTABLE tInterfaceParameters; // table for passing information to driver

        tInterfaceParameters.Channel = OUR_SSC_CHANNEL; // set I2S channel for use
        tInterfaceParameters.usSpeed = ((MASTER_CLOCK/2)/(1700000));// data rate
6.0MHz (should be 5.4MHz but seems to work)
```

```
	tInterfaceParameters.Task_to_wake = TASK_MY_SSC_TASK;// wake self when
frames received
	tInterfaceParameters.ucTxDataShift = 193;
	// bit shift from sync to start of transmit data
	tInterfaceParameters.ucRxDataShift = 1;
	// bit shift from sync to start of receive data
	tInterfaceParameters.usConfig = (TX_MSB_FIRST | TX_POSITIVE_FRAME_PULSE |
RX_MSB_FIRST | RX_SYNC_RISING);
	tInterfaceParameters.ucWordSize = 32;
	// transmit and receive data treated as words of this width
	tInterfaceParameters.ucFrameLength = 6; // the number of words in a frame
	tInterfaceParameters.Rx_tx_sizes.RxQueueSize = 256;
	// input buffer size (in words)
	tInterfaceParameters.Rx_tx_sizes.TxQueueSize = 128;
	// output buffer size (in words)

#ifdef SSC_SUPPORT_DMA
	tInterfaceParameters.ucDMAConfig = (UART_TX_DMA | UART_RX_DMA);
	// activate DMA on transmission and reception
#endif

	if ((qhMySSCHandle = fnOpen(TYPE_SSC, FOR_I_O, &tInterfaceParameters)) != 0)
	{
		// open the channel with defined parameters
		fnDriver(qhMySSCHandle, (TX_ON | RX_ON), 0); // enable rx and tx
	}

	return qhMySSCHandle;
}

extern void fnMyInitialiseTask(TTASKTABLE *ptrTaskTable)
{
	_CONFIG_PORT_OUTPUT(B, BLINK_LED2);
	_CONFIG_PORT_OUTPUT(B, BLINK_LED3);
	_CONFIG_PORT_OUTPUT(B, BLINK_LED4);

	qhMySSCHandle = fnOpenMySSC();
	qhMyUARTHandle = fnOpenMyUART();
}

extern void fnMyFirstTask1(TTASKTABLE *ptrTaskTable)
{
	// Replace with GPS UART recieve message
	//fnWrite(qhMySSCHandle, (unsigned char *)DummyMsg, 1); //transmit dummy
message to ADS - this will trigger data from the ADS
}
```

Appendix G – MATLAB Source Code – Noise Analysis

The following code extract was used to post-process the data which performs filtering and plotting [76] of pre- and post-processed data in the time and frequency domain. This was done to isolate the AC signals and attribute them to the various DCBL motor speeds or other spurious anomalies.

The code was run using MATLAB R2009a [77]. The work of Morrison *et al.* [78] was used as a guide for the noise analysis.

```
function pe=process_edit()

    global MaxCode MinCode MaxV MinV

    [FileName,PathName] = uigetfile('*.txt',...
        'Select the .txt data file to process');
    fid=fopen(strcat(PathName, FileName));
    InputText = textscan(fid, '%s %s %s %s %s %s %s %s %s',...
        'delimiter', '\t');
    fclose(fid);

    % Setup user input
    prompt = {'Enter Fc/[Hz]:',...                % Cut-off frequency
              'Enter m [0 .. 1]:',...             % Amount of 'chopped' data
              'Enter BW [0.01 .. 1]/[Hz]:',...    % Bandwidth of notch filter
              'Display Sensor 1/[Y/N]:',...
              'Display Sensor 2/[Y/N]:',...
              'Display Noise vs Time/[Y/N]:',...
              'Display Spectral Analysis/[Y/N]:',...
              'Export Figure/[Y/N]:'};

    dlg_title = 'User Input';
    num_lines = 1;
    % Defualt user values
    def = {'15', '0.10', '0.01', 'Y', 'Y', 'Y', 'Y', 'N'};
    answer = inputdlg(prompt, dlg_title ,num_lines, def);

    % Get cut-off frequency /[Hz]
    f3dB = str2double(answer(1));

    % Get 'chopped' percentage
    m = str2double(answer(2));
    %Check 0 < m < 1
    if (m < 0)
        m = 0;
    end;

    if (m > 1)
        m = 1;
    end;

    % Get notch filter bandwidth /[Hz]
    BW = str2double(answer(3));
    %Check 0.01 < BW < 1
    if (BW < 0.01)
        BW = 0.01;
    end;

    if (BW > 1)
        BW = 1;
    end;

    if (strcmp(answer(4), 'Y'))
        DisplaySensor1 = 1;
    else
        DisplaySensor1 = 0;
    end;
```

```
if (strcmp(answer(5), 'Y'))
    DisplaySensor2 = 1;
else
    DisplaySensor2 = 0;
end;

if (strcmp(answer(6), 'Y'))
    DisplayNoise = 1;
else
    DisplayNoise = 0;
end;

if (strcmp(answer(7), 'Y'))
    DisplaySpectra = 1;
else
    DisplaySpectra = 0;
end;

if (strcmp(answer(8), 'Y'))
    ExportFigure = 1;
else
    ExportFigure = 0;
end;

col1=InputText{1};
col2=InputText{2};
col3=InputText{3};
col4=InputText{4};
col5=InputText{5};
col6=InputText{6};
col7=InputText{7};
col8=InputText{8};
col9=InputText{9};

count = zeros(1, 1);

% Process the .txt file into the data to be processed
for i=1:length(col1);
    count = count + 1;

    if (strcmp(col1(i), '[Channel Data]'))
        break;
    elseif (strcmp(col1(i), 'Max Voltage'))
        MaxV=str2double(col2(i));       %V
    elseif (strcmp(col1(i), 'Min Voltage'))
        MinV=str2double(col2(i));       %V
    elseif (strcmp(col1(i), 'Max Code'))
        MaxCode=str2double(col2(i));    %2^23 -1 i.e. +60000nT
    elseif (strcmp(col1(i), 'Min Code'))
        MinCode=str2double(col2(i));    %-2^23   i.e. -60000nT
    elseif (strcmp(col1(i), 'Sampling Frequency'))
        SampF=str2double(col2(i));      %sps
    elseif (strcmp(col1(i),'Notes'))
        Note=char(col2(i));
    elseif (strcmp(col1(i), 'Date and Time'))
        DateandTime=char(col2(i));
    elseif (strcmp(col1(i), 'EVM Device Name'))
        EVMDeviceName=char(col2(i));
    end;
end;
```

```
% Setup inital offset values and constants
B0x1=-123;    %nT
B0y1=-54;     %nT
B0z1=-136;    %nT
B0x2=4;       %nT
B0y2=-93;     %nT
B0z2=-148;    %nT
S=36.5;       %uV/nT

numsamples=length(col2) - count;

%Pre-allocate arrays
x1 = zeros(1, numsamples);
y1 = zeros(1, numsamples);
z1 = zeros(1, numsamples);
t1 = zeros(1, numsamples);
x2 = zeros(1, numsamples);
y2 = zeros(1, numsamples);
z2 = zeros(1, numsamples);
t2 = zeros(1, numsamples);

Vx1 = zeros(1, numsamples);
Vy1 = zeros(1, numsamples);
Vz1 = zeros(1, numsamples);
Vx2 = zeros(1, numsamples);
Vy2 = zeros(1, numsamples);
Vz2 = zeros(1, numsamples);

Bx1 = zeros(1, numsamples);
By1 = zeros(1, numsamples);
Bz1 = zeros(1, numsamples);
Bx2 = zeros(1, numsamples);
By2 = zeros(1, numsamples);
Bz2 = zeros(1, numsamples);

noiseBx1 = zeros(1, numsamples);
noiseBy1 = zeros(1, numsamples);
noiseBz1 = zeros(1, numsamples);
noiseBx2 = zeros(1, numsamples);
noiseBy2 = zeros(1, numsamples);
noiseBz2 = zeros(1, numsamples);

newBx1 = zeros(1, numsamples);
newBy1 = zeros(1, numsamples);
newBz1 = zeros(1, numsamples);

newBx2 = zeros(1, numsamples);
newBy2 = zeros(1, numsamples);
newBz2 = zeros(1, numsamples);

% Calculate the shift based on the user input
shiftval = numsamples*m;
```

```
% 'Chop' the 'ringing' data based on the shift
filterednoiseBx1 = zeros(1, numsamples - shiftval);
filterednoiseBy1 = zeros(1, numsamples - shiftval);
filterednoiseBz1 = zeros(1, numsamples - shiftval);
filterednoiseBx2 = zeros(1, numsamples - shiftval);
filterednoiseBy2 = zeros(1, numsamples - shiftval);
filterednoiseBz2 = zeros(1, numsamples - shiftval);

t = zeros(1, numsamples);
sample = zeros(1, numsamples);

% Convert .txt data into doubles
for i=count+1:numsamples+count
    x1(i-count)=str2double(col2(i));
    y1(i-count)=str2double(col3(i));
    z1(i-count)=str2double(col4(i));
    t1(i-count)=str2double(col5(i));

    x2(i-count)=str2double(col6(i));
    y2(i-count)=str2double(col7(i));
    z2(i-count)=str2double(col8(i));
    t2(i-count)=str2double(col9(i));
end;

for i=1:numsamples
    % Calculate the ADC voltage output
    Vx1(i)=evaluateVolts(x1(i));
    Vy1(i)=evaluateVolts(y1(i));
    Vz1(i)=evaluateVolts(z1(i));

    Vx2(i)=evaluateVolts(x2(i));
    Vy2(i)=evaluateVolts(y2(i));
    Vz2(i)=evaluateVolts(z2(i));

    % Calculate the nT value based on the voltage
    Bx1(i)=evaluateNanoTesla(Vx1(i), S, B0x1);
    By1(i)=evaluateNanoTesla(Vy1(i), S, B0y1);
    Bz1(i)=evaluateNanoTesla(Vz1(i), S, B0z1);

    Bx2(i)=evaluateNanoTesla(Vx2(i), S, B0x2);
    By2(i)=evaluateNanoTesla(Vy2(i), S, B0y2);
    Bz2(i)=evaluateNanoTesla(Vz2(i), S, B0z2);

    % Calcualte the time and sample value
    t(i)=(i-1)/SampF;
    sample(i)=i;
end;

%Compute the noise but subtracting the mean from the calculated value
for i=1:numsamples
    noiseBx1(i)=Bx1(i)-mean(Bx1);
    noiseBy1(i)=By1(i)-mean(By1);
    noiseBz1(i)=Bz1(i)-mean(Bz1);
    noiseBx2(i)=Bx2(i)-mean(Bx2);
    noiseBy2(i)=By2(i)-mean(By2);
    noiseBz2(i)=Bz2(i)-mean(Bz2);
end;
```

```
% Calculate the TMI
TMI1 = sqrt((mean(Bx1))^2 + (mean(By1))^2 + (mean(Bz1))^2);
TMI2 = sqrt((mean(Bx2))^2 + (mean(By2))^2 + (mean(Bz2))^2);

% fft computation
% number of fft points as a power of 2 (makes the processing more
% efficient)
fft_points = 2^nextpow2(numsamples);

% Only show half the plot
plot_div = 2;

% Note on the Power calculation:
% ------------------------------
% Power has several definitions depending on what one is trying to do.
% The one I've used (division by N^2) is the most self-consistent with
% Perceval's theorem, which is a conservation law regarding the power
% and/or energy in the time and frequency domains.

% Taking twice the absolute value gives a one-sided spectrum with
% negative frequencies folded onto positive axis.

% Dividing by the length of signal normalizes the amplitude to the
% original input signal.

FBx1 = 2*abs(fft(noiseBx1, fft_points)/length(noiseBx1));
PBx1 = FBx1.^2/numsamples^2;

FBy1 = 2*abs(fft(noiseBy1, fft_points)/length(noiseBy1));
PBy1 = FBy1.^2/numsamples^2;

FBz1 = 2*abs(fft(noiseBz1, fft_points)/length(noiseBz1));
PBz1 = FBz1.^2/numsamples^2;

FBx2 = 2*abs(fft(noiseBx2, fft_points)/length(noiseBx2));
PBx2 = FBx2.^2/numsamples^2;

FBy2 = 2*abs(fft(noiseBy2, fft_points)/length(noiseBy2));
PBy2 = FBy2.^2/numsamples^2;

FBz2 = 2*abs(fft(noiseBz2, fft_points)/length(noiseBz2));
PBz2 = FBz2.^2/numsamples^2;

% Calculate the frequency
f=SampF*(0:fft_points/2)/fft_points;

%Find the peaks in the spectra
start_f_idx = find(f > f3dB, 1);

% Set the minimum peak threshold for each data set
minpeakheightPBx1 = mean(PBx1(1:start_f_idx));
minpeakheightPBy1 = mean(PBy1(1:start_f_idx));
minpeakheightPBz1 = mean(PBz1(1:start_f_idx));

minpeakheightPBx2 = mean(PBx2(1:start_f_idx));
minpeakheightPBy2 = mean(PBy2(1:start_f_idx));
minpeakheightPBz2 = mean(PBz2(1:start_f_idx));
```

```
% Find the peaks of each data set above the threshold calculated above
[peakmagX1, peakidxX1] = findpeaks(PBx1(start_f_idx:fft_points...
    / plot_div), 'minpeakheight', minpeakheightPBx1);
peakidxX1 = peakidxX1 + start_f_idx - 1;
fmaxX1 = f(peakidxX1);

[peakmagY1, peakidxY1] = findpeaks(PBy1(start_f_idx:fft_points...
    / plot_div), 'minpeakheight', minpeakheightPBy1);
peakidxY1 = peakidxY1 + start_f_idx - 1;
fmaxY1 = f(peakidxY1);

[peakmagZ1, peakidxZ1] = findpeaks(PBz1(start_f_idx:fft_points...
    / plot_div), 'minpeakheight', minpeakheightPBz1);
peakidxZ1 = peakidxZ1 + start_f_idx - 1;
fmaxZ1 = f(peakidxZ1);

[peakmagX2, peakidxX2] = findpeaks(PBx2(start_f_idx:fft_points...
    / plot_div), 'minpeakheight', minpeakheightPBx2);
peakidxX2 = peakidxX2 + start_f_idx - 1;
fmaxX2 = f(peakidxX2);

[peakmagY2, peakidxY2] = findpeaks(PBy2(start_f_idx:fft_points...
    / plot_div), 'minpeakheight', minpeakheightPBy2);
peakidxY2 = peakidxY2 + start_f_idx - 1;
fmaxY2 = f(peakidxY2);

[peakmagZ2, peakidxZ2] = findpeaks(PBz2(start_f_idx:fft_points...
    / plot_div), 'minpeakheight', minpeakheightPBz2);
peakidxZ2 = peakidxZ2 + start_f_idx - 1;
fmaxZ2 = f(peakidxZ2);

% Convert to Power/Frequency in dB/Hz to logarithmic scale
PBx1 = 10*log10(PBx1);     % Power/Frequency /[dB/Hz]
PBy1 = 10*log10(PBy1);     % Power/Frequency /[dB/Hz]
PBz1 = 10*log10(PBz1);     % Power/Frequency /[dB/Hz]

PBx2 = 10*log10(PBx2);     % Power/Frequency /[dB/Hz]
PBy2 = 10*log10(PBy2);     % Power/Frequency /[dB/Hz]
PBz2 = 10*log10(PBz2);     % Power/Frequency /[dB/Hz]

% Preserve the B values
newBx1 = Bx1;
newBy1 = By1;
newBz1 = Bz1;

newBx2 = Bx2;
newBy2 = By2;
newBz2 = Bz2;

% Calculate the unique frequencies to be filtered
frequencypeaks = unique(cat(2, fmaxX1, fmaxY1, fmaxZ1,...
    fmaxX2, fmaxY2, fmaxZ2));

% Calculate the unique peak index
idxpeaks = unique(cat(2, peakidxX1, peakidxY1, peakidxZ1,...
    peakidxX2, peakidxY2, peakidxZ2));
```

```
for i=1:length(frequencypeaks)
    % Recursively apply filters although this might cause 'ringing'
    % artefacts especially with the higher sampling frequencies
    % (e.g. 52kSPS and 100kSPS) as the excitations frequencies and
    % harmonics will be 'visible'.
    Wo = frequencypeaks(i)/(SampF/2);
    [b,a] = iirnotch(Wo, BW);

    newBx1 = filter(b, a, newBx1);
    newBy1 = filter(b, a, newBy1);
    newBz1 = filter(b, a, newBz1);
    newBx2 = filter(b, a, newBx2);
    newBy2 = filter(b, a, newBy2);
    newBz2 = filter(b, a, newBz2);
end;

% 'Shift'/'chop' the rining data off the front
filteredBx1 = newBx1(shiftval + 1:numsamples);
filteredBy1 = newBy1(shiftval + 1:numsamples);
filteredBz1 = newBz1(shiftval + 1:numsamples);

filteredBx2 = newBx2(shiftval + 1:numsamples);
filteredBy2 = newBy2(shiftval + 1:numsamples);
filteredBz2 = newBz2(shiftval + 1:numsamples);

%Compute the noise
for i=1:(numsamples - shiftval)
    filterednoiseBx1(i)=filteredBx1(i) - mean(filteredBx1);
    filterednoiseBy1(i)=filteredBy1(i) - mean(filteredBy1);
    filterednoiseBz1(i)=filteredBz1(i) - mean(filteredBz1);
    filterednoiseBx2(i)=filteredBx2(i) - mean(filteredBx2);
    filterednoiseBy2(i)=filteredBy2(i) - mean(filteredBy2);
    filterednoiseBz2(i)=filteredBz2(i) - mean(filteredBz2);
end;

% Compute the TMI of the filtered data
filteredTMI1 = sqrt((mean(filteredBx1))^2 +...
    (mean(filteredBy1))^2 +...
    (mean(filteredBz1))^2);
filteredTMI2 = sqrt((mean(filteredBx2))^2 +...
    (mean(filteredBy2))^2 +...
    (mean(filteredBz2))^2);
```

```
% Calculate filtered spectra
FFBx1=2*abs(fft(filterednoiseBx1, fft_points)...
    / length(filterednoiseBx1));
FPBx1=10*log10(FFBx1.^2/numsamples^2);  % Power/Frequency /[dB/Hz]

FFBy1=2*abs(fft(filterednoiseBy1, fft_points)...
    / length(filterednoiseBy1));
FPBy1=10*log10(FFBy1.^2/numsamples^2);  % Power/Frequency /[dB/Hz]

FFBz1=2*abs(fft(filterednoiseBz1, fft_points)...
    / length(filterednoiseBz1));
FPBz1=10*log10(FFBz1.^2/numsamples^2);  % Power/Frequency /[dB/Hz]

FFBx2=2*abs(fft(filterednoiseBx2, fft_points)...
    / length(filterednoiseBx2));
FPBx2=10*log10(FFBx2.^2/numsamples^2);  % Power/Frequency /[dB/Hz]

FFBy2=2*abs(fft(filterednoiseBy2, fft_points)...
    / length(filterednoiseBy2));
FPBy2=10*log10(FFBy2.^2/numsamples^2);  % Power/Frequency /[dB/Hz]

FFBz2=2*abs(fft(filterednoiseBz2, fft_points)...
    / length(filterednoiseBz2));
FPBz2=10*log10(FFBz2.^2/numsamples^2);  % Power/Frequency /[dB/Hz]

% Calculate the scales for x and y axes of the various plots to make
% them the 'same' for better viewing.
maxylim = max([abs(noiseBx1), abs(noiseBy1), abs(noiseBz1),...
    abs(noiseBx2), abs(noiseBy2), abs(noiseBz2)]);
maxfliteredylim = max([abs(filterednoiseBx1),...
    abs(filterednoiseBy1), abs(filterednoiseBy1),...
    abs(filterednoiseBx2), abs(filterednoiseBy2),...
    abs(filterednoiseBz2)]);

maxFFTylim = roundn(max([PBx1(2:end), PBy1(2:end), PBz1(2:end),...
    PBx2(2:end), PBy2(2:end), PBz2(2:end),...
    FPBx1(2:end), FPBy1(2:end), FPBz1(2:end),...
    FPBx2(2:end), FPBy2(2:end), FPBz2(2:end)]) / 1.2, 1);
minFFTylim = roundn(min([PBx1(2:end), PBy1(2:end), PBz1(2:end),...
    PBx2(2:end), PBy2(2:end), PBz2(2:end),...
    FPBx1(2:end), FPBy1(2:end), FPBz1(2:end),...
    FPBx2(2:end), FPBy2(2:end), FPBz2(2:end)]) * 1.2, 1);

maxxlim = t(length(t))* 1.1;
maxfliteredxlim = t(length(t))* 1.1;
```

```
if (DisplayNoise == 1)
    if (DisplaySensor1 == 1)
        % Plot Sensor 1 noise data
        figure('units', 'normalized', 'outerposition', [0 0 1 1]),...
            plot(t, noiseBx1, t, noiseBy1, t, noiseBz1);
        title({strcat(PathName, FileName);...
            'Noise: Bx1, By1, Bz1';...
            EVMDeviceName;Note;DateandTime;...
            strcat(num2str(SampF/1000), 'kSPS')});
        ylabel('B /[nT]');
        xlabel('time /[s]');
        ylim([(maxylim * -1.2) (maxylim * 1.2)]);
        xlim([0 maxxlim]);
        text(t(round(length(t) * 0.1)), maxylim * 1.02,...
            strcat('TMI1 = ',...
            num2str(TMI1, ' %6.2f'), ' nT'),...
            'VerticalAlignment', 'bottom');
        grid on;
        legend(strcat('Bx1: std.= ',...
            num2str(std(Bx1), ' %5.2f'), ' nT; mean = ',...
            num2str(mean(Bx1), ' %5.2f'), ' nT'),...
            strcat('By1: std.= ',...
            num2str(std(By1), ' %5.2f'), ' nT; mean = ',...
            num2str(mean(By1), ' %5.2f'), ' nT'),...
            strcat('Bz1: std.= ',...
            num2str(std(Bz1), ' %5.2f'), ' nT; mean = ',...
            num2str(mean(Bz1), ' %5.2f'), ' nT'));
        set(gcf, 'Color', 'white');                 % white bckgr
        if (ExportFigure == 1)
            export_fig( gcf,...                     % figure handle
                strcat(PathName,...
                FileName(1:end-4),...
                ' - Sensor 1 Noise'),...            % filename w/o ext.
                '-painters', ...                    % renderer
                '-jpg', ...                         % file format
                '-r300' );                          % resolution in dpi
        end;

        % Plot Sensor 1 filtered noise data
        figure('units', 'normalized', 'outerposition', [0 0 1 1]),...
            plot(t(shiftval + 1:numsamples), filterednoiseBx1,...
                    t(shiftval + 1:numsamples), filterednoiseBy1,...
                    t(shiftval + 1:numsamples), filterednoiseBz1);
        title({strcat(PathName, FileName);...
            'Filtered Noise: Bx1, By1, Bz1';...
            EVMDeviceName;Note;DateandTime;...
            strcat(num2str(SampF/1000), 'kSPS')});
        ylabel('B /[nT]');
        xlabel('time /[s]');
        ylim([(maxfliteredylim * -1.2) (maxfliteredylim * 1.2)]);
        xlim([0 maxfliteredxlim]);
        text(t(round(length(t) * 0.1)), maxfliteredylim * 1.02,...
            strcat('TMI1 = ',...
            num2str(filteredTMI1, ' %6.2f'), ' nT'),...
            'VerticalAlignment', 'bottom');
        grid on;
        legend(strcat('Bx1: std.= ',...
            num2str(std(filteredBx1), ' %5.2f'), ' nT; mean = ',...
            num2str(mean(filteredBx1), ' %5.2f'), ' nT'),...
            strcat('By1: std.= ',...
            num2str(std(filteredBy1), ' %5.2f'), ' nT; mean = ',...
```

```
            num2str(mean(filteredBy1), ' %5.2f'), ' nT'),...
            strcat('Bz1: std.= ',...
            num2str(std(filteredBz1), ' %5.2f'), ' nT; mean = ',...
            num2str(mean(filteredBz1), ' %5.2f'), ' nT'));
        set(gcf, 'Color', 'white');                 % white bckgr
        if (ExportFigure == 1)
            export_fig( gcf,...                     % figure handle
                strcat(PathName,...
                FileName(1:end-4),...
                ' - Sensor 1 Filtered Noise'),...   % filename w/o ext.
                '-painters', ...                    % renderer
                '-jpg', ...                         % file format
                '-r300' );                          % resolution in dpi
        end;
    end;

    if (DisplaySensor2 == 1)
        % Plot Sensor 2 noise data
        figure('units', 'normalized', 'outerposition', [0 0 1 1]),...
            plot(t, noiseBx2, t, noiseBy2, t, noiseBz2);
        title({strcat(PathName, FileName);...
            'Noise: Bx2, By2, Bz2';...
            EVMDeviceName;Note;DateandTime;...
            strcat(num2str(SampF/1000), 'kSPS')});
        ylabel('B /[nT]');
        xlabel('time /[s]');
        ylim([(maxylim * -1.2) (maxylim * 1.2)]);
        xlim([0 maxxlim]);
        text(t(round(length(t) * 0.1)),...
            maxylim * 1.02,...
            strcat('TMI2 = ', num2str(TMI2, ' %6.2f'), ' nT'),...
            'VerticalAlignment', 'bottom');
        grid on;
        legend(strcat('Bx2: std. = ',...
            num2str(std(Bx2), ' %5.2f'), ' nT; mean = ',...
            num2str(mean(Bx2), ' %5.2f'), ' nT'),...
            strcat('By2: std. = ',...
            num2str(std(By2), ' %5.2f'), ' nT; mean = ',...
            num2str(mean(By2), ' %5.2f'), ' nT'),...
            strcat('Bz2: std. = ',...
            num2str(std(Bz2), ' %5.2f'), ' nT; mean = ',...
            num2str(mean(Bz2), ' %5.2f'), ' nT'));
        set(gcf, 'Color', 'white');                 % white bckgr
        if (ExportFigure == 1)
            export_fig( gcf, ...                    % figure handle
                strcat(PathName,...
                FileName(1:end-4),...
                ' - Sensor 2 Noise'),...            % filename w/o ext.
                '-painters', ...                    % renderer
                '-jpg', ...                         % file format
                '-r300' );                          % resolution in dpi
        end;
```

```
            % Plot Sensor 2 filtered noise data
            figure('units', 'normalized', 'outerposition', [0 0 1 1]),...
                plot(t(shiftval + 1:numsamples), filterednoiseBx2,...
                        t(shiftval + 1:numsamples), filterednoiseBy2,...
                        t(shiftval + 1:numsamples), filterednoiseBz2);
            title({strcat(PathName, FileName);...
                'Filtered Noise: Bx2, By2, Bz2';...
                EVMDeviceName;Note;DateandTime;...
                strcat(num2str(SampF/1000), 'kSPS')});
            ylabel('B /[nT]');
            xlabel('time /[s]');
            ylim([(maxfliteredylim * -1.2) (maxfliteredylim * 1.2)]);
            xlim([0 maxfliteredxlim]);
            text(t(round(length(t) * 0.1)),...
                maxfliteredylim * 1.02, strcat('TMI2 = ',...
                num2str(filteredTMI2, '%6.2f'), ' nT'),...
                'VerticalAlignment', 'bottom');
            grid on;
            legend(strcat('Bx2: std. = ',...
                num2str(std(filteredBx2), ' %5.2f'), ' nT; mean = ',...
                num2str(mean(filteredBx2), ' %5.2f'), ' nT'),...
                strcat('By2: std. = ',...
                num2str(std(filteredBy2), ' %5.2f'), ' nT; mean = ',...
                num2str(mean(filteredBy2), ' %5.2f'), ' nT'),...
                strcat('Bz2: std. = ',...
                num2str(std(filteredBz2), ' %5.2f'), ' nT; mean = ',...
                num2str(mean(filteredBz2), ' %5.2f'), ' nT'));
            set(gcf, 'Color', 'white');                 % white bckgr
            if (ExportFigure == 1)
                export_fig( gcf, ...                    % figure handle
                    strcat(PathName,...
                    FileName(1:end-4),...
                    ' - Sensor 2 Filtered Noise'),...   % filename w/o ext.
                    '-painters', ...                    % renderer
                    '-jpg', ...                         % file format
                    '-r300' );                          % resolution in dpi
            end;
        end;
    end;

    if (DisplaySpectra == 1)
        if (DisplaySensor1 == 1)
            % Plot sensor 1 noise spectra
            for i=1:length(idxpeaks)
                maxpeaks(i) = max([PBx1(idxpeaks(i)),...
                    PBy1(idxpeaks(i)),...
                    PBz1(idxpeaks(i))]);
            end;

            figure('units', 'normalized', 'outerposition', [0 0 1 1]),...
                semilogx(f(1:fft_points / plot_div),...
                PBx1(1:fft_points / plot_div),...
                     f(1:fft_points / plot_div),...
                     PBy1(1:fft_points / plot_div),...
                     f(1:fft_points / plot_div),...
                     PBz1(1:fft_points / plot_div));

            title({strcat(PathName,FileName);...
                'Periodogram using FFT of: Bx1, By1, Bz1';...
                EVMDeviceName;Note;DateandTime;...
                strcat(num2str(SampF/1000), 'kSPS')});
```

```
        ylabel('Power/Frequency /[dB/Hz]');
        xlabel('Frequency /[Hz]');
        ylim([minFFTylim maxFFTylim]);
        grid on;
        legend('PBx1', 'PBy1', 'PBz1');
        strValues = strtrim(cellstr(num2str([frequencypeaks(:)],...
            '%5.2f Hz')));
        text(frequencypeaks, maxpeaks, strValues,...
            'VerticalAlignment', 'bottom');
        set(gcf, 'Color', 'white');                 % white bckgr
        if (ExportFigure == 1)
        export_fig( gcf, ...                        % figure handle
            strcat(PathName, FileName(1:end-4),...
            ' - Sensor 1 Noise Spectra'),...        % filename w/o ext.
            '-painters', ...                        % renderer
            '-jpg', ...                             % file format
            '-r300' );                              % resolution in dpi
        end;

        % Plot sensor 1 filtered noise spectra
        figure('units', 'normalized', 'outerposition', [0 0 1 1]),...
            semilogx(f(1:fft_points / plot_div),...
            FPBx1(1:fft_points / plot_div),...
                 f(1:fft_points / plot_div),...
                 FPBy1(1:fft_points / plot_div),...
                 f(1:fft_points / plot_div),...
                 FPBz1(1:fft_points / plot_div));

        title({strcat(PathName, FileName);...
            'Periodogram using FFT of filtered: Bx1, By1, Bz1';...
            EVMDeviceName;Note;DateandTime;...
            strcat(num2str(SampF/1000), 'kSPS')});
        ylabel('Power/Frequency /[dB/Hz]');
        xlabel('Frequency /[Hz]');
        ylim([minFFTylim maxFFTylim]);
        grid on;
        legend('PBx1', 'PBy1', 'PBz1');
        set(gcf, 'Color', 'white');                 % white bckgr
        if (ExportFigure == 1)
            export_fig( gcf, ...                    % figure handle
                strcat(PathName,...
                FileName(1:end-4),...
                ' - Sensor 1 Filt. Spectra'),...    % filename w/o ext.
                '-painters', ...                    % renderer
                '-jpg', ...                         % file format
                '-r300' );                          % resolution in dpi
        end;

    end;
```

```
if (DisplaySensor2 == 1)
    % Plot sensor 2 noise spectra
    for i=1:length(idxpeaks)
        maxpeaks(i) = max([PBx2(idxpeaks(i)),...
            PBy2(idxpeaks(i)),...
            PBz2(idxpeaks(i))]);
    end;

    figure('units', 'normalized', 'outerposition', [0 0 1 1]),...
        semilogx(f(1:fft_points / plot_div),...
        PBx2(1:fft_points / plot_div),...
             f(1:fft_points / plot_div),...
             PBy2(1:fft_points / plot_div),...
             f(1:fft_points / plot_div),...
             PBz2(1:fft_points / plot_div));

    title({strcat(PathName, FileName);...
        'Periodogram using FFT of: Bx2, By2, Bz2';...
        EVMDeviceName;Note;DateandTime;...
        strcat(num2str(SampF/1000), 'kSPS')});
    ylabel('Power/Frequency /[dB/Hz]');
    xlabel('Frequency /[Hz]');
    ylim([minFFTylim maxFFTylim]);
    grid on;
    legend('PBx2', 'PBy2', 'PBz2');
    strValues = strtrim(cellstr(num2str([frequencypeaks(:)],...
        '%5.2f Hz')));
    text(frequencypeaks, maxpeaks, strValues,...
        'VerticalAlignment', 'bottom');
    set(gcf, 'Color', 'white');                 % white bckgr
    if (ExportFigure == 1)
        export_fig( gcf,...                     % figure handle
            strcat(PathName,...
            FileName(1:end-4),...
            ' - Sensor 2 Noise Spectra'),...    % filename w/o ext.
            '-painters', ...                    % renderer
            '-jpg', ...                         % file format
            '-r300' );                          % resolution in dpi
    end;
```

```
            % Plot sensor 2 filtered noise spectra
            figure('units', 'normalized', 'outerposition', [0 0 1 1]),...
                semilogx(f(1:fft_points / plot_div),...
                FPBx2(1:fft_points / plot_div),...
                     f(1:fft_points / plot_div),...
                     FPBy2(1:fft_points / plot_div),...
                     f(1:fft_points / plot_div),...
                     FPBz2(1:fft_points / plot_div));

            title({strcat(PathName, FileName);...
                'Periodogram using FFT of filtered: Bx2, By2, Bz2';...
                EVMDeviceName;Note;DateandTime;...
                strcat(num2str(SampF/1000), 'kSPS')});
            ylabel('Power/Frequency /[dB/Hz]');
            xlabel('Frequency /[Hz]');
            ylim([minFFTylim maxFFTylim]);
            grid on;
            legend('PBx2', 'PBy2', 'PBz2');
            set(gcf, 'Color', 'white');                 % white bckgr
            if (ExportFigure == 1)
                export_fig( gcf,...                     % figure handle
                    strcat(PathName,...
                    FileName(1:end-4),...
                    ' - Sensor 2 Filt. Spectra'),...    % filename w/o ext.
                    '-painters', ...                    % renderer
                    '-jpg', ...                         % file format
                    '-r300' );                          % resolution in dpi
            end;
        end;
    end;

function voltage = evaluateVolts(code)
    % Calculate the ADC voltage from the code
    global MaxCode MinCode MaxV MinV
    if(code >= 0)
        voltage=code/MaxCode*MaxV;
    elseif(code < 0)
        voltage=code/MinCode*MinV;
    end;

function nanotesla = evaluateNanoTesla(voltage, S, B0)
    % Calculate the nT from the ADC voltage
    nanotesla=voltage*10^6/S-B0;
```

Appendix H – MATLAB Source Code – Grid Data Analysis

The following code extract was used to process data and do the grid data analysis. The code was run using MATLAB R2009a [77], [72].

```
function pgd=process_grid_data()

    global X1 Y1 Z1 X2 Y2 Z2

    prompt = {'How many rows?:', 'How many columns?:', 'How many samples to
process?:'};
    dlg_title = 'User Input';
    num_lines = 1;
    def = {'8', '15', '2048'};
    answer = inputdlg(prompt, dlg_title ,num_lines, def);

    num_rows = str2double(answer(1));
    num_columns = str2double(answer(2));
    num_samples_to_process = str2double(answer(3));

    X1 = zeros(num_rows, num_columns);
    Y1 = zeros(num_rows, num_columns);
    Z1 = zeros(num_rows, num_columns);

    X2 = zeros(num_rows, num_columns);
    Y2 = zeros(num_rows, num_columns);
    Z2 = zeros(num_rows, num_columns);

    return_matrix = zeros(1,6);

    [FileNames,PathName] = uigetfile('*.txt','Select the .txt data file(s)
to process','MultiSelect', 'on');

    if(iscellstr(FileNames)==1)
        for filenum=1:length(FileNames)
            FileName=FileNames{length(FileNames)-filenum+1};
            return_matrix =
processfile(PathName,FileName,num_samples_to_process);
            getdata(return_matrix, FileName, num_columns);
        end;
    else
        FileName=FileNames;
        return_matrix =
processfile(PathName,FileName,num_samples_to_process);
        getdata(return_matrix, FileName, num_columns);
    end;

    count = cast((num_columns / 2), 'int16');
```

```
fidX1=fopen(strcat(PathName,'processeddataX1.txt'), 'a', 'native');
fidY1=fopen(strcat(PathName,'processeddataY1.txt'), 'a', 'native');
fidZ1=fopen(strcat(PathName,'processeddataZ1.txt'), 'a', 'native');

fidX2=fopen(strcat(PathName,'processeddataX2.txt'), 'a', 'native');
fidY2=fopen(strcat(PathName,'processeddataY2.txt'), 'a', 'native');
fidZ2=fopen(strcat(PathName,'processeddataZ2.txt'), 'a', 'native');

fprintf(fidX1, '\t');
fprintf(fidY1, '\t');
fprintf(fidZ1, '\t');

fprintf(fidX2, '\t');
fprintf(fidY2, '\t');
fprintf(fidZ2, '\t');

for col=1:(num_columns - 1)
    fprintf(fidX1, '%+3dm\t', count - col);
    fprintf(fidY1, '%+3dm\t', count - col);
    fprintf(fidZ1, '%+3dm\t', count - col);

    fprintf(fidX2, '%+3dm\t', count - col);
    fprintf(fidY2, '%+3dm\t', count - col);
    fprintf(fidZ2, '%+3dm\t', count - col);

    count - col
end;
fprintf(fidX1, '%+3dm\n', count - col - 1);
fprintf(fidY1, '%+3dm\n', count - col - 1);
fprintf(fidZ1, '%+3dm\n', count - col - 1);

fprintf(fidX2, '%+3dm\n', count - col - 1);
fprintf(fidY2, '%+3dm\n', count - col - 1);
fprintf(fidZ2, '%+3dm\n', count - col - 1);

count - col - 1

for row=1:num_rows
    fprintf(fidX1, '%+3dm\t', row);
    fprintf(fidY1, '%+3dm\t', row);
    fprintf(fidZ1, '%+3dm\t', row);

    fprintf(fidX2, '%+3dm\t', row);
    fprintf(fidY2, '%+3dm\t', row);
    fprintf(fidZ2, '%+3dm\t', row);

    for col=1:(num_columns - 1)
        fprintf(fidX1, '%+5.4f\t', X1(row, col));
        fprintf(fidY1, '%+5.4f\t', Y1(row, col));
        fprintf(fidZ1, '%+5.4f\t', Z1(row, col));

        fprintf(fidX2, '%+5.4f\t', X2(row, col));
        fprintf(fidY2, '%+5.4f\t', Y2(row, col));
        fprintf(fidZ2, '%+5.4f\t', Z2(row, col));
    end;
```

```
        fprintf(fidX1, '%+5.4f\n', X1(row, col + 1));
        fprintf(fidY1, '%+5.4f\n', Y1(row, col + 1));
        fprintf(fidZ1, '%+5.4f\n', Z1(row, col + 1));

        fprintf(fidX2, '%+5.4f\n', X2(row, col + 1));
        fprintf(fidY2, '%+5.4f\n', Y2(row, col + 1));
        fprintf(fidZ2, '%+5.4f\n', Z2(row, col + 1));
    end;

    fclose(fidX1);
    fclose(fidY1);
    fclose(fidZ1);

    fclose(fidX2);
    fclose(fidY2);
    fclose(fidZ2);


function gd=getdata(return_matrix, FileName, num_columns)
    global X1 Y1 Z1 X2 Y2 Z2

    row = str2num(FileName(1:(strfind(FileName, 'm;') - 1)));
    col = str2num(FileName((strfind(FileName, 'm;') + 3):(strfind(FileName,
'm ctr') - 1)));

    count = cast((num_columns / 2), 'int16');

    X1(row, count - col) = return_matrix(1,1);
    Y1(row, count - col) = return_matrix(1,2);
    Z1(row, count - col) = return_matrix(1,3);
    X2(row, count - col) = return_matrix(1,4);
    Y2(row, count - col) = return_matrix(1,5);
    Z2(row, count - col) = return_matrix(1,6);


function pf=processfile(PathName,FileName, num_samples_to_process)

    global MaxCode MinCode MaxV MinV

    fid=fopen(strcat(PathName,FileName));
    InputText=textscan(fid,'%s %s %s %s %s %s %s %s %s','delimiter','\t');
    fclose(fid);

    col1=InputText{1};
    col2=InputText{2};
    col3=InputText{3};
    col4=InputText{4};
    col6=InputText{6};
    col7=InputText{7};
    col8=InputText{8};

    count = 0;
```

```
for i=1:length(col1);
    count = count + 1;

    if (strcmp(col1(i), '[Channel Data]'))
        break;
    elseif (strcmp(col1(i), 'Max Voltage'))
        MaxV=str2double(col2(i));       %V
    elseif (strcmp(col1(i), 'Min Voltage'))
        MinV=str2double(col2(i));       %V
    elseif (strcmp(col1(i), 'Max Code'))
        MaxCode=str2double(col2(i));    %2^23 -1 i.e. +60000nT
    elseif (strcmp(col1(i), 'Min Code'))
        MinCode=str2double(col2(i));    %-2^23   i.e. -60000nT
    elseif (strcmp(col1(i), 'Sampling Frequency'))
        SampF=str2double(col2(i));      %sps
    elseif (strcmp(col1(i),'Notes'))
        Note=char(col2(i));
    elseif (strcmp(col1(i), 'Date and Time'))
        DateandTime=char(col2(i));
    end;
end;

B0x1=-123;    %nT
B0y1=-54;     %nT
B0z1=-136;    %nT
B0x2=4;       %nT
B0y2=-93;     %nT
B0z2=-148;    %nT
S=36.5;       %uV/nT

if (num_samples_to_process == 0)
    numsamples=length(col2) - count;
else
    numsamples = num_samples_to_process;
end;

%Pre-allocate arrays
x1 = zeros(1, numsamples);
y1 = zeros(1, numsamples);
z1 = zeros(1, numsamples);

x2 = zeros(1, numsamples);
y2 = zeros(1, numsamples);
z2 = zeros(1, numsamples);

Vx1 = zeros(1, numsamples);
Vy1 = zeros(1, numsamples);
Vz1 = zeros(1, numsamples);
Vx2 = zeros(1, numsamples);
Vy2 = zeros(1, numsamples);
Vz2 = zeros(1, numsamples);

Bx1 = zeros(1, numsamples);
By1 = zeros(1, numsamples);
Bz1 = zeros(1, numsamples);
```

```
    Bx2 = zeros(1, numsamples);
    By2 = zeros(1, numsamples);
    Bz2 = zeros(1, numsamples);

    for i=count+1:numsamples+count
        x1(i-count)=str2double(col2(i));
        y1(i-count)=str2double(col3(i));
        z1(i-count)=str2double(col4(i));

        x2(i-count)=str2double(col6(i));
        y2(i-count)=str2double(col7(i));
        z2(i-count)=str2double(col8(i));
    end;

    for i=1:numsamples
        Vx1(i)=evaluateVolts(x1(i));
        Vy1(i)=evaluateVolts(y1(i));
        Vz1(i)=evaluateVolts(z1(i));

        Vx2(i)=evaluateVolts(x2(i));
        Vy2(i)=evaluateVolts(y2(i));
        Vz2(i)=evaluateVolts(z2(i));

        Bx1(i)=evaluateNanoTesla(Vx1(i),S,B0x1);
        By1(i)=evaluateNanoTesla(Vy1(i),S,B0y1);
        Bz1(i)=evaluateNanoTesla(Vz1(i),S,B0z1);

        Bx2(i)=evaluateNanoTesla(Vx2(i),S,B0x2);
        By2(i)=evaluateNanoTesla(Vy2(i),S,B0y2);
        Bz2(i)=evaluateNanoTesla(Vz2(i),S,B0z2);
    end;

    pf = [mean(Bx1), mean(By1), mean(Bz1), mean(Bx2), mean(By2),
mean(Bz2)];

function voltage = evaluateVolts(code)
    global MaxCode MinCode MaxV MinV
    if(code >= 0)
        voltage=code/MaxCode*MaxV;
    elseif(code < 0)
        voltage=code/MinCode*MinV;
    end;

function nanotesla = evaluateNanoTesla(voltage,S,B0)
    nanotesla=voltage*10^6/S-B0;
```

This page is intentionally left blank.

Appendix I– Propulsion System Characterisation Results

The test results are presented below for the different motor speeds and distances. The sample rate of the ADS1278 was set to 10 kSPS. When the higher sampling frequencies of the ADS1278 (52 kSPS and 105 kSPS) are used the excitation frequency and harmonics become visible. This causes the recursive notch filter implemented in the MATLAB code (see Appendix G) to filter these frequencies out. Doing this causes "ringing" artefacts in the spectra (frequency domain) and a large impulse response (time domain) in the filtered data due to the excessive phase shift caused by multiple notch filters. Hence the low sample rate is preferred when removing the effects of the AC mains and DCBL motor.

1.1 Initial Propulsion System Test Results (Comparison of various distances of the DCBL motor to the sensor at a fixed motor speed)

The figures for the initial propulsion system testing at different distances are given in full below. The values found in the tables (Table I - 1 and Table I - 2) and figures (Figure I - 1 through Figure I - 12) below were calculated/generated using the code found in Appendix G run using MATLAB 2009a. The values for the PTP readings were "eyeballed" from the noise plots while the standard deviation and mean values were calculated and displayed on the noise plots. The peak values for power were read off the periodograms using the MATLAB data cursor tool.

The results for the 500 mm sensor to DCBL motor tests are given in Figure I - 1 through Figure I - 6 below. A summary of these plots is given in Table I - 1. The results for the 1100 mm sensor to DCBL motor tests are given in Figure I - 7 through Figure I - 12 below. A summary of these plots is given in Table I - 2.

1.1.1 Test Results: 500 mm from DCBL

Table I - 1: Summarised data for 500 mm spacing DCBL motor (@ 2160 RPM) to the magnetometers

	Bx	***By***	***Bz***
Sensor 1			
Noise PTP before notch filter(s) applied	± 3.5 nT		
TMI before notch filters applied	25 760.29 nT		
Std. dev. before notch filter(s) applied	0.855 nT	1.388 nT	0.680 nT
Mean before notch filter(s) applied	-10858 nT	-295 nT	-23358 nT
Noise PTP after notch filter(s) applied	± 2.0 nT		
TMI after notch filters applied	25 760.29 nT		
Std. dev. after notch filter(s) applied	0.628 nT	0.490 nT	0.563 nT
Mean after notch filter(s) applied	-10858 nT	-295 nT	-23358 nT
Power @ 36 Hz	-92 dB/Hz	-85 dB/Hz	-107 dB/Hz
Power @ 50 Hz	N/A	N/A	-97 dB/Hz
Sensor 2			
Noise PTP before notch filter(s) applied	± 3.5 nT		
TMI before notch filters applied	26 320.44 nT		
Std. dev. before notch filter(s) applied	1.238 nT	1.425 nT	0.699 nT
Mean before notch filter(s) applied	-11603 nT	325 nT	-23623 nT
Noise PTP after notch filter(s) applied	± 2.0 nT		
TMI after notch filters applied	26 320.49 nT		
Std. dev. after notch filter(s) applied	0.870 nT	0.534 nT	0.582 nT
Mean after notch filter(s) applied	-11603 nT	325 nT	-23623 nT
Power @ 36 Hz	-89 dB/Hz	-85 dB/Hz	-108 dB/Hz
Power @ 50 Hz	N/A	N/A	-98 dB/Hz

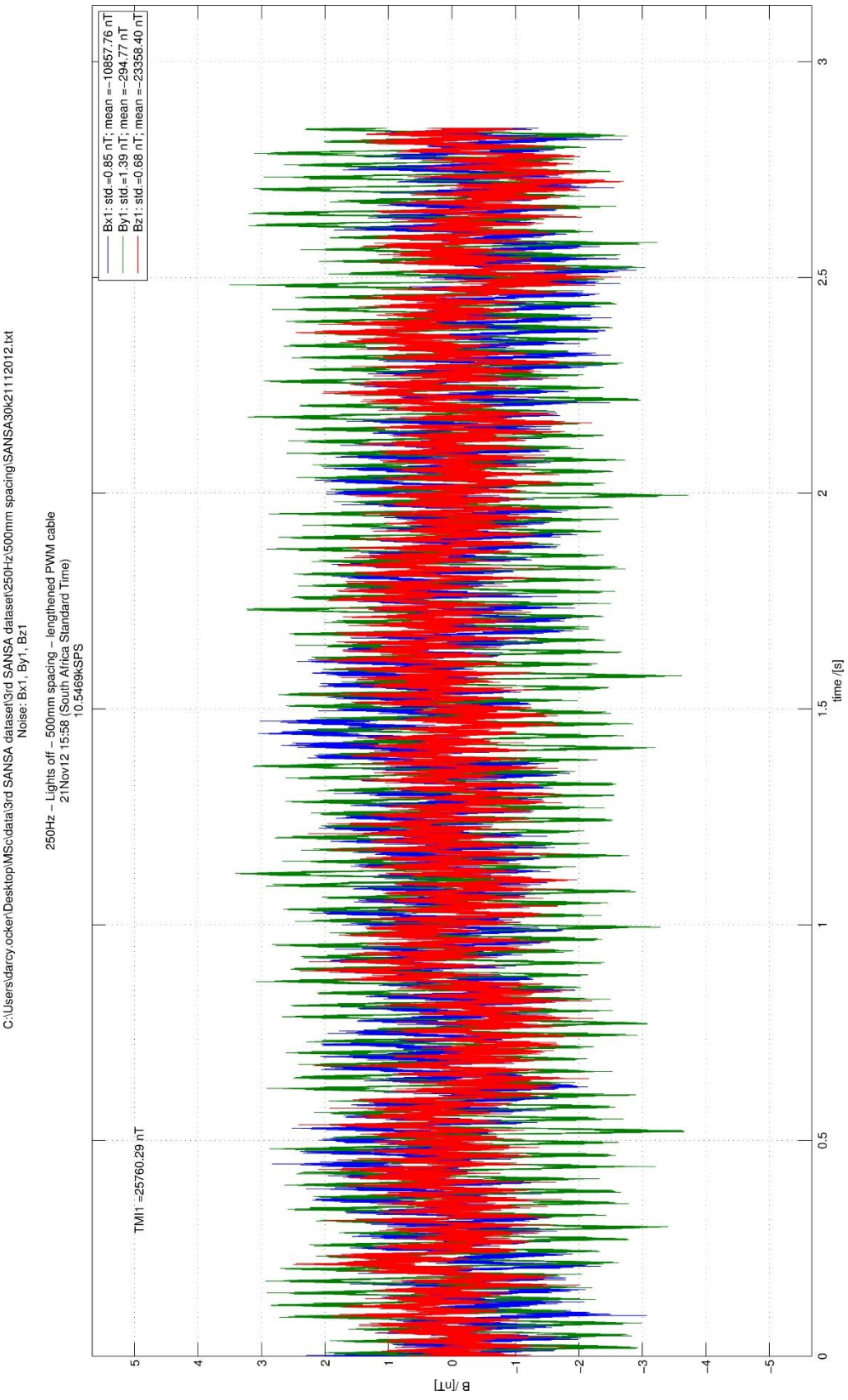


Figure I - 1: Noise vs. time of sensor 1 at 500 mm from the DCBL motor (2160 RPM).

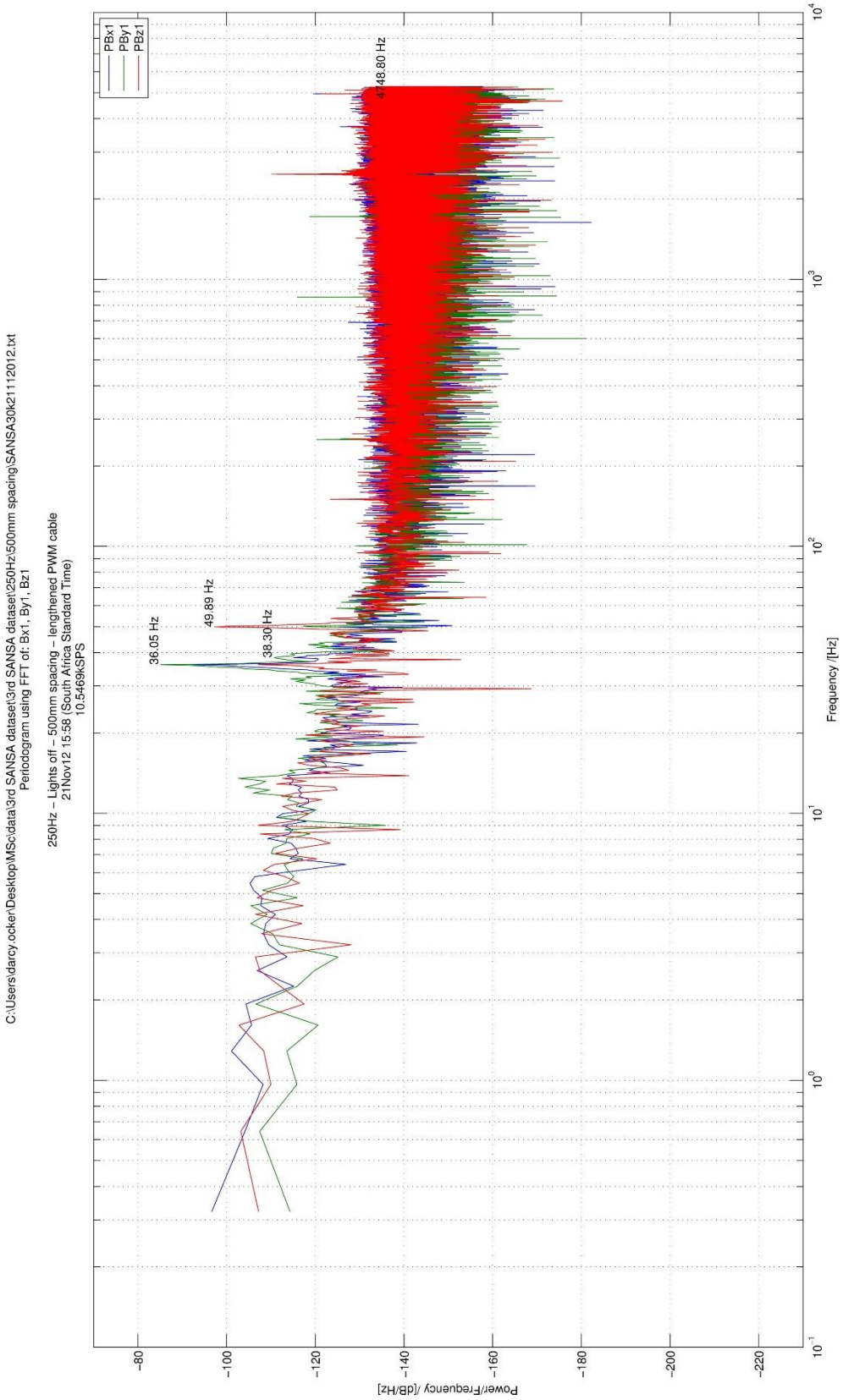


Figure I - 2: Spectral analysis of sensor 1 at 500 mm from the DCBL motor (2160 RPM). The reason for the peaks at 2.4 kHz and harmonic at 4.8 kHz is unknown. These only appeared in data measured in November 2012.

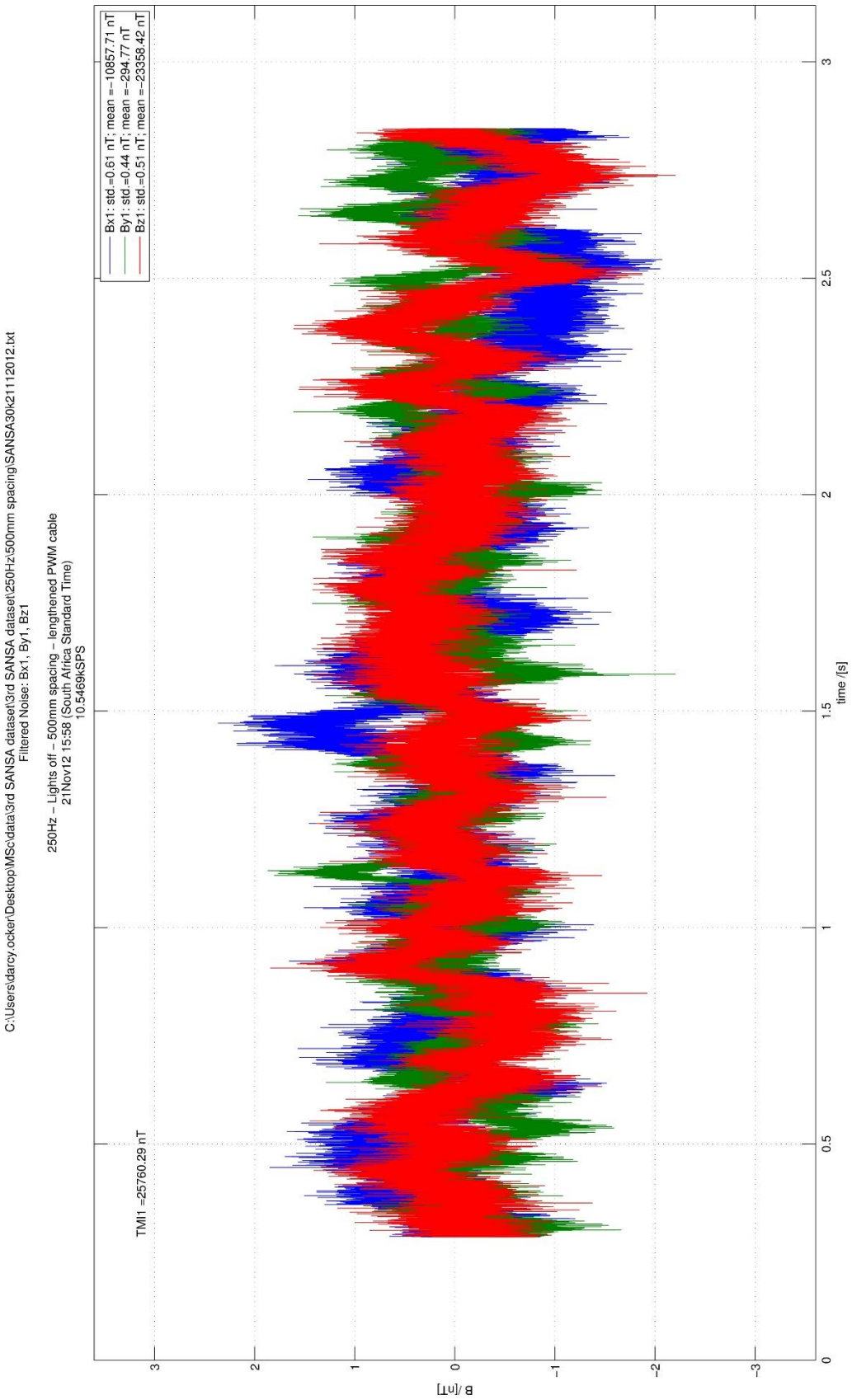


Figure I - 3: Noise vs. time (notch filters applied) of sensor 1 at 500 mm from the DCBL motor (2160 RPM).

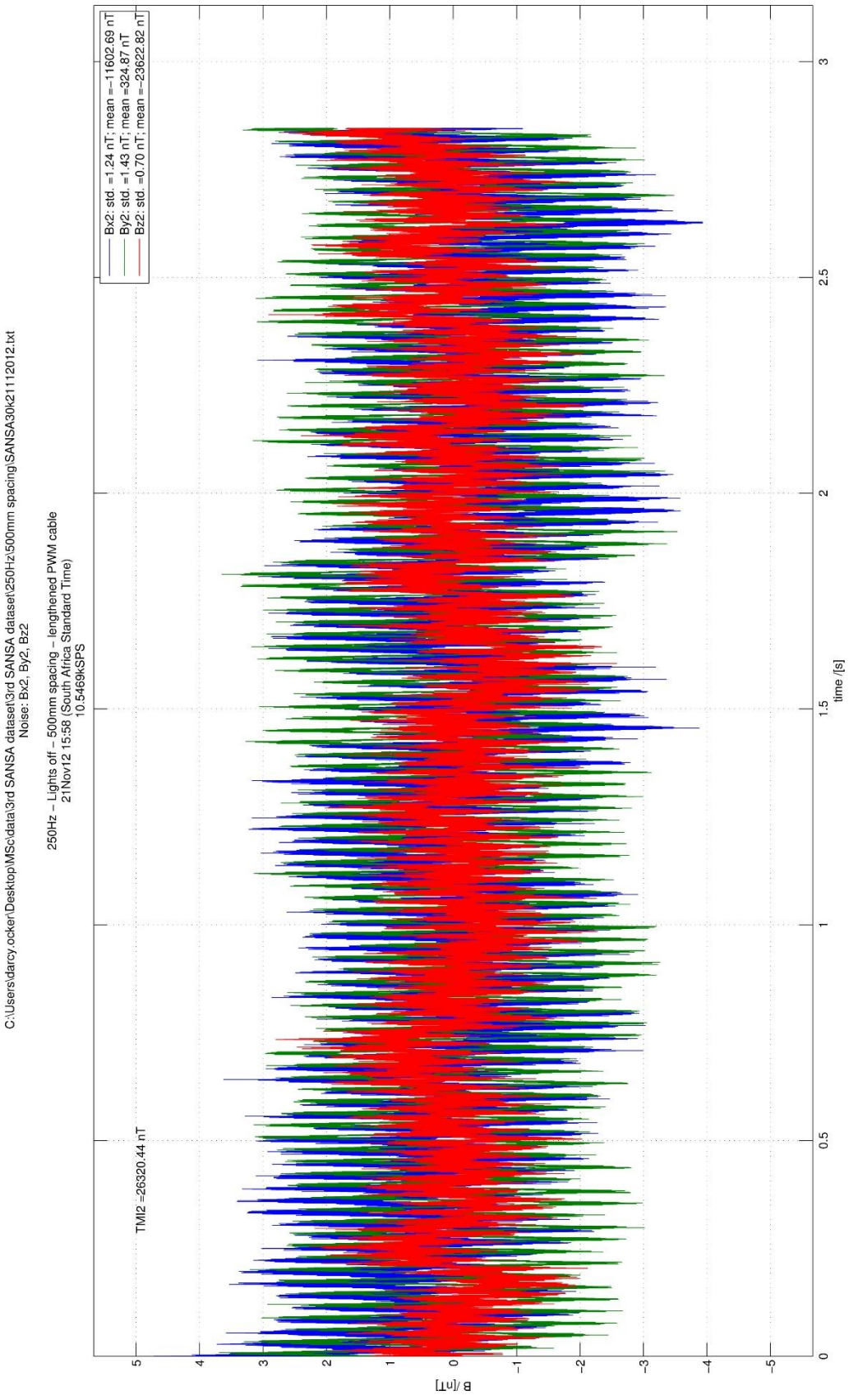


Figure I - 4: Noise vs. time of sensor 2 at 500 mm from the DCBL motor (2160 RPM).

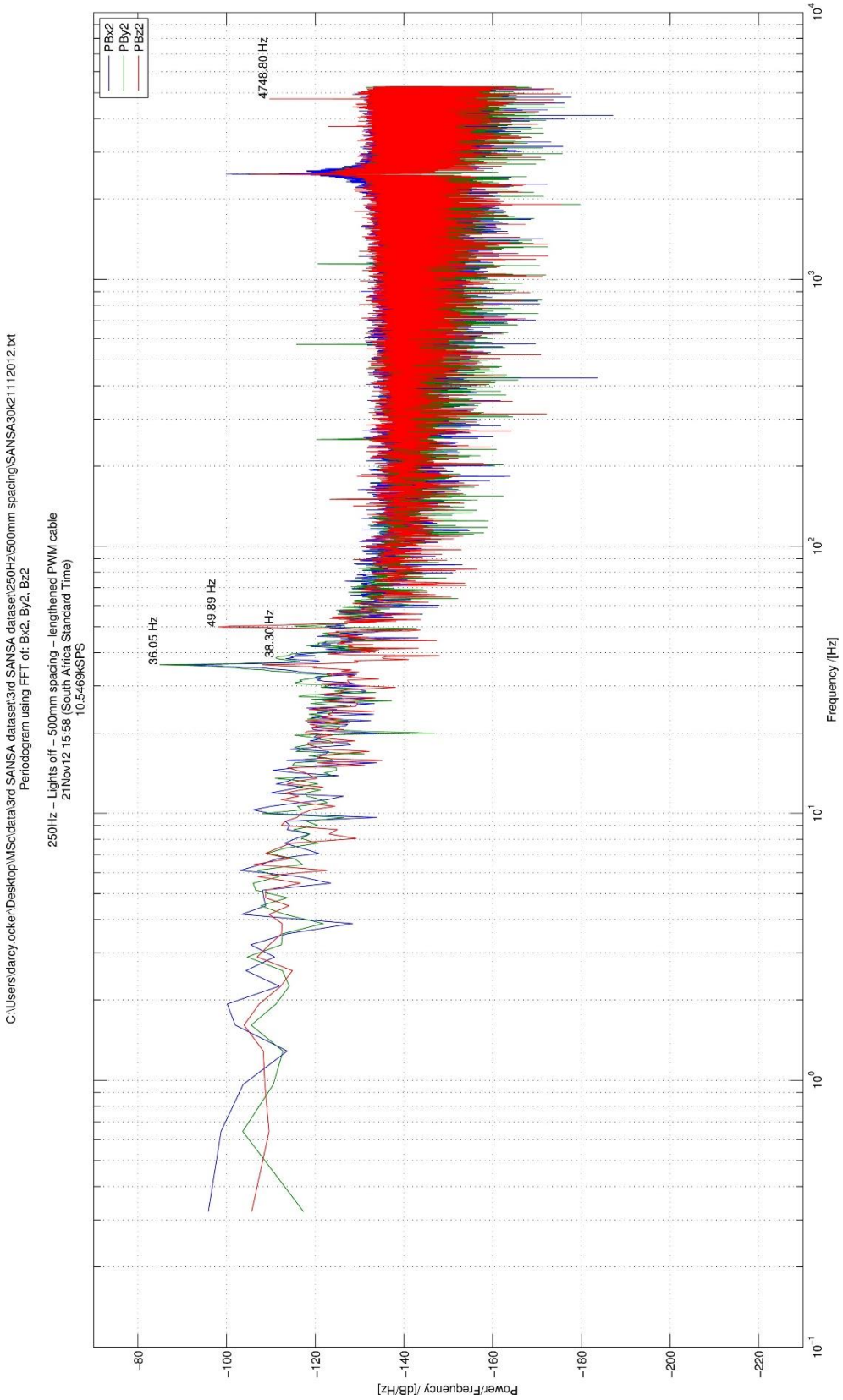


Figure I - 5: Spectral analysis of sensor 2 at 500 mm from the DCBL motor (2160 RPM). The reason for the peaks at 2.4 kHz and harmonic at 4.8 kHz is unknown. These only appeared in data measured in November 2012.

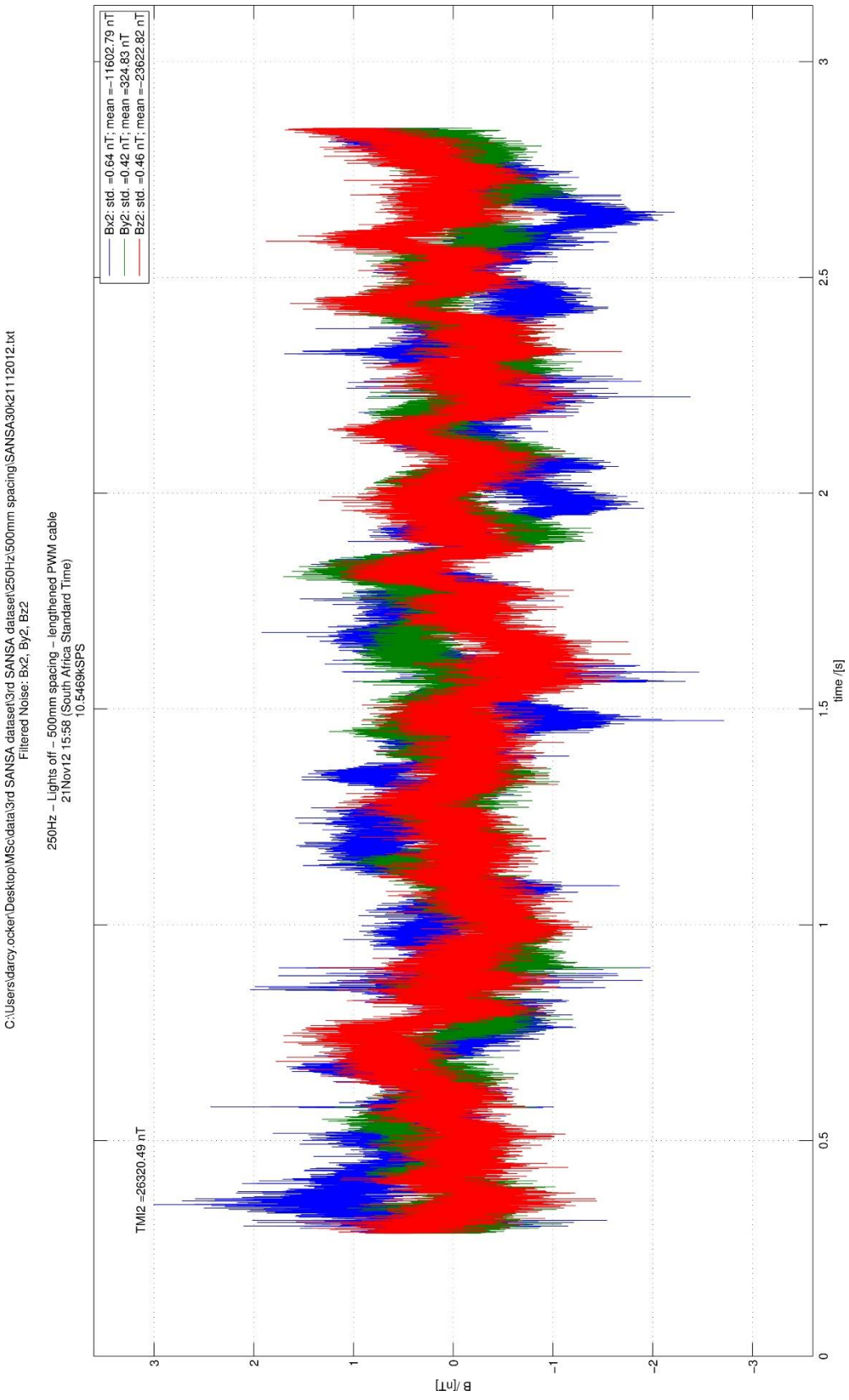


Figure I - 6: Noise vs. time (notch filters applied) of sensor 2 at 500 mm from the DCBL motor (2160 RPM).

1.1.2 Test Results: 1100 mm from DCBL

Table I - 2: Summarised data for 1100 mm spacing DCBL motor (@ 2160 RPM) to the magnetometers

	Bx	***By***	***Bz***
Sensor 1			
Noise PTP before notch filter(s) applied	± 2 nT		
TMI before notch filters applied	25 676.69 nT		
Std. dev. before notch filter(s) applied	0.634 nT	0.502 nT	0.523 nT
Mean before notch filter(s) applied	-10832 nT	-318 nT	-23278 nT
Noise PTP after notch filter(s) applied	± 1.5 nT		
TMI after notch filters applied	25 676.66 nT		
Std. dev. after notch filter(s) applied	0.428 nT	0.362 nT	0.386 nT
Mean after notch filter(s) applied	-10832 nT	-318 nT	-23278 nT
Power @ 36 Hz	N/A	-106 dB/Hz	N/A
Power @ 50 Hz	N/A	N/A	-111 dB/Hz
Sensor 2			
Noise PTP before notch filter(s) applied	± 2 nT		
TMI before notch filters applied	26 226.25 nT		
Std. dev. before notch filter(s) applied	0.829 nT	0.636 nT	0.646 nT
Mean before notch filter(s) applied	-11583 nT	417 nT	-23526 nT
Noise PTP after notch filter(s) applied	± 1.5 nT		
TMI after notch filters applied	26 226.27 nT		
Std. dev. after notch filter(s) applied	0.467 nT	0.393 nT	0.448 nT
Mean after notch filter(s) applied	-11583 nT	417 nT	-23526 nT
Power @ 36 Hz	N/A	-105 dB/Hz	N/A
Power @ 50 Hz	N/A	N/A	-111 dB/Hz

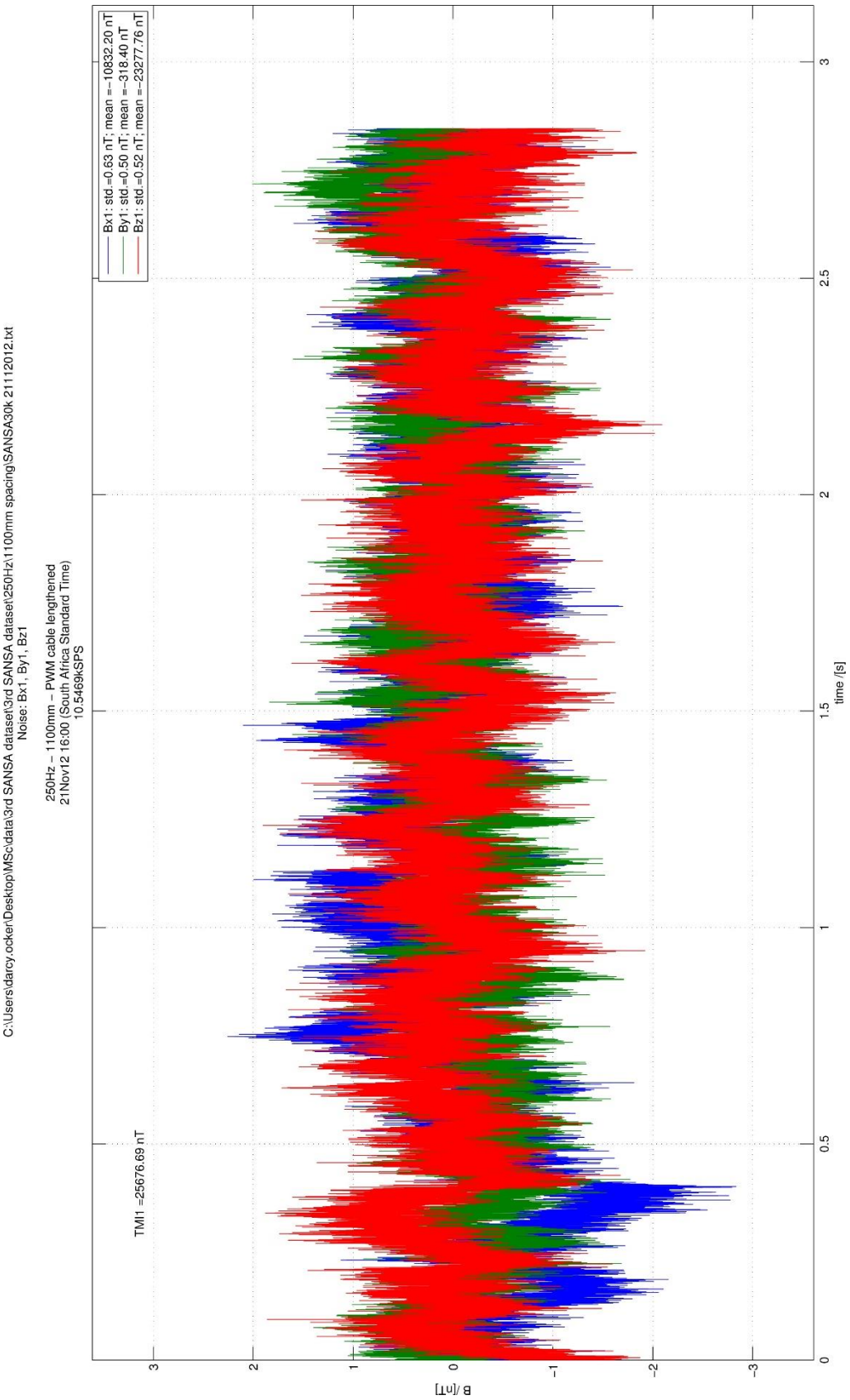


Figure I - 7: Noise vs. time of sensor 1 at 1100 mm from the DCBL motor (2160 RPM).

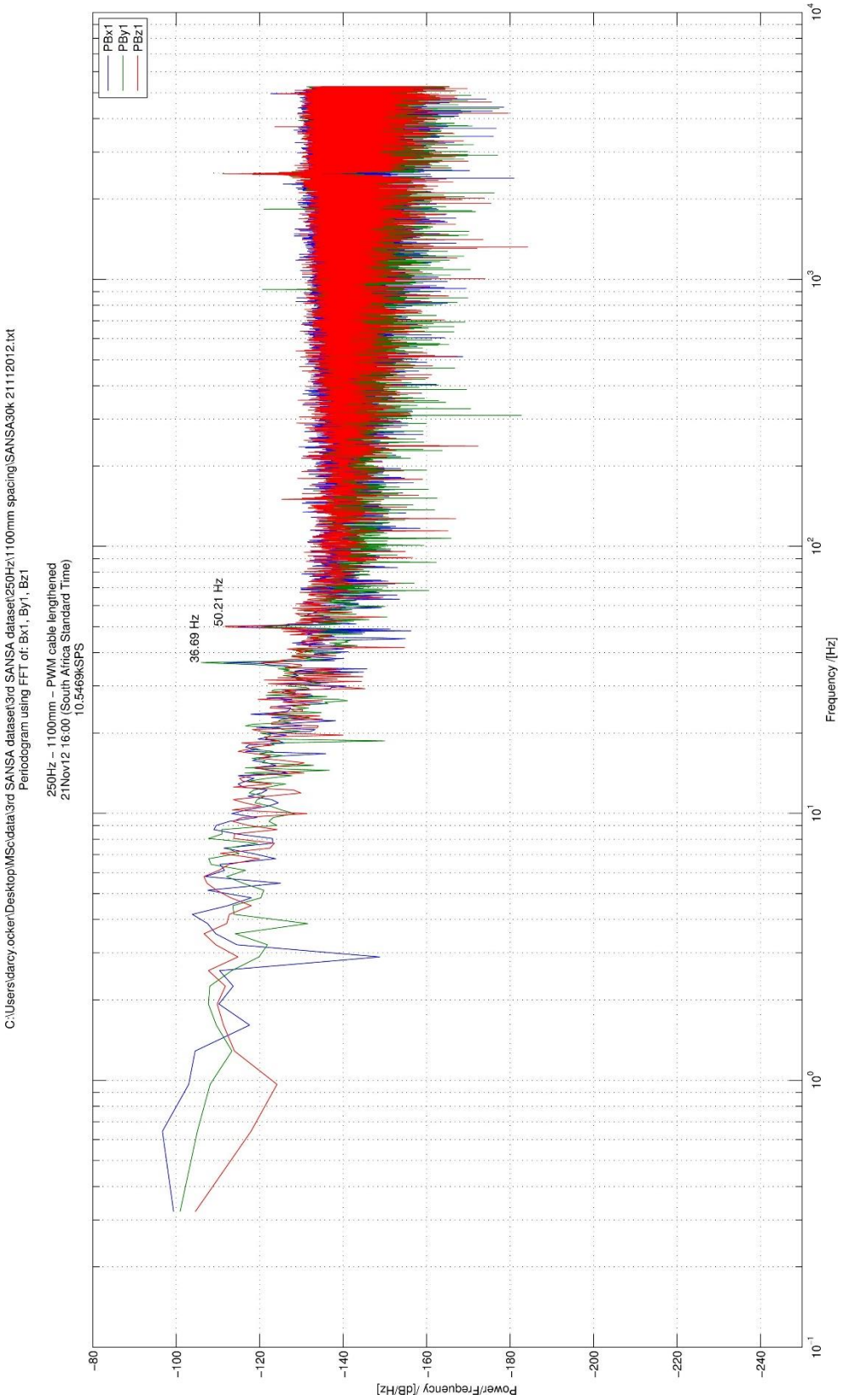


Figure I - 8: Spectral analysis of sensor 1 at 1100 mm from the DCBL motor (2160 RPM). The reason for the peaks at 2.4 kHz and harmonic at 4.8 kHz is unknown. These only appeared in data measured in November 2012.

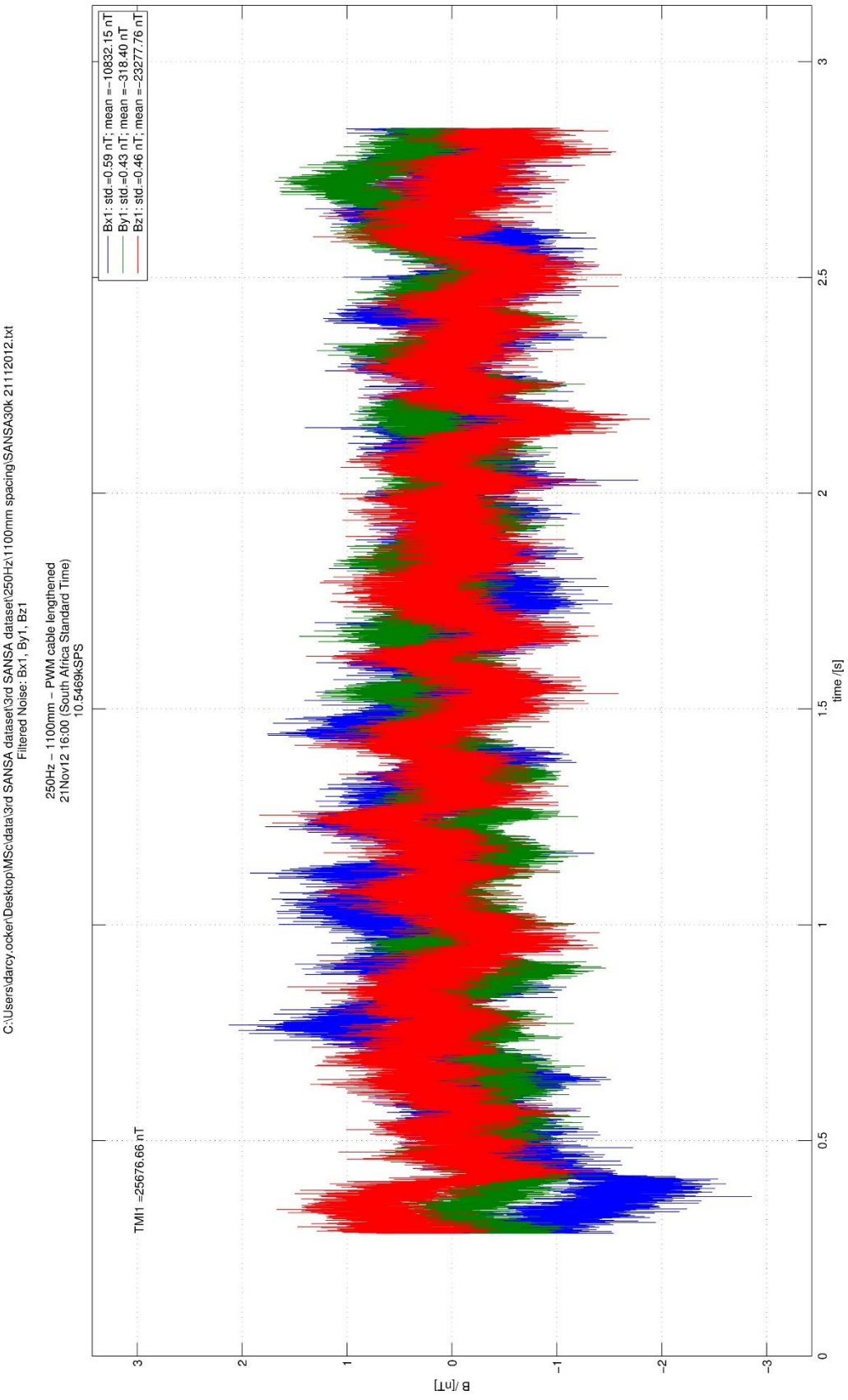


Figure I - 9: Noise vs. time (notch filters applied) of sensor 1 at 1100 mm from the DCBL motor (2160 RPM).

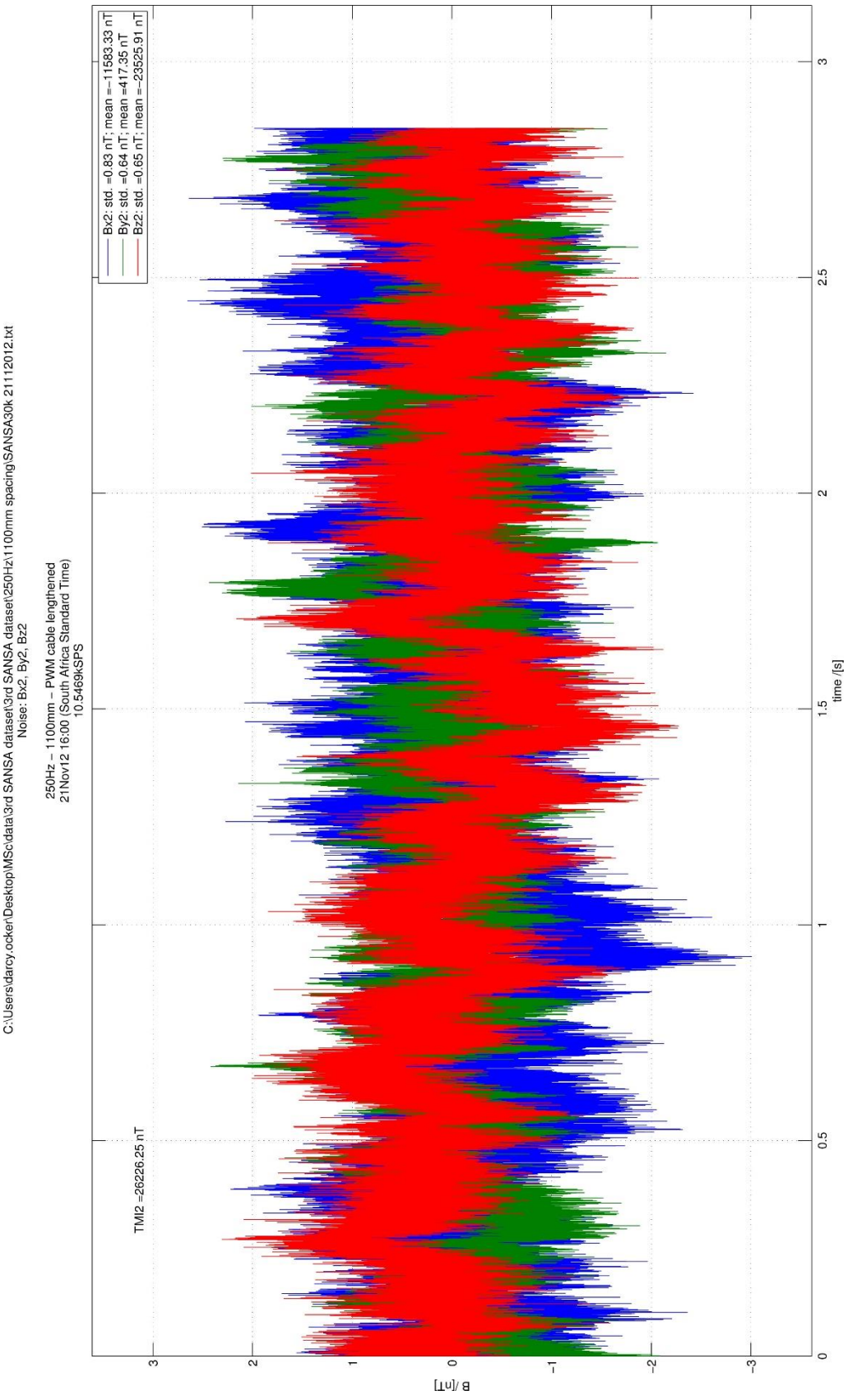


Figure I - 10: Noise vs. time of sensor 2 at 1100 mm from the DCBL motor (2160 RPM).

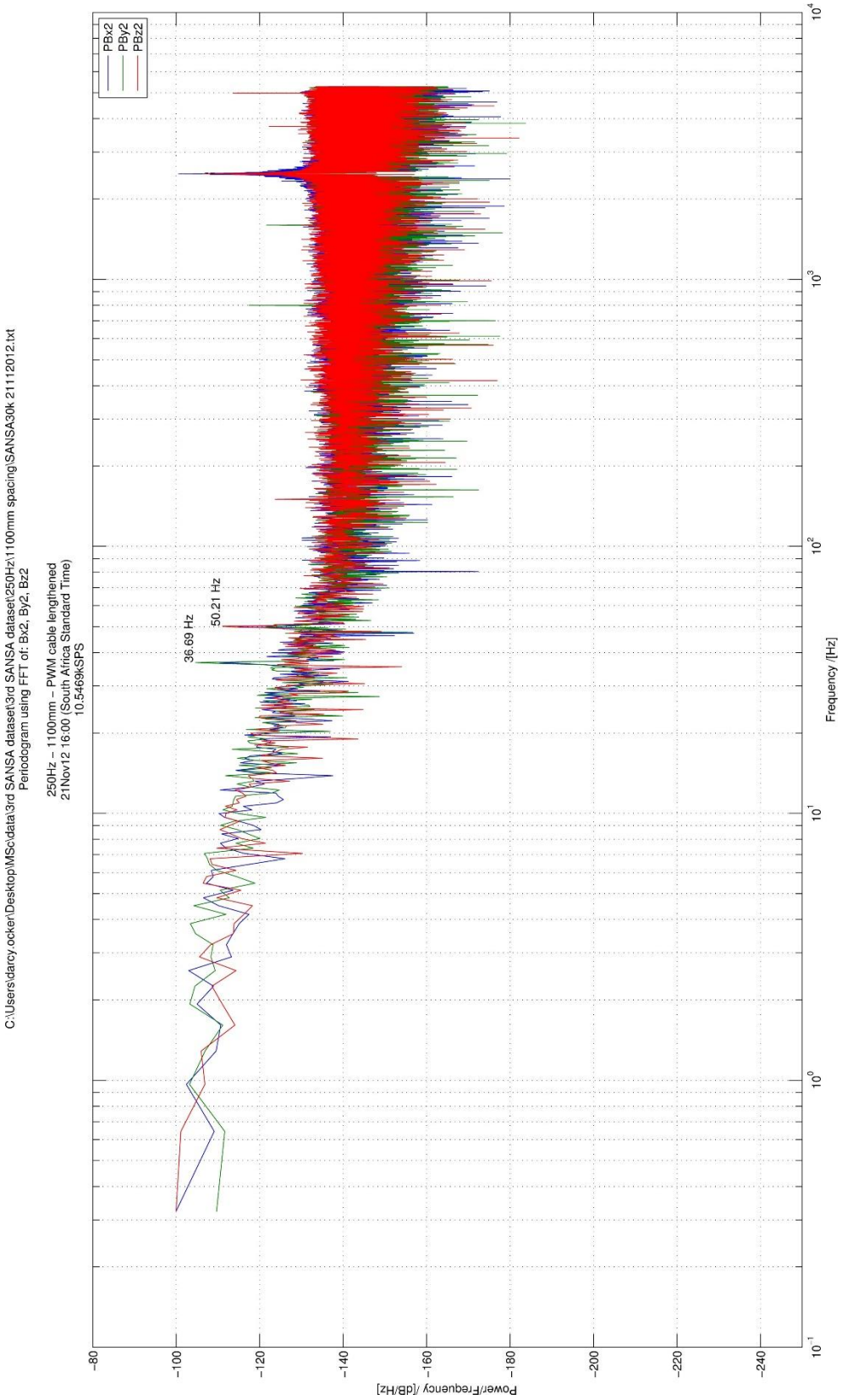


Figure I - 11: Spectral analysis of sensor 2 at 1100 mm from the DCBL motor (2160 RPM). The reason for the peaks at 2.4 kHz and harmonic at 4.8 kHz is unknown. These only appeared in data measured in November 2012.

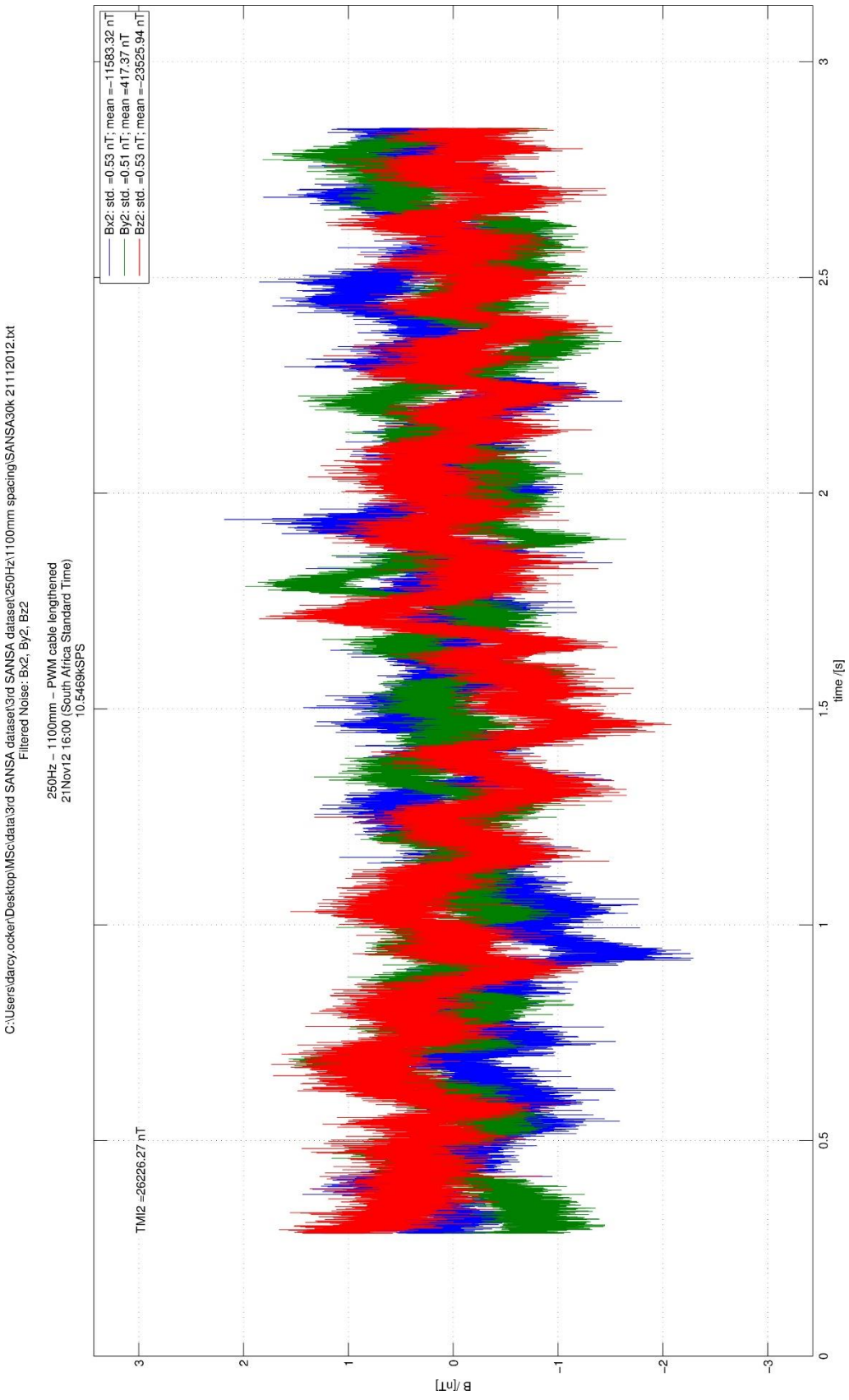


Figure I - 12: Noise vs. time (notch filters applied) of sensor 2 at 1100 mm from the DCBL motor (2160 RPM).

1.2 UAV Platform Tests (Comparison of DCBL motor signature at various motor speeds):

The figures for the various motor speeds/throttle openings are presented in full below.

The results are given as follows:

1. 0 % throttle: Figure I - 13 through Figure I - 18
2. 25 % throttle: Figure I - 19 through Figure I - 24
3. 50% throttle: Figure I - 25 through Figure I - 30
4. 75 % throttle: Figure I - 31 through Figure I - 36
5. 100 % throttle: Figure I - 37 through Figure I - 42

1.2.1 Test Results: 0 % throttle

The tabulated results below were acquired with the UAV powered but the throttle setting was 0 %. This gives the baseline for the test results that follow.

Note that with filtering the AC mains signal the noise level of the magnetometer and ADC can be dropped to similar levels of the LEMI-011b specification.

Table I - 3: Summary data for UAV platform test with DCBL motor at 0 RPM

	Bx	***By***	***Bz***
Sensor 1			
Noise PTP before notch filter(s) applied	± 3.0 nT		
TMI before notch filters applied	25 432.81 nT		
Std. dev. before notch filter(s) applied	0.528 nT	0.598 nT	1.113 nT
Mean before notch filter(s) applied	172 nT	13111 nT	-21792 nT
Noise PTP after notch filter(s) applied	± 2.0 nT		
TMI after notch filters applied	25 432.77 nT		
Std. dev. after notch filter(s) applied	0.513 nT	0.526 nT	0.547 nT
Mean after notch filter(s) applied	172 nT	13111 nT	-21792 nT
Power @ 50 Hz (AC Mains)	-106 dB/Hz	-98 dB/Hz	-88 dB/Hz
Sensor 2			
Noise PTP before notch filter(s) applied	± 3.0 nT		
TMI before notch filters applied	25 708.83 nT		
Std. dev. before notch filter(s) applied	0.607 nT	0.745 nT	1.079 nT
Mean before notch filter(s) applied	-1292 nT	-11675 nT	-22869 nT
Noise PTP after notch filter(s) applied	± 2.0 nT		
TMI after notch filters applied	25 708.85 nT		
Std. dev. after notch filter(s) applied	0.597 nT	0.527 nT	0.578 nT
Mean after notch filter(s) applied	-1292 nT	-11675 nT	-22868 nT
Power @ 50 Hz (AC Mains)	-107 dB/Hz	-94 dB/Hz	-88 dB/Hz

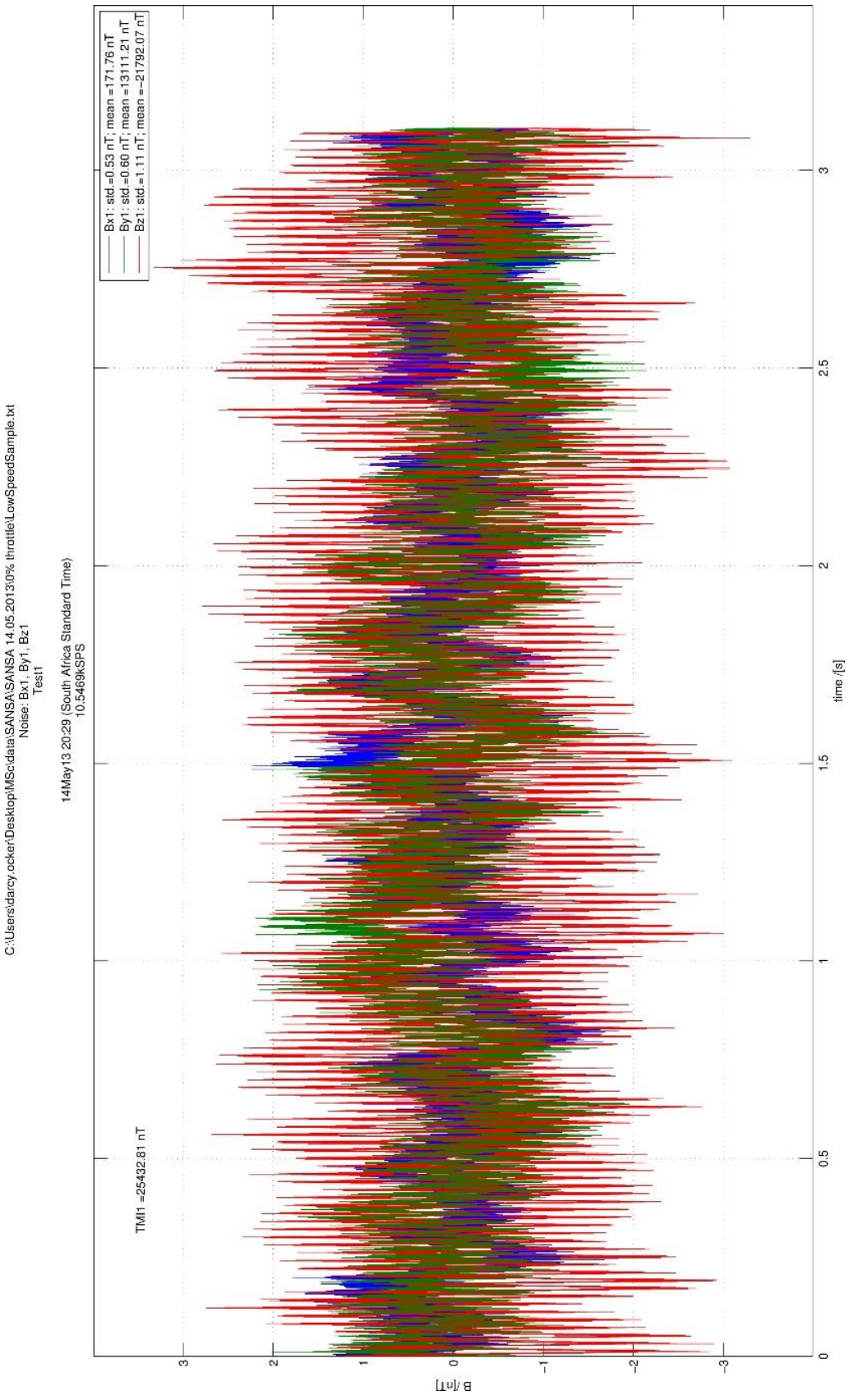


Figure I - 13: Noise vs. time of sensor 1 with DCBL motor at 0 % throttle.

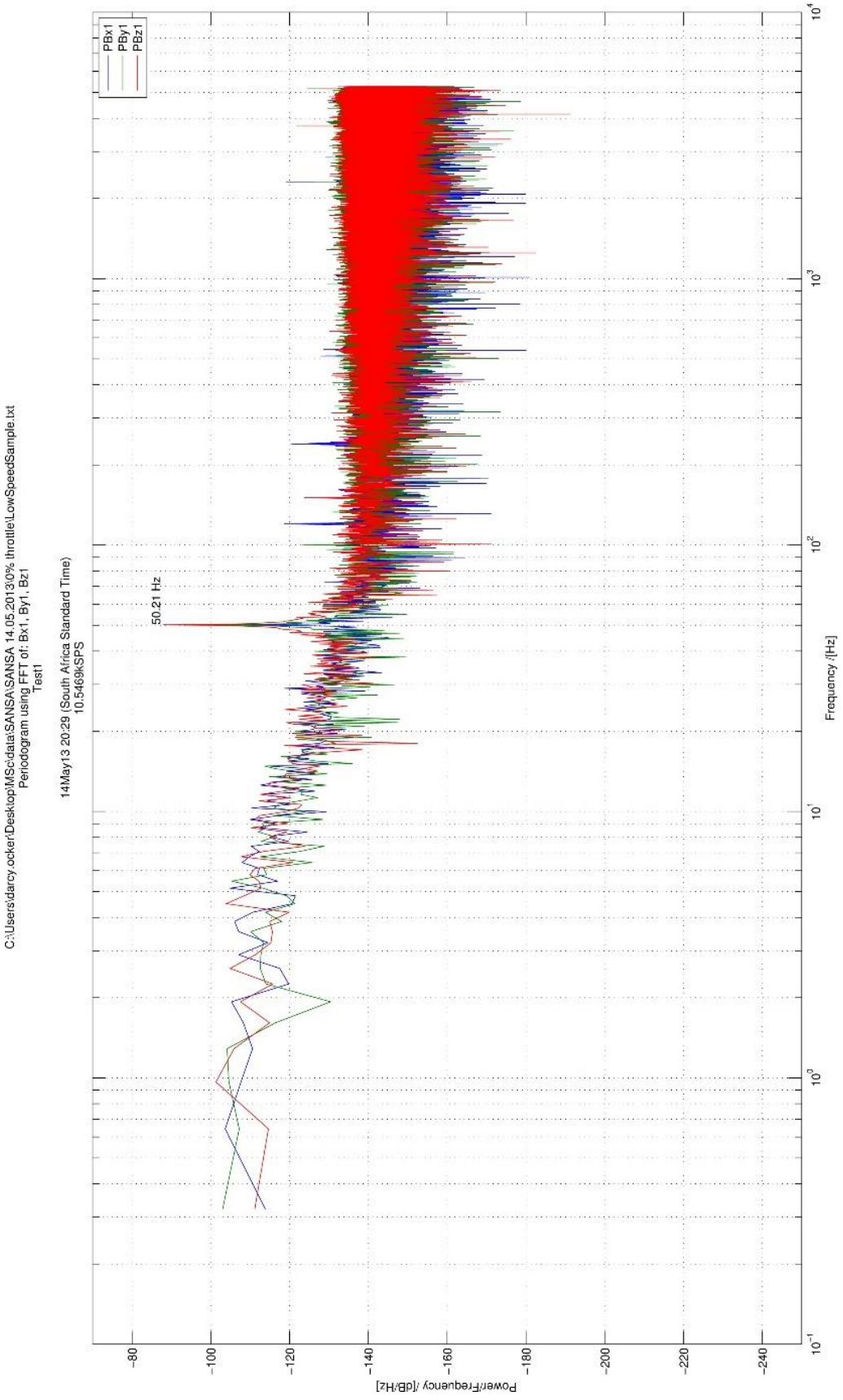


Figure I - 14: Spectral analysis of sensor 1 with DCBL motor at 0 % throttle.

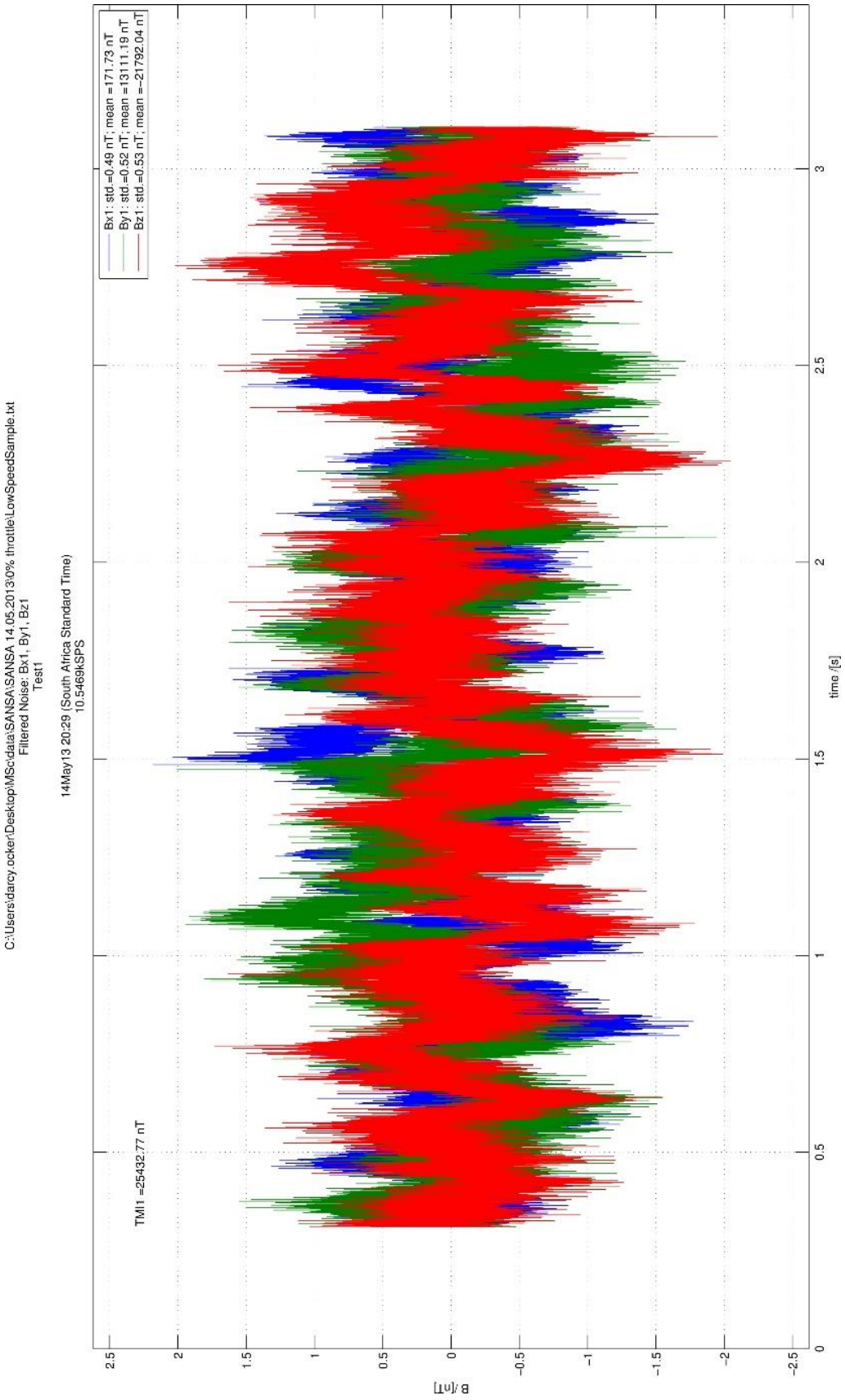


Figure I - 15: Noise vs. time (notch filters applied) of sensor 1 with DCBL motor at 0% throttle.

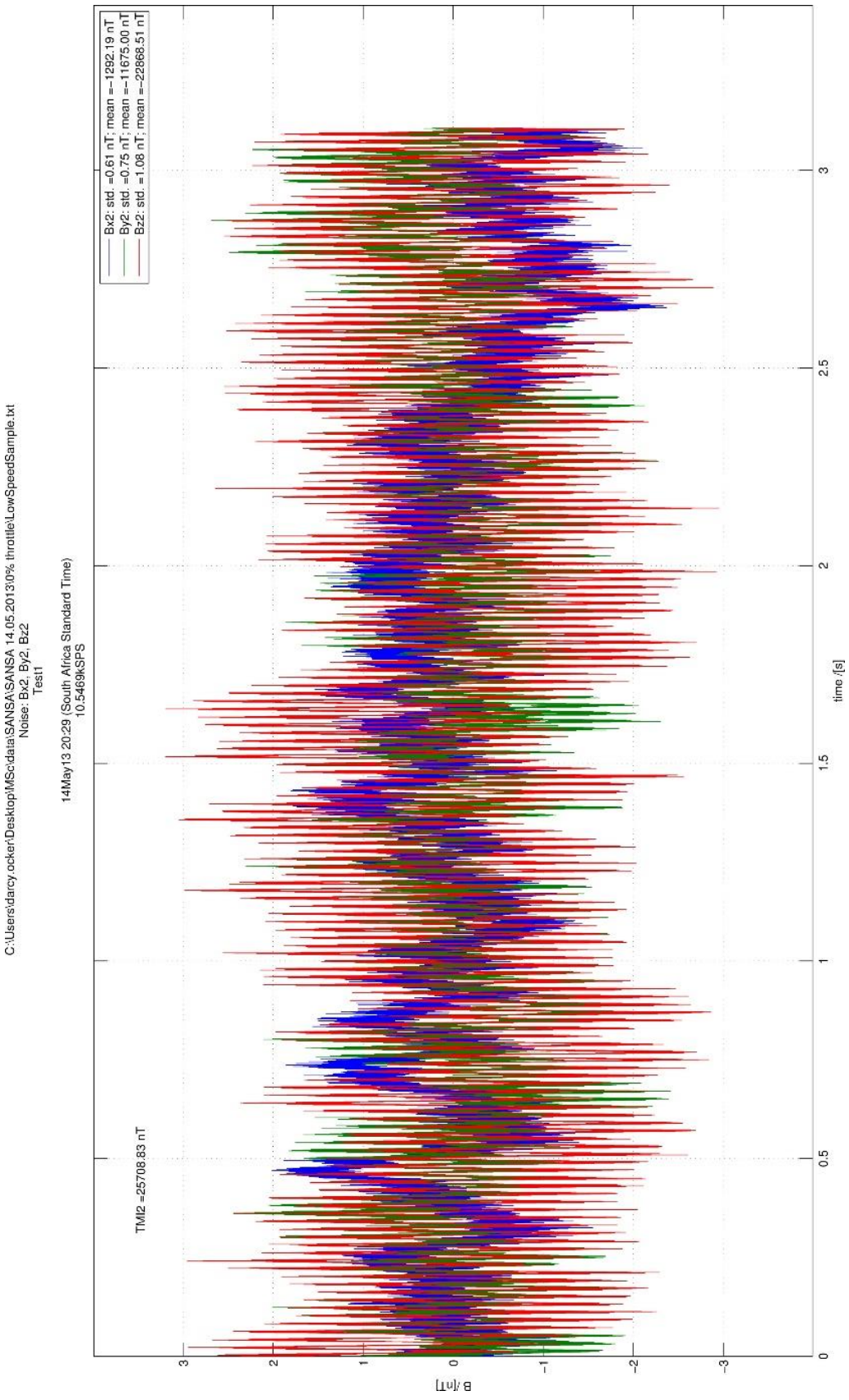


Figure I - 16: Noise vs. time of sensor 2 with DCBL motor at 0 % throttle.

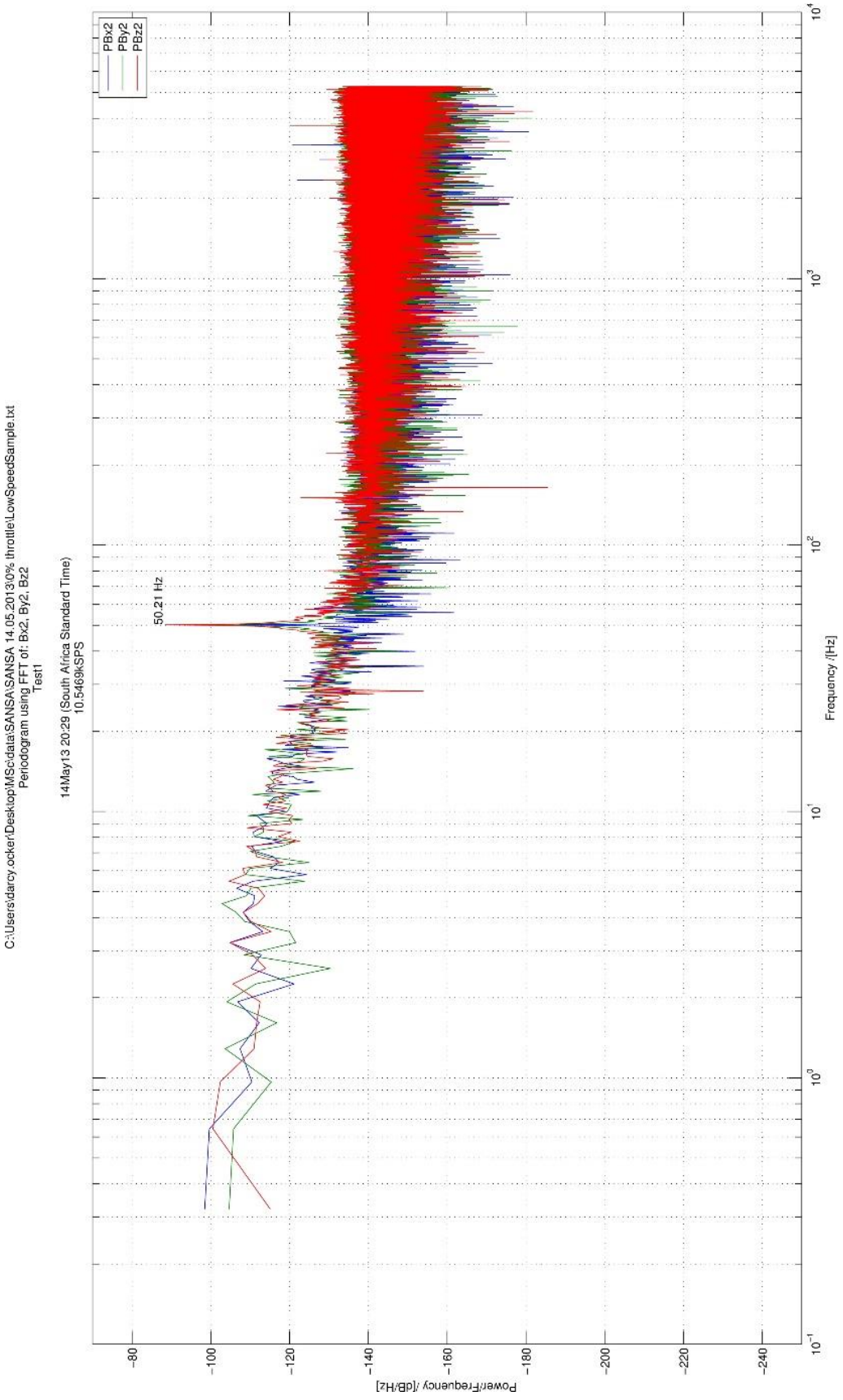


Figure I - 17: Spectral analysis of sensor 2 with DCBL motor at 0 % throttle.

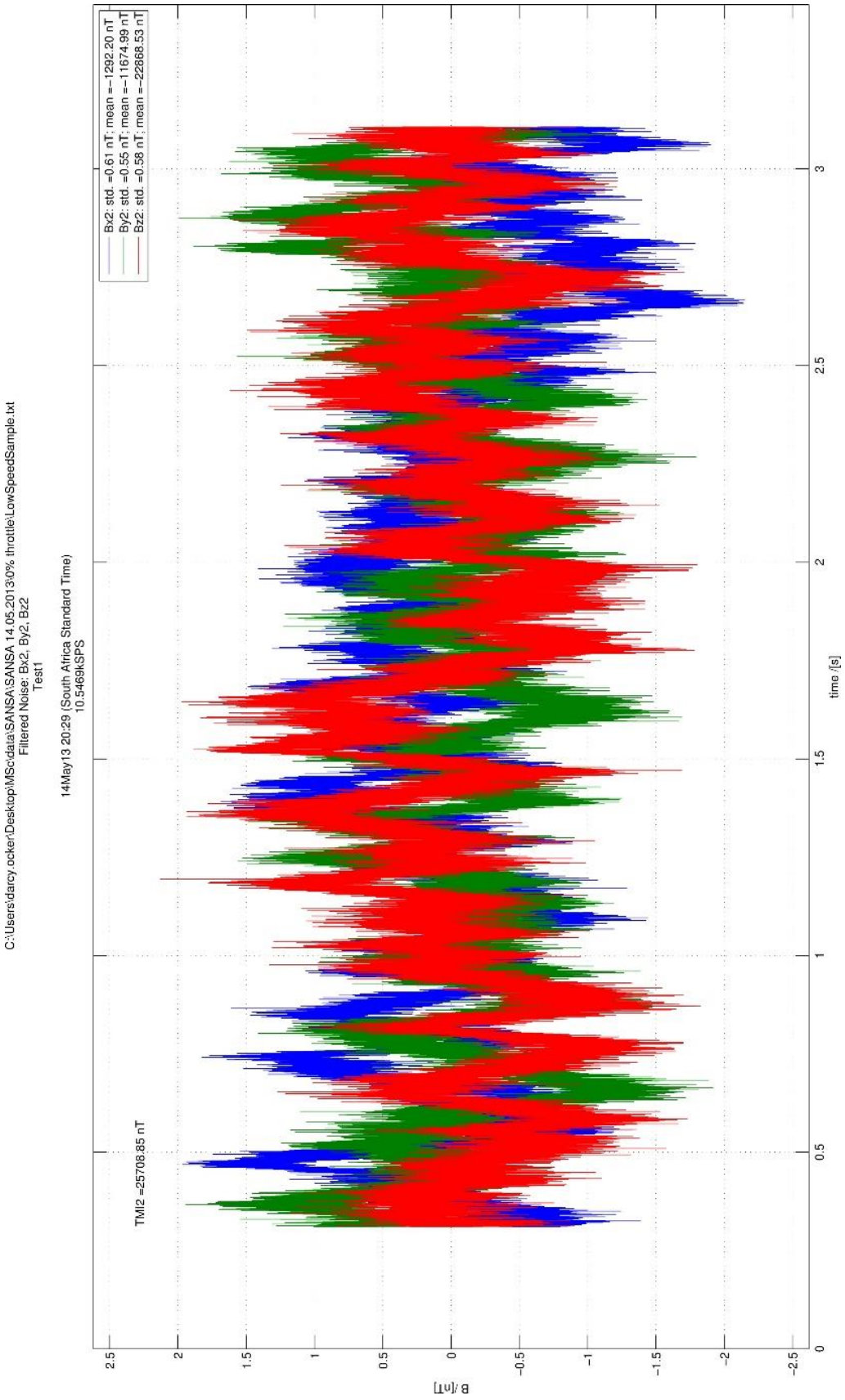


Figure I - 18: Noise vs. time (notch filters applied) of sensor 2 with DCBL motor at 0% throttle.

1.2.2 Test Results: 25 % throttle

The throttle was set to approximately 25 % which allowed the motor to turn at 2144 RPM. The results are tabulated below. This shows two noticeable lines at 50 Hz (AC Mains) and 36 Hz (DCBL motor shaft speed). The current draw on the LiPo batteries at 25 % throttle opening was around 1 A.

With light throttle openings such as this the performance of the combination of the magnetometer and ADC can be similar to that of the LEMI-011b magnetometer specifications.

Table I - 4: Summary data for UAV platform test with DCBL motor at 2144 RPM

	Bx	***By***	***Bz***
Sensor 1			
Noise PTP before notch filter(s) applied	± 3.5 nT		
TMI before notch filters applied	25 435.90 nT		
Std. dev. before notch filter(s) applied	0.688 nT	0.645 nT	1.672 nT
Mean before notch filter(s) applied	177 nT	13094 nT	-21806 nT
Noise PTP after notch filter(s) applied	± 2.0 nT		
TMI after notch filters applied	25 435.91 nT		
Std. dev. after notch filter(s) applied	0.534 nT	0.474 nT	0.491 nT
Mean after notch filter(s) applied	177 nT	13094 nT	-21806 nT
Power @ 36 Hz (DCBL Shaft Speed)	-95 dB/Hz	-113 dB/Hz	-105 dB/Hz
Power @ 50 Hz (AC Mains)	-106 dB/Hz	-96 dB/Hz	-85 dB/Hz
Sensor 2			
Noise PTP before notch filter(s) applied	± 3.5 nT		
TMI before notch filters applied	25 713.35 nT		
Std. dev. before notch filter(s) applied	0.768 nT	0.810 nT	1.650 nT
Mean before notch filter(s) applied	-1290 nT	-11657 nT	-22883 nT
Noise PTP after notch filter(s) applied	± 2.0 nT		
TMI after notch filters applied	25 713.36 nT		
Std. dev. after notch filter(s) applied	0.439 nT	0.538 nT	0.638 nT
Mean after notch filter(s) applied	-1290 nT	-11657 nT	-22883 nT
Power @ 36 Hz (DCBL Shaft Speed)	-92 dB/Hz	N/A	-106 dB/Hz
Power @ 50 Hz (AC Mains)	-105 dB/Hz	-93 dB/Hz	-85 dB/Hz

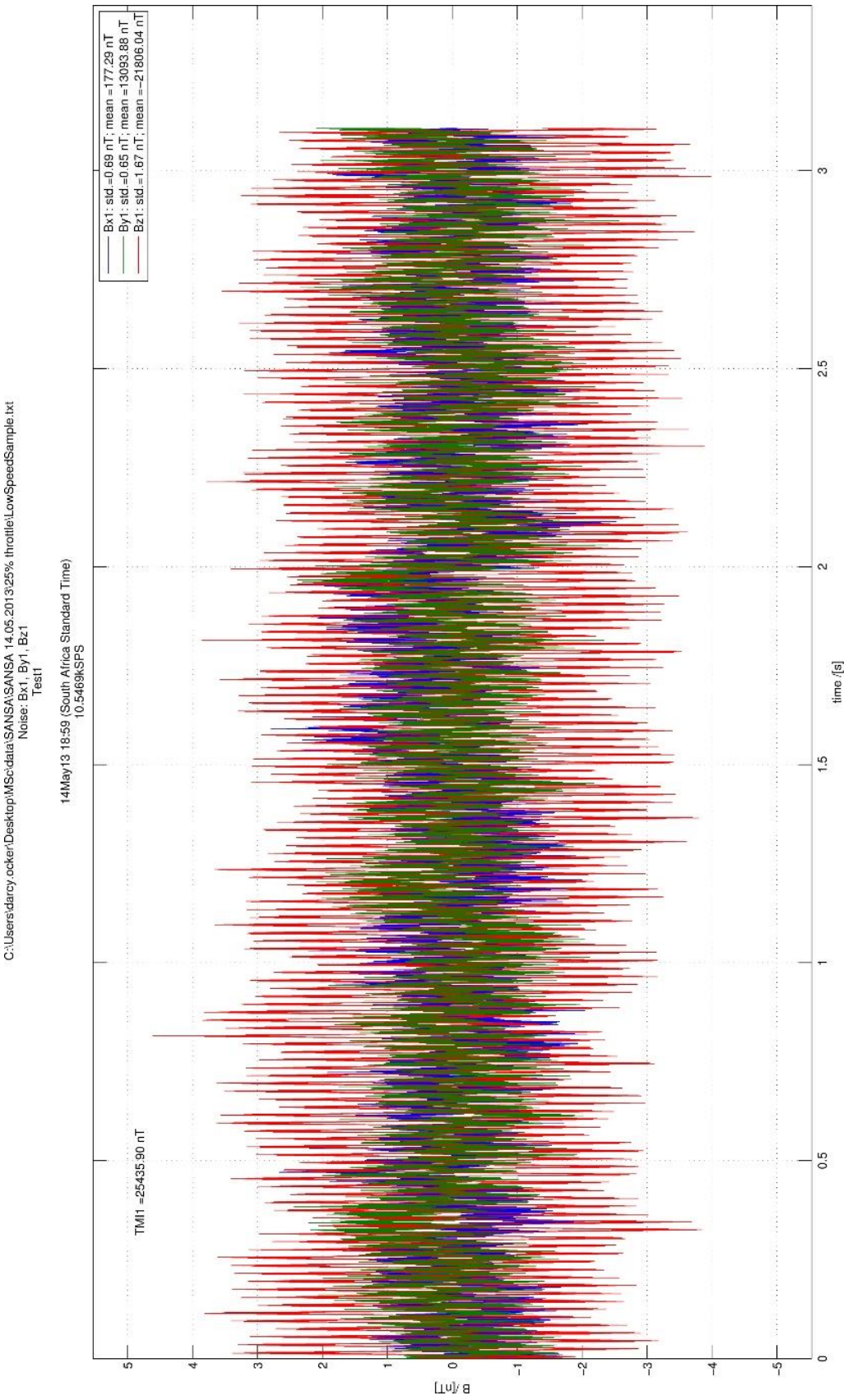


Figure I - 19: Noise vs. time of sensor 1 with DCBL motor a 25 % throttle.

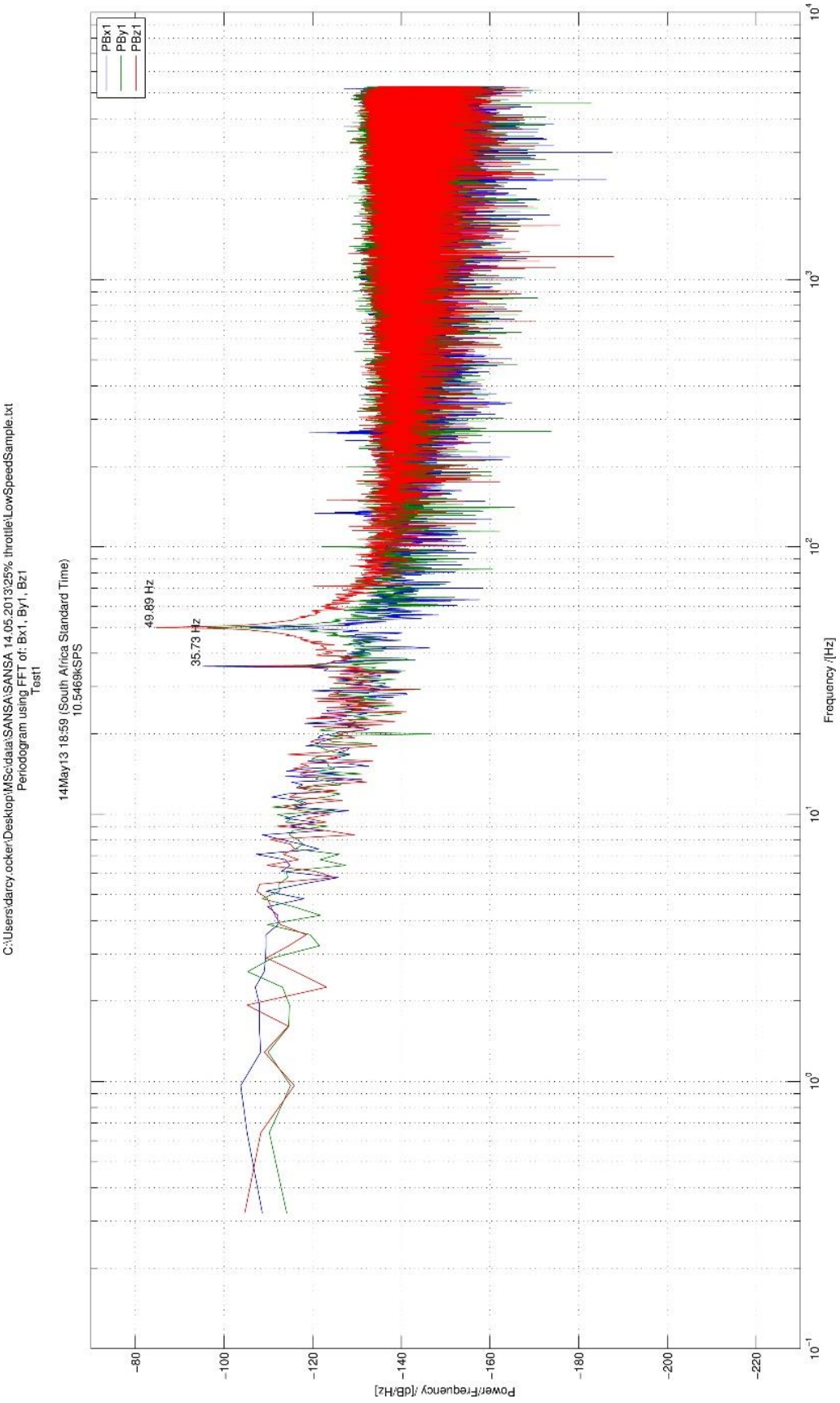


Figure I - 20: Spectral analysis of sensor 1 with DCBL motor at 25 % throttle.

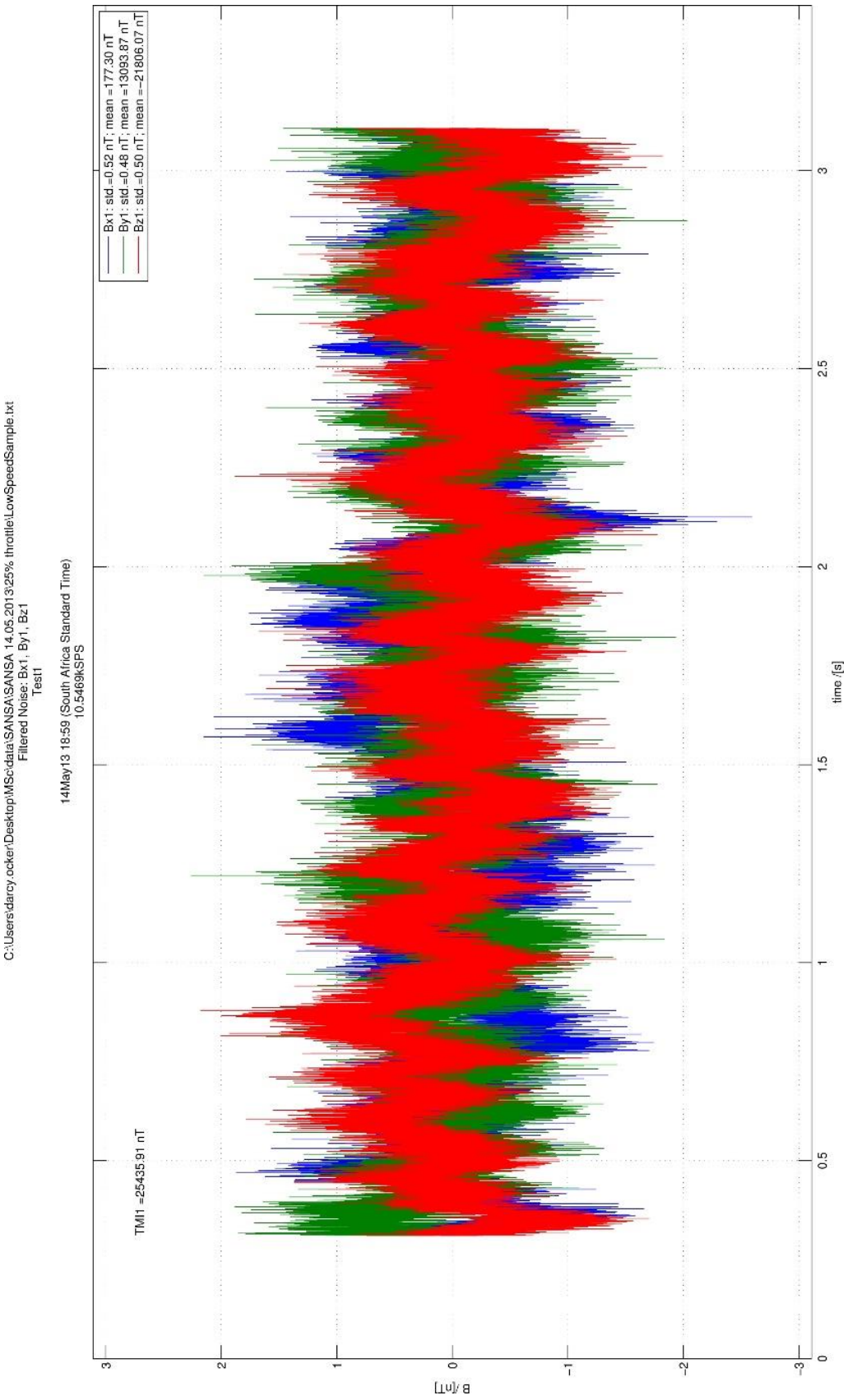


Figure I - 21: Noise vs. time (notch filters applied) of sensor 1 with DCBL motor at 25% throttle.

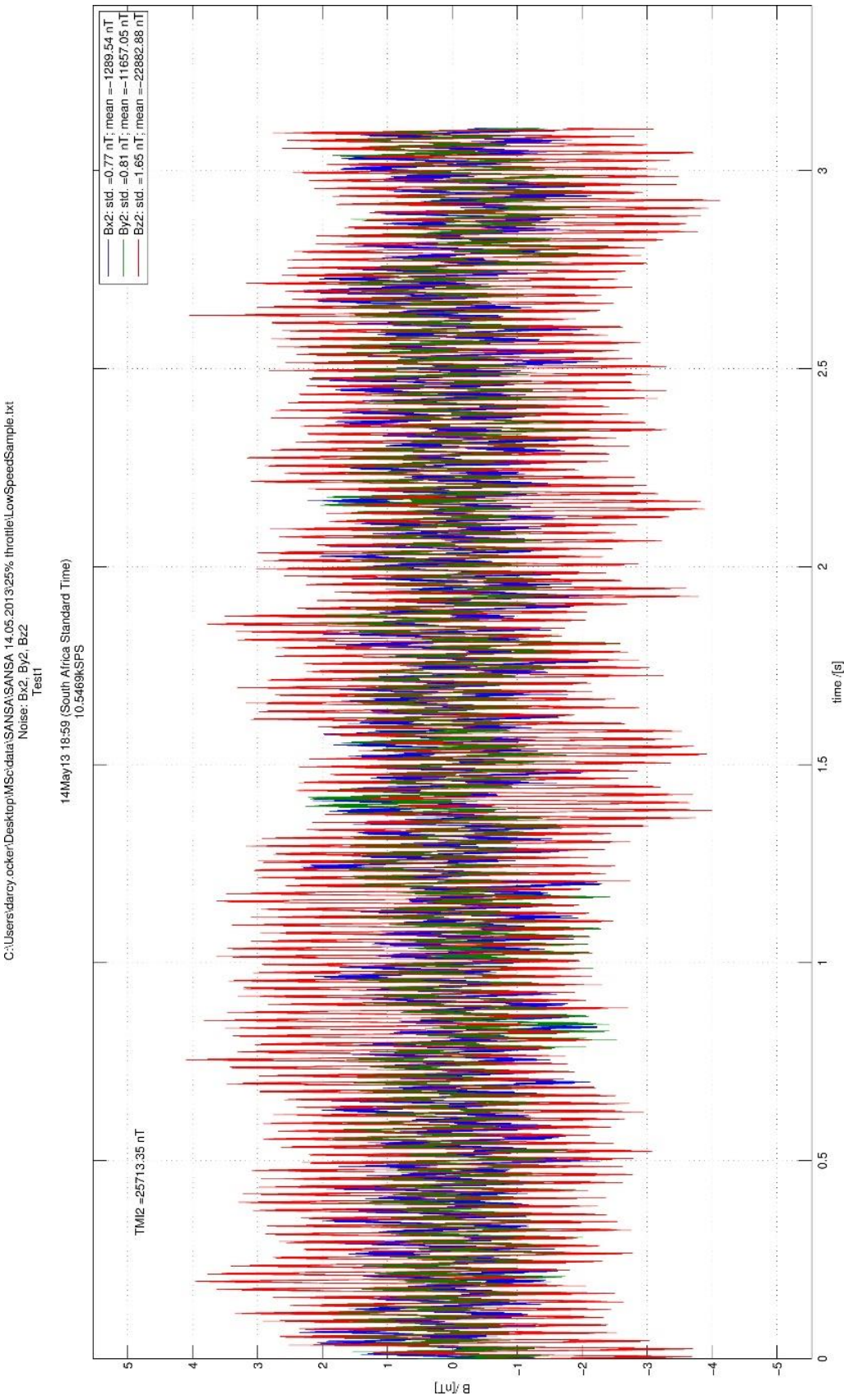


Figure I - 22: Noise vs. time of sensor 2 with DCBL motor at 25 % throttle.

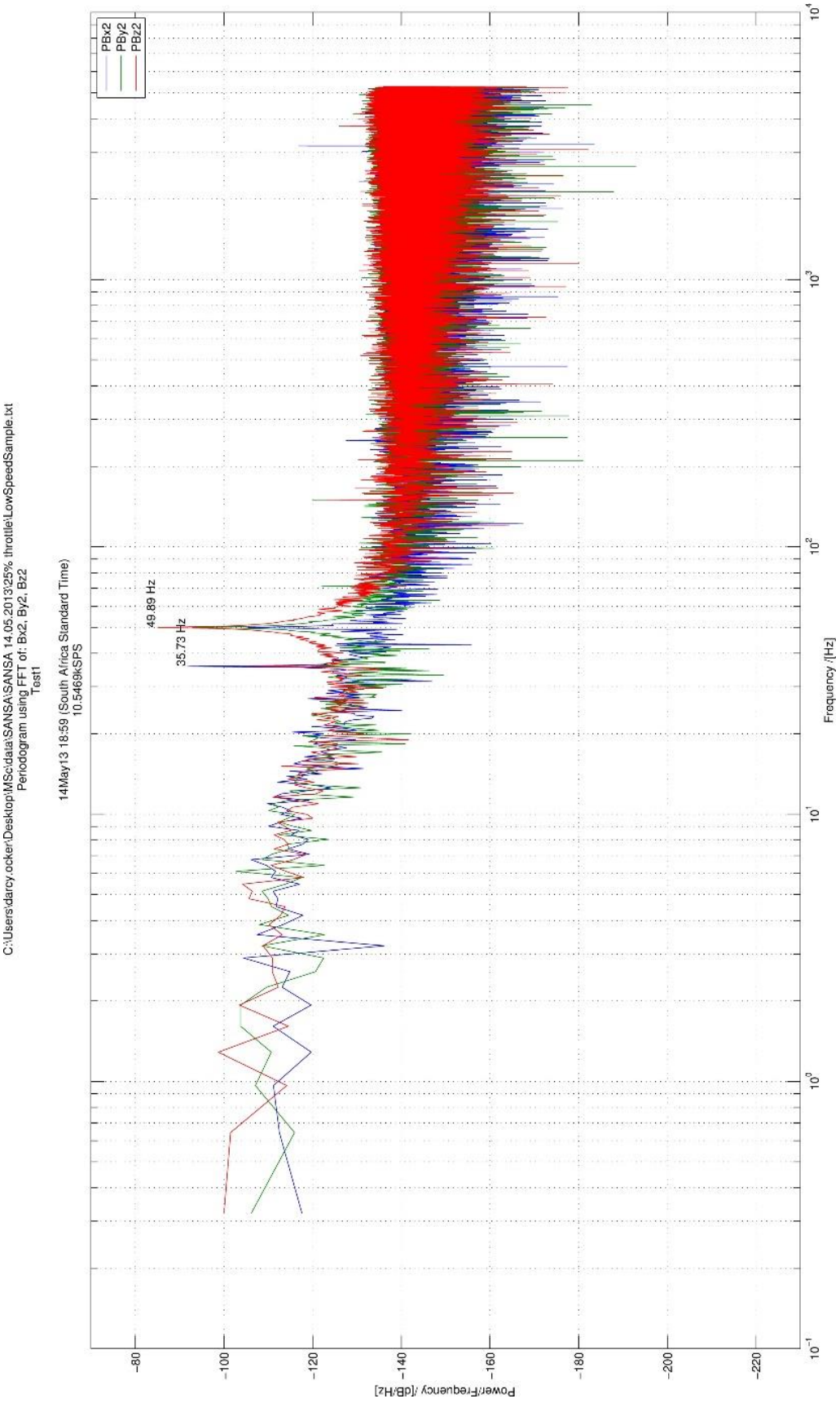


Figure I - 23: Spectral analysis of sensor 2 with DCBL motor at 25 % throttle.

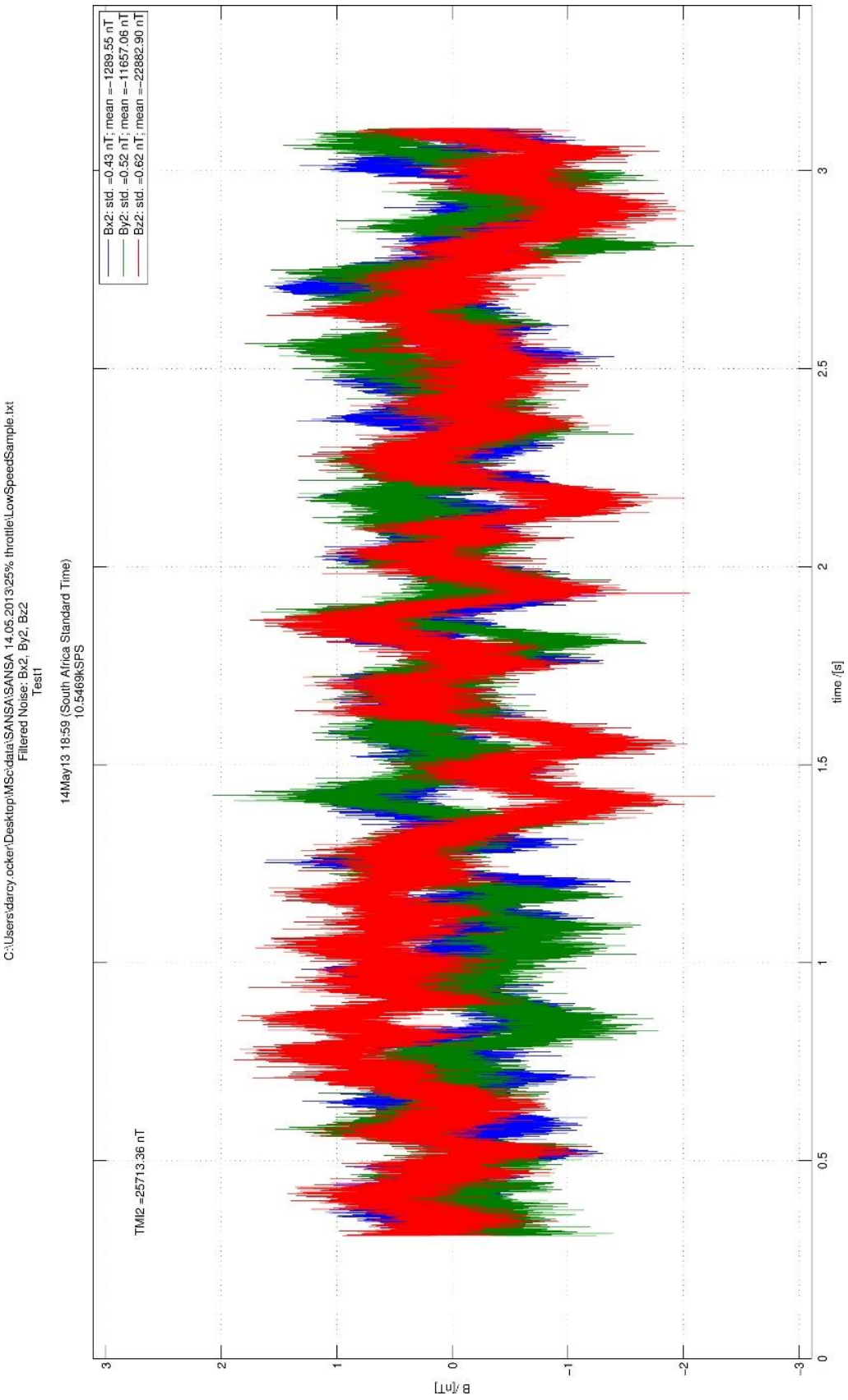


Figure I - 24: Noise vs. time (notch filters applied) of sensor 2 with DCBL motor at 25 % throttle.

1.2.3 Test Results: 50 % throttle

The throttle was set to approximately 50 % which allowed the motor to turn at 4287 RPM. The results are tabulated below. This shows two noticeable lines at 50 Hz (AC Mains) and 72 Hz (DCBL motor shaft speed). The current draw on the LiPo batteries at 50 % throttle opening was around 5 A.

With light throttle openings such as this the performance of the combination of the magnetometer and ADC can be similar to that of the LEMI-011b magnetometer specifications.

Table I - 5: Summary data for UAV platform test with DCBL motor at 4287 RPM

	Bx	***By***	***Bz***
Sensor 1			
Noise PTP before notch filter(s) applied	± 3.0 nT		
TMI before notch filters applied	25 434.90 nT		
Std. dev. before notch filter(s) applied	0.630 nT	0.740 nT	1.672 nT
Mean before notch filter(s) applied	178 nT	13105 nT	-21798 nT
Noise PTP after notch filter(s) applied	± 2.5 nT		
TMI after notch filters applied	25 434.92 nT		
Std. dev. after notch filter(s) applied	0.600 nT	0.632 nT	0.582 nT
Mean after notch filter(s) applied	178 nT	13106 nT	-21798 nT
Power @ 72 Hz (DCBL Shaft Speed)	-107 dB/Hz	-103 dB/Hz	-110 dB/Hz
Power @ 50 Hz (AC Mains)	-110 dB/Hz	N/A	-93 dB/Hz
Sensor 2			
Noise PTP before notch filter(s) applied	± 3.0 nT		
TMI before notch filters applied	25 708.69 nT		
Std. dev. before notch filter(s) applied	0.622 nT	0.694 nT	1.023 nT
Mean before notch filter(s) applied	-1294 nT	-11669 nT	-22871 nT
Noise PTP after notch filter(s) applied	± 2.0 nT		
TMI after notch filters applied	25 708.68 nT		
Std. dev. after notch filter(s) applied	0.439 nT	0.538 nT	0.638 nT
Mean after notch filter(s) applied	-1290 nT	-11657 nT	-22883 nT
Power @ 72 Hz (DCBL Shaft Speed)	-100 dB/Hz	-109 dB/Hz	N/A
Power @ 50 Hz (AC Mains)	N/A	-101 dB/Hz	-93 dB/Hz

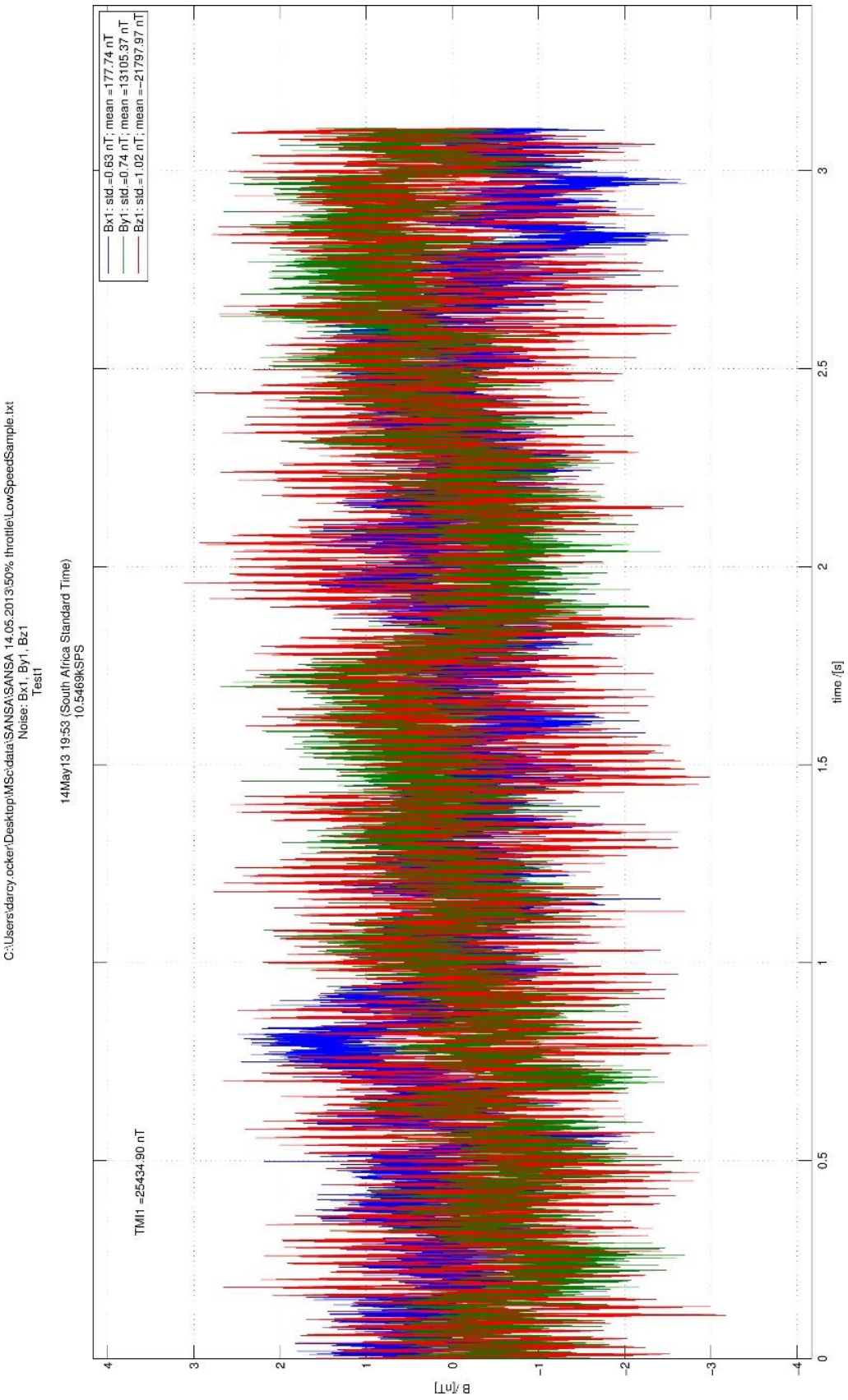


Figure I - 25: Noise vs. time of sensor 1 with DCBL motor at 50 % throttle.

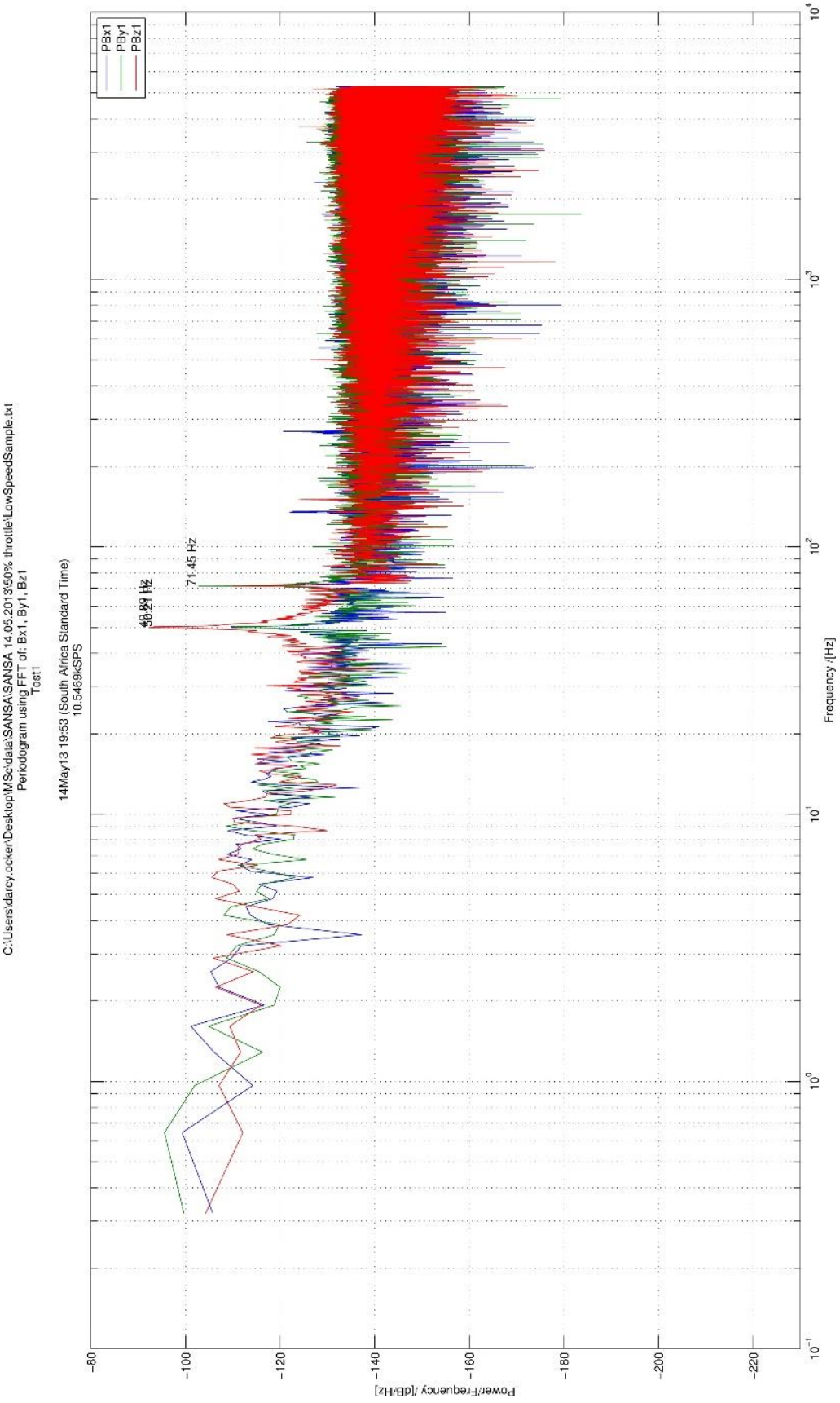


Figure I - 26: Spectral analysis of sensor 1 with DCBL motor at 50 % throttle.

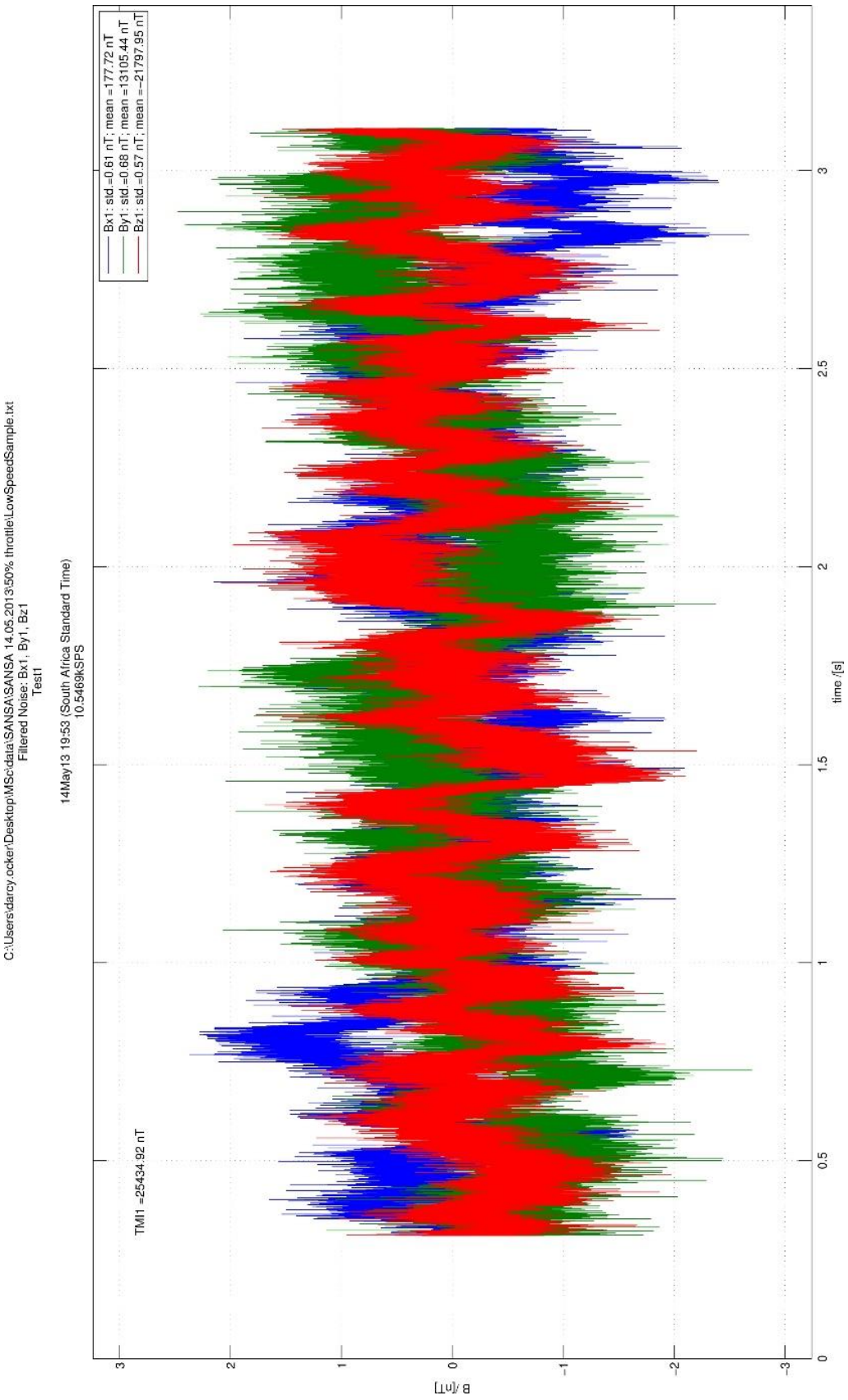


Figure I - 27: Noise vs. time (notch filters applied) of sensor 1 with DCBL motor at 50% throttle.

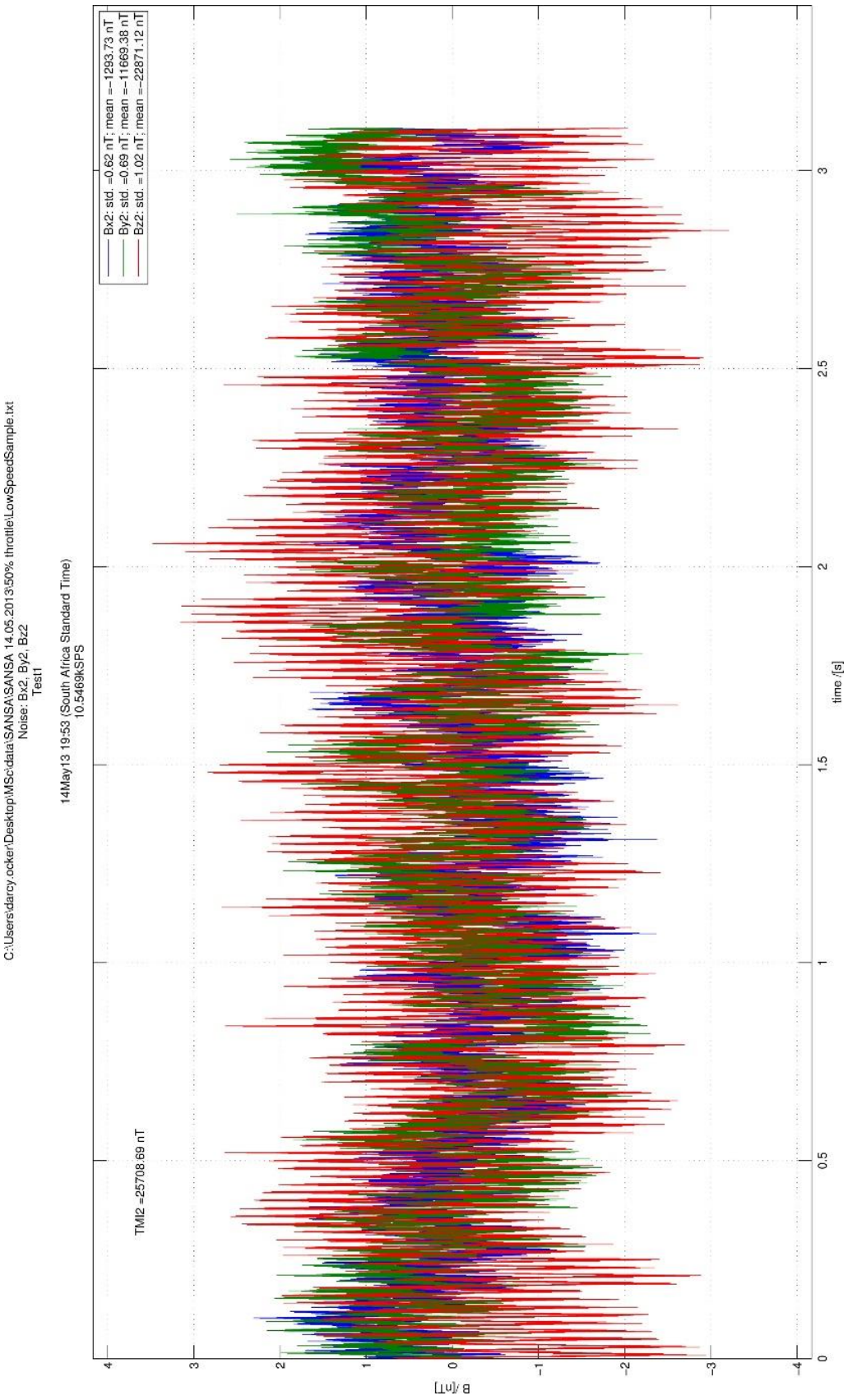


Figure I - 28: Noise vs. time of sensor 2 with DCBL motor at 50 % throttle.

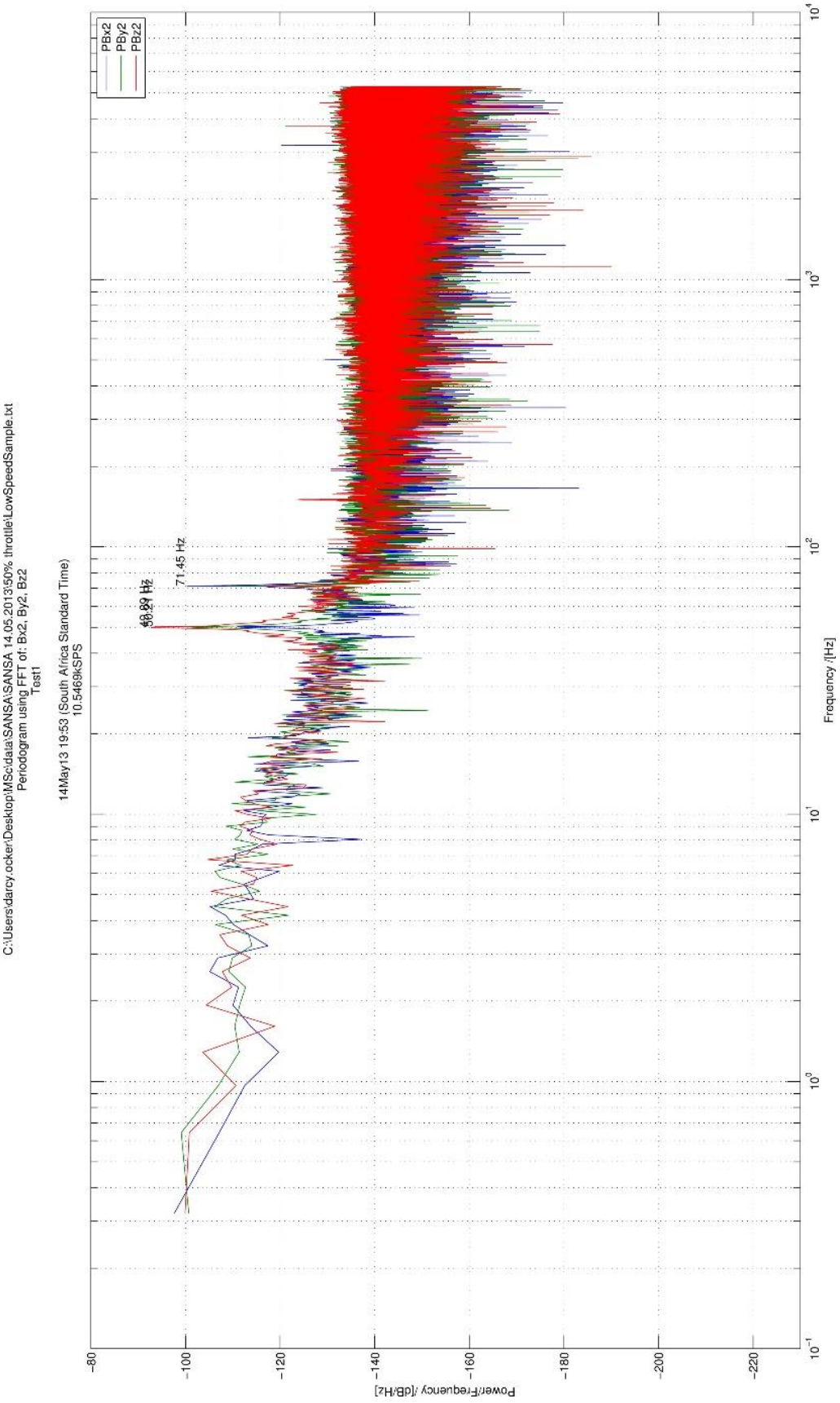


Figure I - 29: Spectral analysis of sensor 2 with DCBL motor at 50 % throttle.

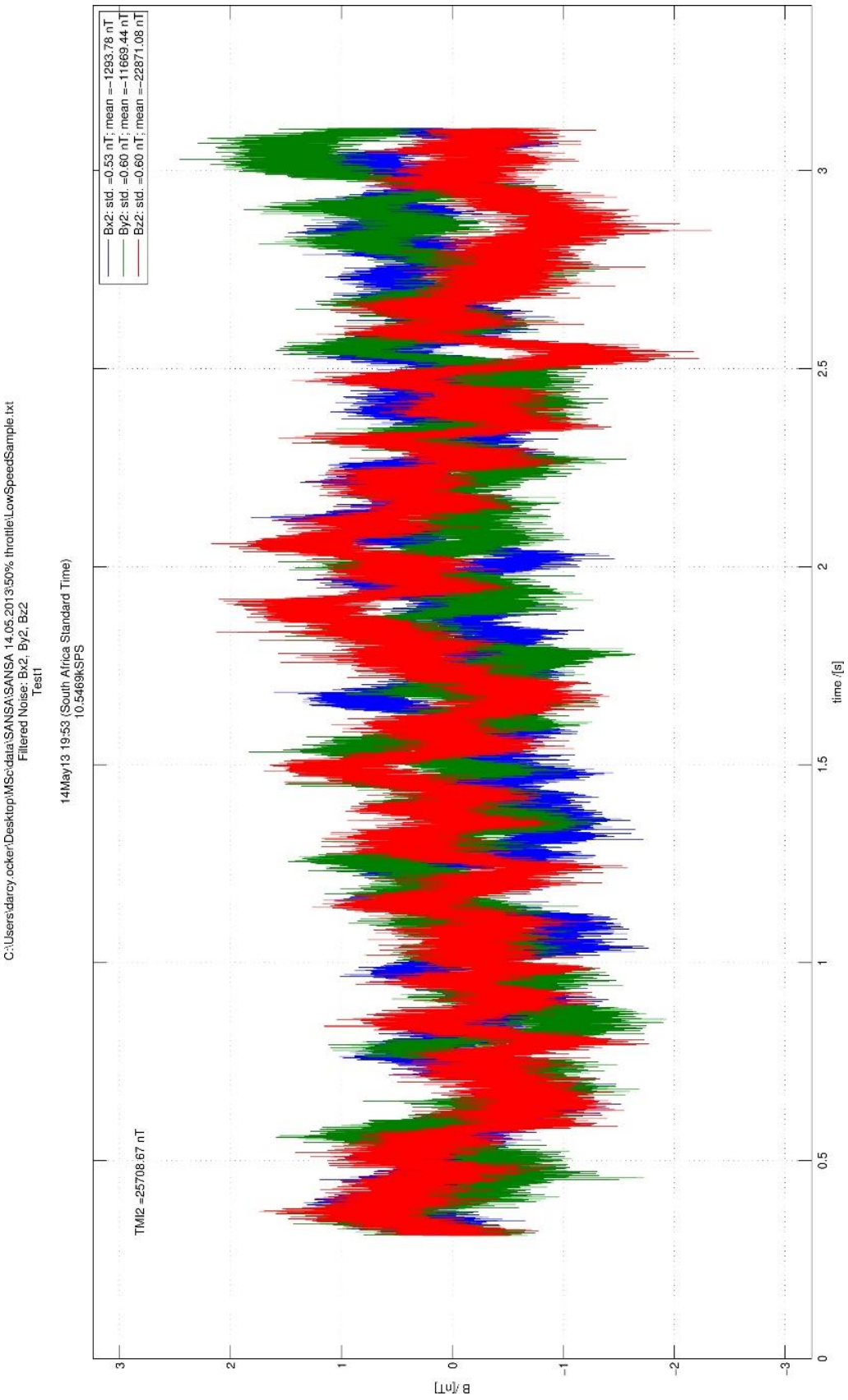


Figure I - 30: Noise vs. time (notch filters applied) of sensor 2 with DCBL motor at 50 % throttle.

1.2.4 Test Results: 75 % throttle

The throttle was set to approximately 75 % which allowed the motor to turn at 6412 RPM. The results are tabulated below. This shows two noticeable lines at 50 Hz (AC Mains) and 108 Hz (DCBL motor shaft speed). The current draw on the LiPo batteries at 75 % throttle opening was around 17 A.

One should note that sensor 2 has a strange noise profile. This can be attributed to wind gust generated by the propeller off the back wall of the test facility causing the starboard side of the UAV wing to flap. This is due to the wing tips not being securely fastened during the tests.

Similar behaviour is more noticeable in the 100% throttle tests below. Note how the TMI values are maintained when compared to low throttle openings.

Table I - 6: Summary data for UAV platform test with DCBL motor at 6412 RPM

	Bx	***By***	***Bz***
Sensor 1			
Noise PTP before notch filter(s) applied	± 3.5 nT		
TMI before notch filters applied	25 430.85 nT		
Std. dev. before notch filter(s) applied	1.026 nT	0.870 nT	1.230 nT
Mean before notch filter(s) applied	176 nT	13107 nT	-21792 nT
Noise PTP after notch filter(s) applied	± 2.5 nT		
TMI after notch filters applied	25 430.84 nT		
Std. dev. after notch filter(s) applied	0.689 nT	0.608 nT	0.616 nT
Mean after notch filter(s) applied	176 nT	13107 nT	-21792 nT
Power @ 108 Hz (DCBL Shaft Speed)	N/A	N/A	N/A
Power @ 50 Hz (AC Mains)	N/A	-102 dB/Hz	-90 dB/Hz
Sensor 2			
Noise PTP before notch filter(s) applied	± 4.0 nT (see explanation)		
TMI before notch filters applied	25 710.32 nT		
Std. dev. before notch filter(s) applied	1.391 nT	1.547 nT	1.378 nT
Mean before notch filter(s) applied	-1300 nT	-11674 nT	-22870 nT
Noise PTP after notch filter(s) applied	± 2.0 nT (see explanation)		
TMI after notch filters applied	25 710.32 nT		
Std. dev. after notch filter(s) applied	1.314 nT	1.366 nT	0.860 nT
Mean after notch filter(s) applied	-1300 nT	-11674 nT	-22870 nT
Power @ 108 Hz (DCBL Shaft Speed)	N/A	-99 dB/Hz	N/A
Power @ 50 Hz (AC Mains)	N/A	-98 dB/Hz	-90 dB/Hz

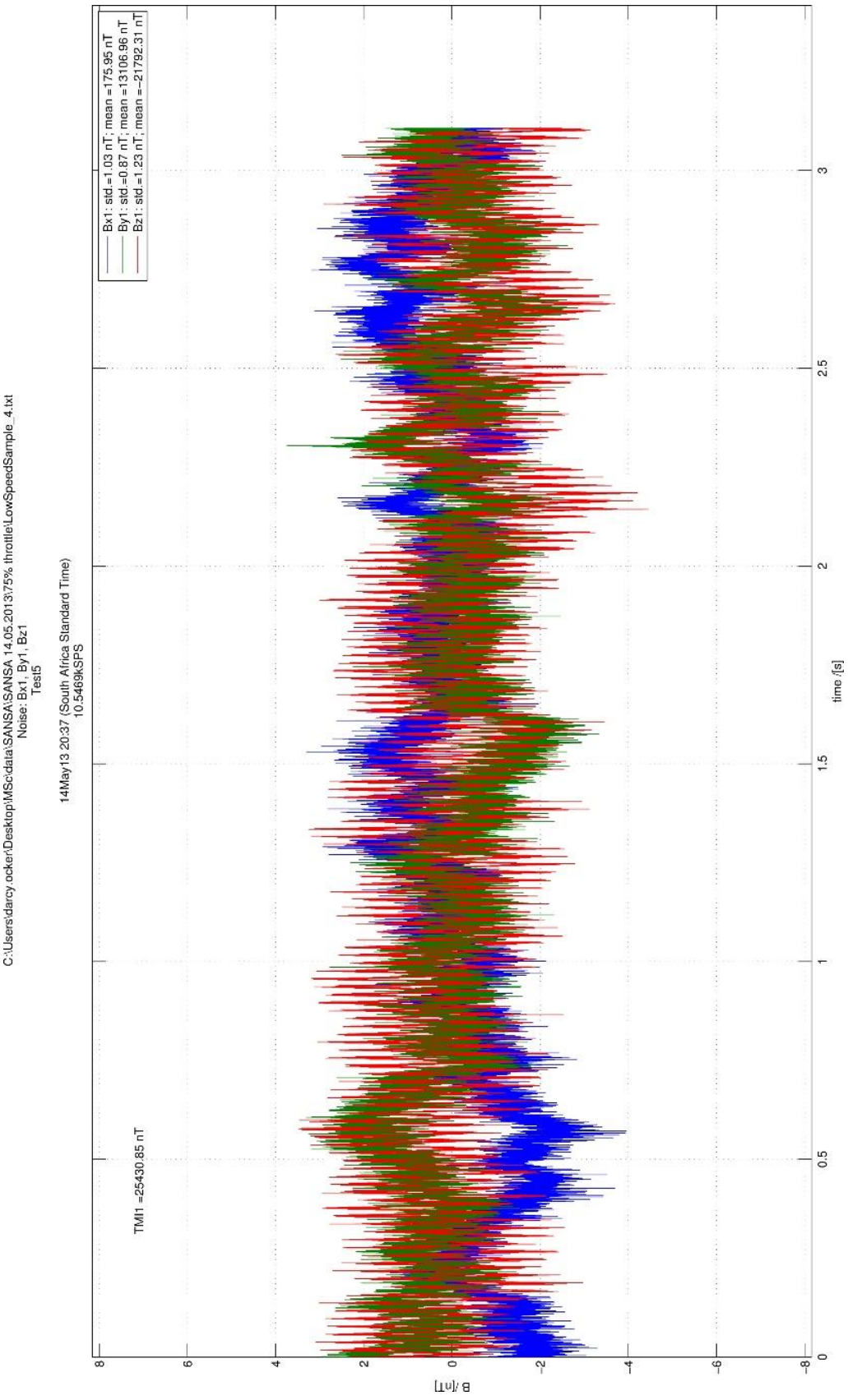


Figure I - 31: Noise vs. time of sensor 1 with DCBL motor at 75 % throttle.

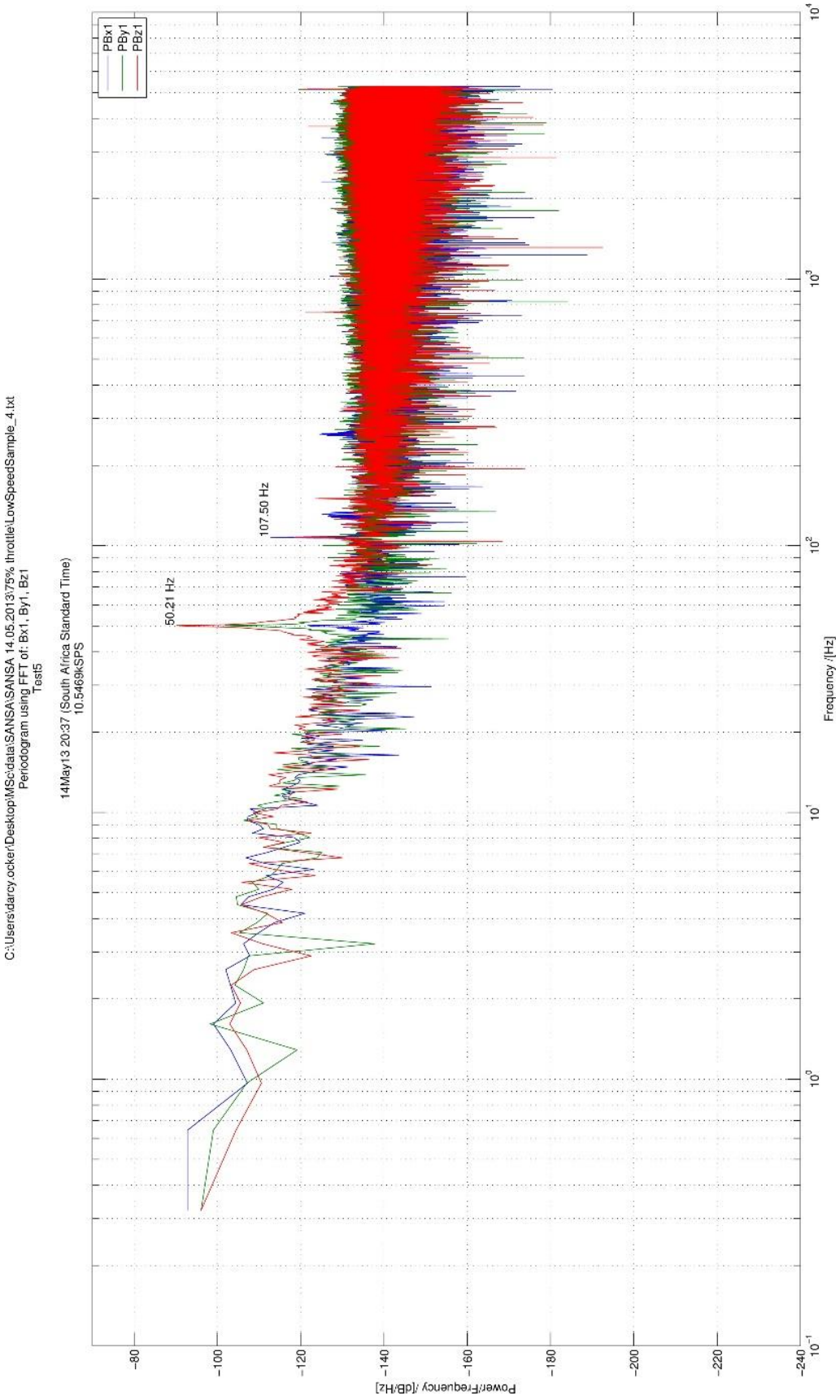


Figure I - 32: Spectral analysis of sensor 1 with DCBL motor at 75 % throttle.

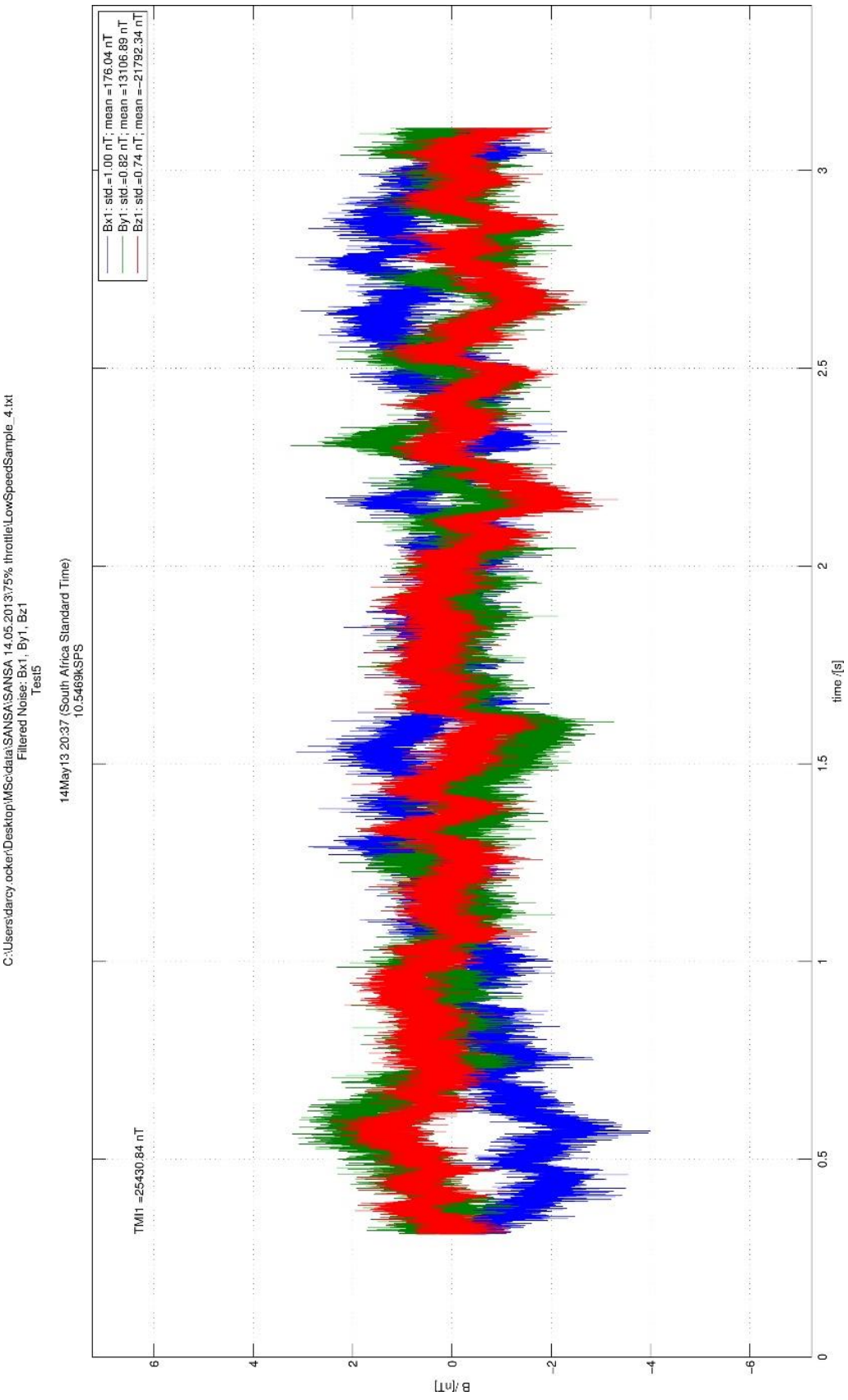


Figure I - 33: Noise vs. time (notch filters applied) of sensor 1 with DCBL motor at 75 % throttle.

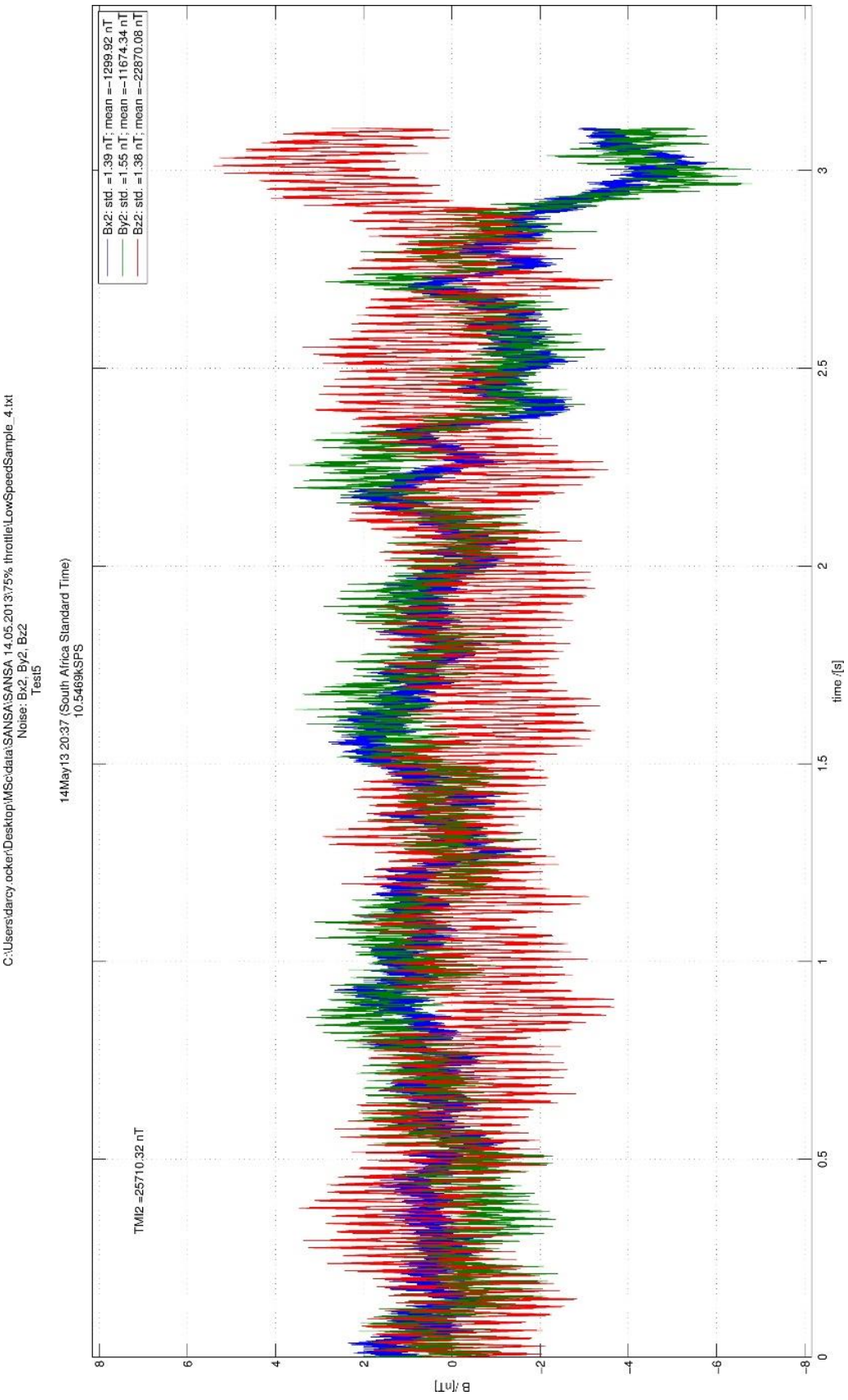


Figure I - 34: Noise vs. time of sensor 2 with DCBL motor at 75 % throttle.

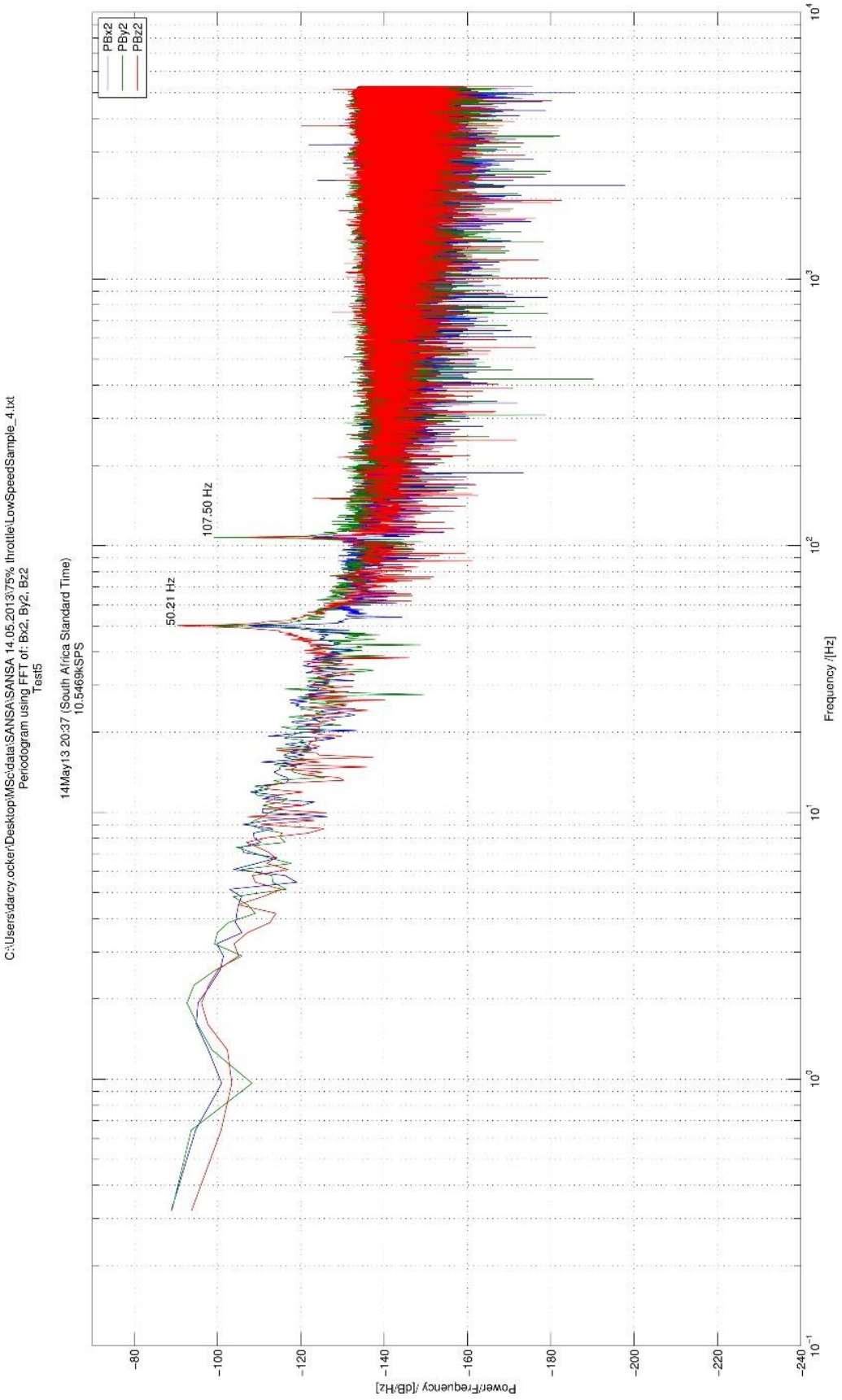


Figure I - 35: Spectral analysis of sensor 2 with DCBL motor at 75 % throttle.

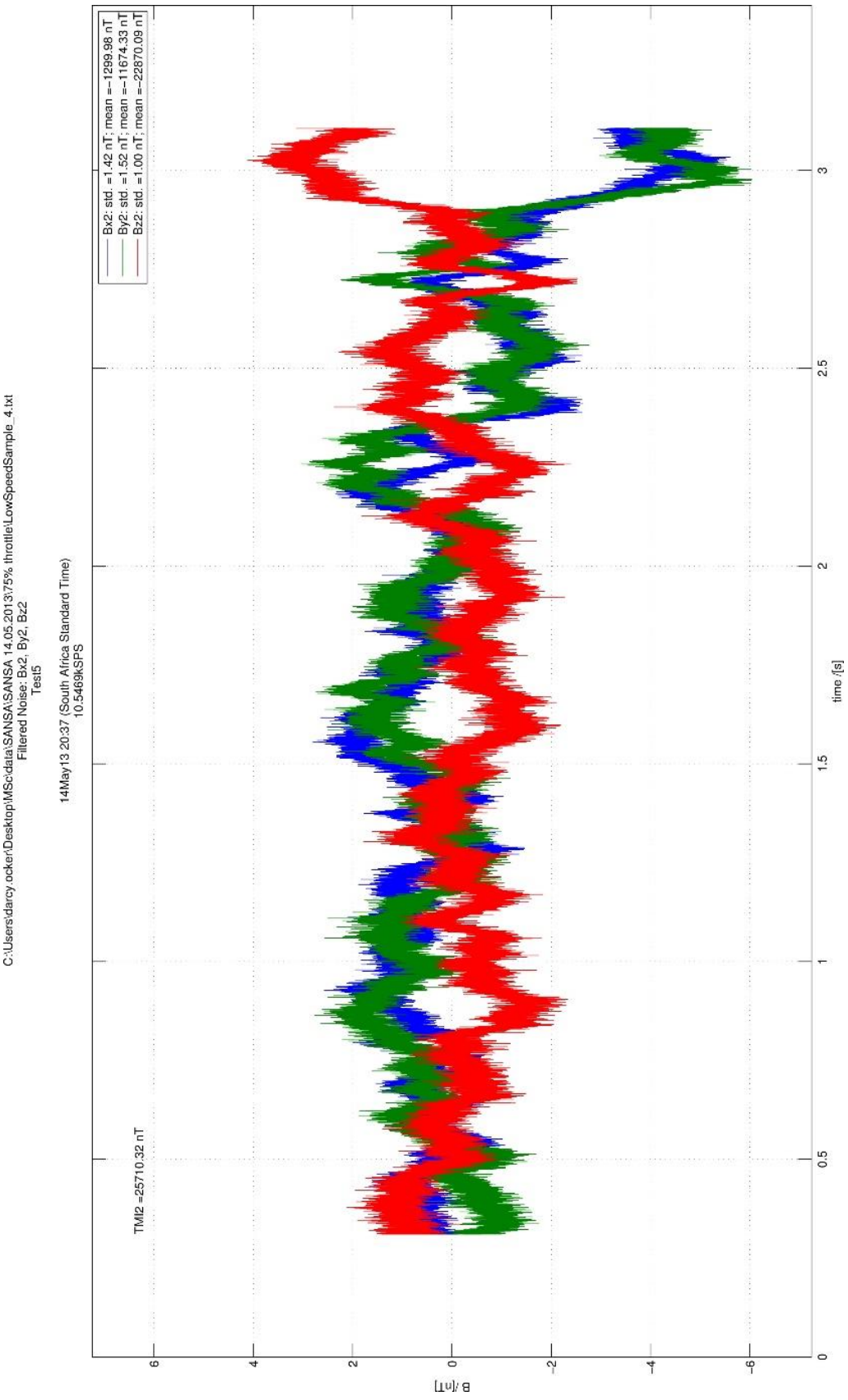


Figure I - 36: Noise vs. time (notch filters applied) of sensor 2 with DCBL motor at 75 % throttle.

1.2.5 Test Results: 100 % throttle

The throttle was set to approximately 100 % which allowed the motor to turn at 10714 RPM. The results are tabulated below. This shows a noticeable line at 50 Hz (AC Mains) while the expected 180 Hz (DCBL motor shaft speed) was not visible (see explanation in chapter 8.7 above). The current draw on the LiPo batteries at 100 % throttle opening was around 65 A.

One should note that as the frequency line attributed to the DCBL motor moved up the frequency axis, the damping effect of the built-in 10 Hz low pass filter of the LEMI-011b becomes more effective. Hence, the reason for not seeing the expected line at 180 Hz for the 100 % throttle tests.

Table I - 7: Summary data for UAV platform test with DCBL motor at 10714 RPM

	Bx	***By***	***Bz***
Sensor 1			
Noise PTP before notch filter(s) applied	± 12.0 nT (see explanation)		
TMI before notch filters applied	25 430.69 nT		
Std. dev. before notch filter(s) applied	4.058 nT	3.829 nT	2.352 nT
Mean before notch filter(s) applied	179 nT	13079 nT	-21809 nT
Noise PTP after notch filter(s) applied	± 10.0 nT (see explanation)		
TMI after notch filters applied	25 430.66 nT		
Std. dev. after notch filter(s) applied	3.908 nT	3.301 nT	1.749 nT
Mean after notch filter(s) applied	179 nT	13080 nT	-21808 nT
Power @ 180 Hz (DCBL Shaft Speed)	N/A	N/A	N/A
Power @ 50 Hz (AC Mains)	N/A	N/A	-89 dB/Hz
Sensor 2			
Noise PTP before notch filter(s) applied	± 15.0 nT (see explanation)		
TMI before notch filters applied	25 716.60 nT		
Std. dev. before notch filter(s) applied	6.981 nT	6.961 nT	3.820 nT
Mean before notch filter(s) applied	-1318 nT	-11649 nT	-22889 nT
Noise PTP after notch filter(s) applied	± 10.0 nT (see explanation)		
TMI after notch filters applied	25 716.57 nT		
Std. dev. after notch filter(s) applied	6.626 nT	6.759 nT	3.560 nT
Mean after notch filter(s) applied	-1317 nT	-11649 nT	-22889 nT
Power @ 180 Hz (DCBL Shaft Speed)	N/A	N/A	N/A
Power @ 50 Hz (AC Mains)	N/A	N/A	-89 dB/Hz

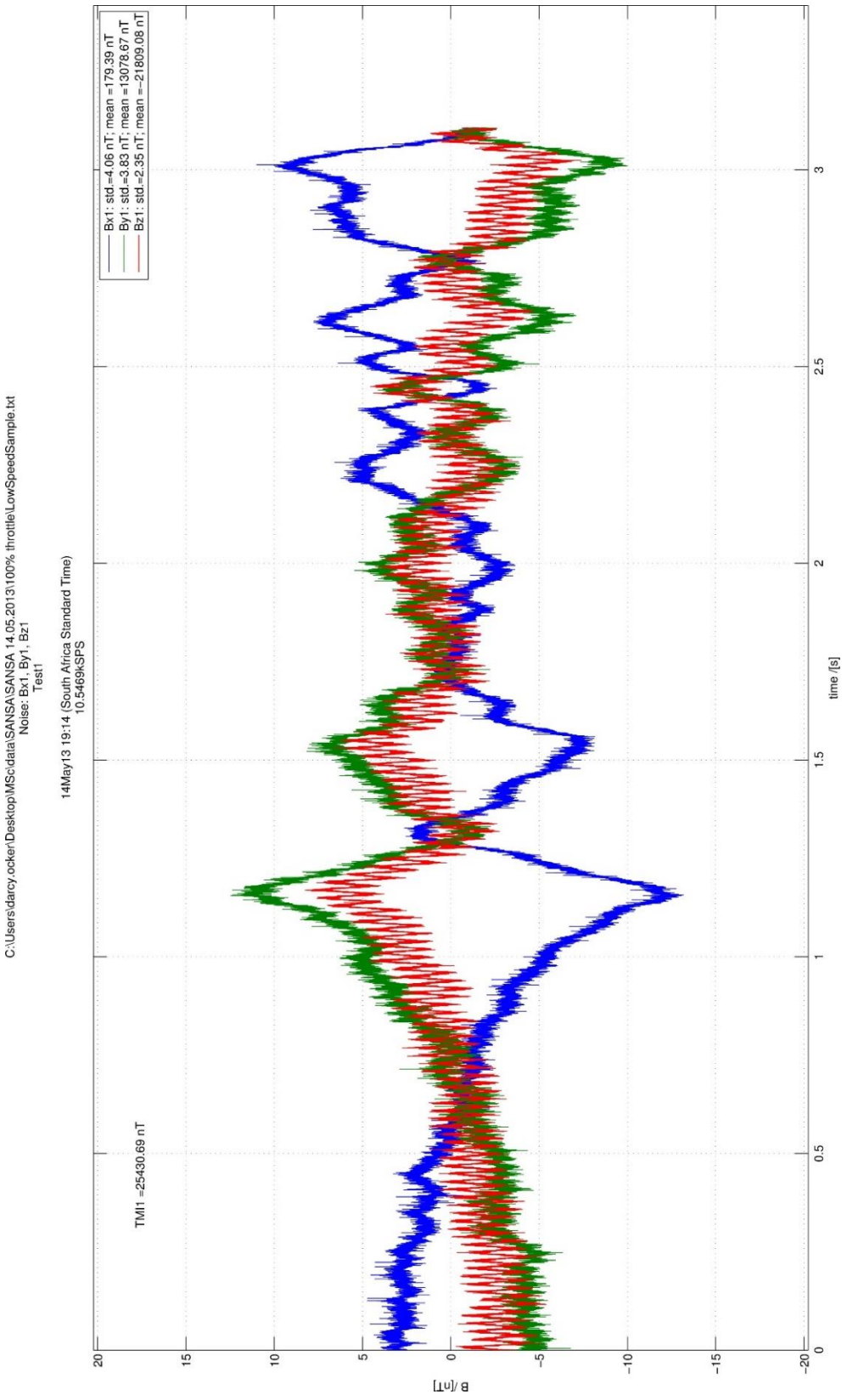


Figure I - 37: Noise vs. time of sensor 1 with DCBL motor at 100 % throttle. Note the 'slew' in the data due to the wingtip flapping movements due to turbulent airflow over the UAV during this test.

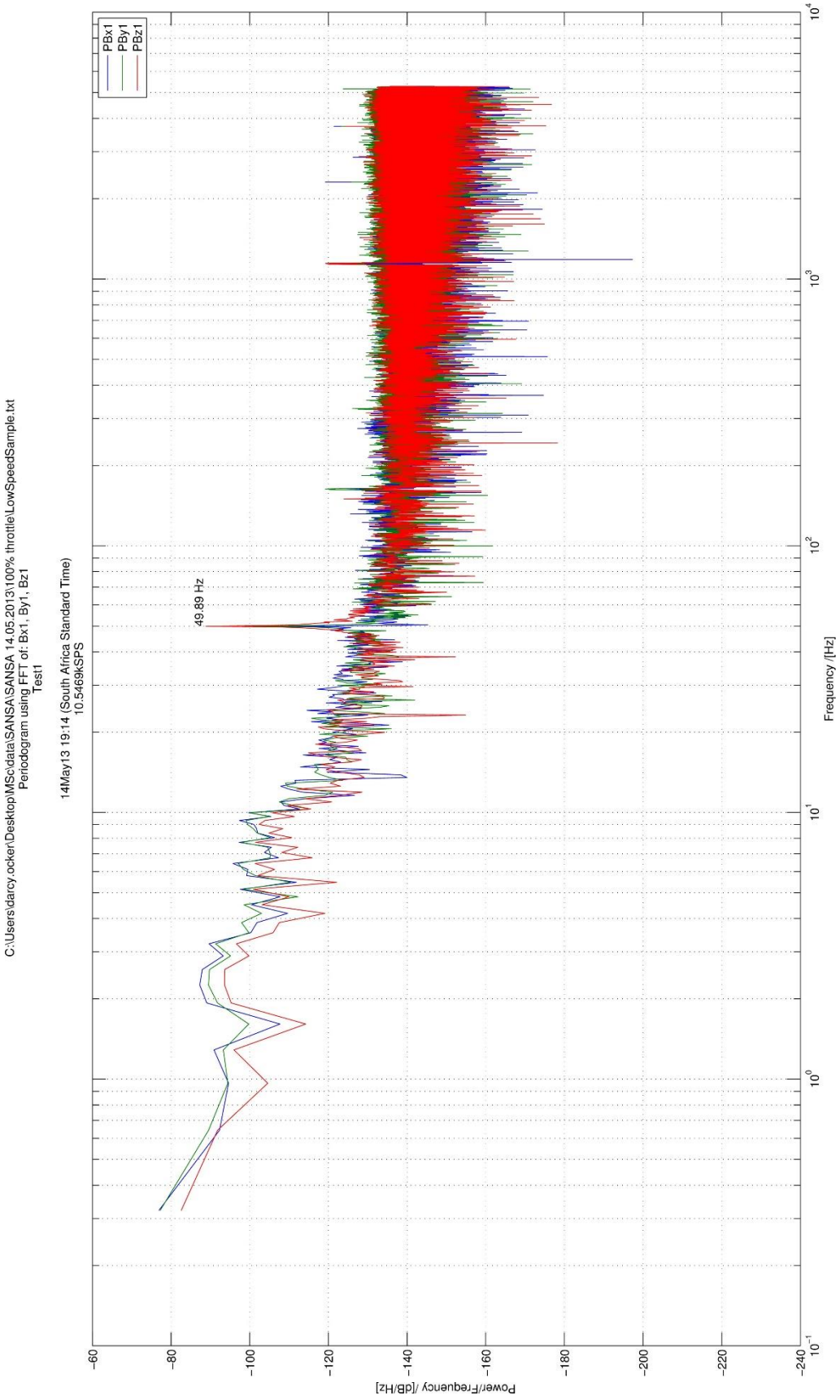


Figure I - 38: Spectral analysis of sensor 1 with DCBL motor at 100 % throttle.

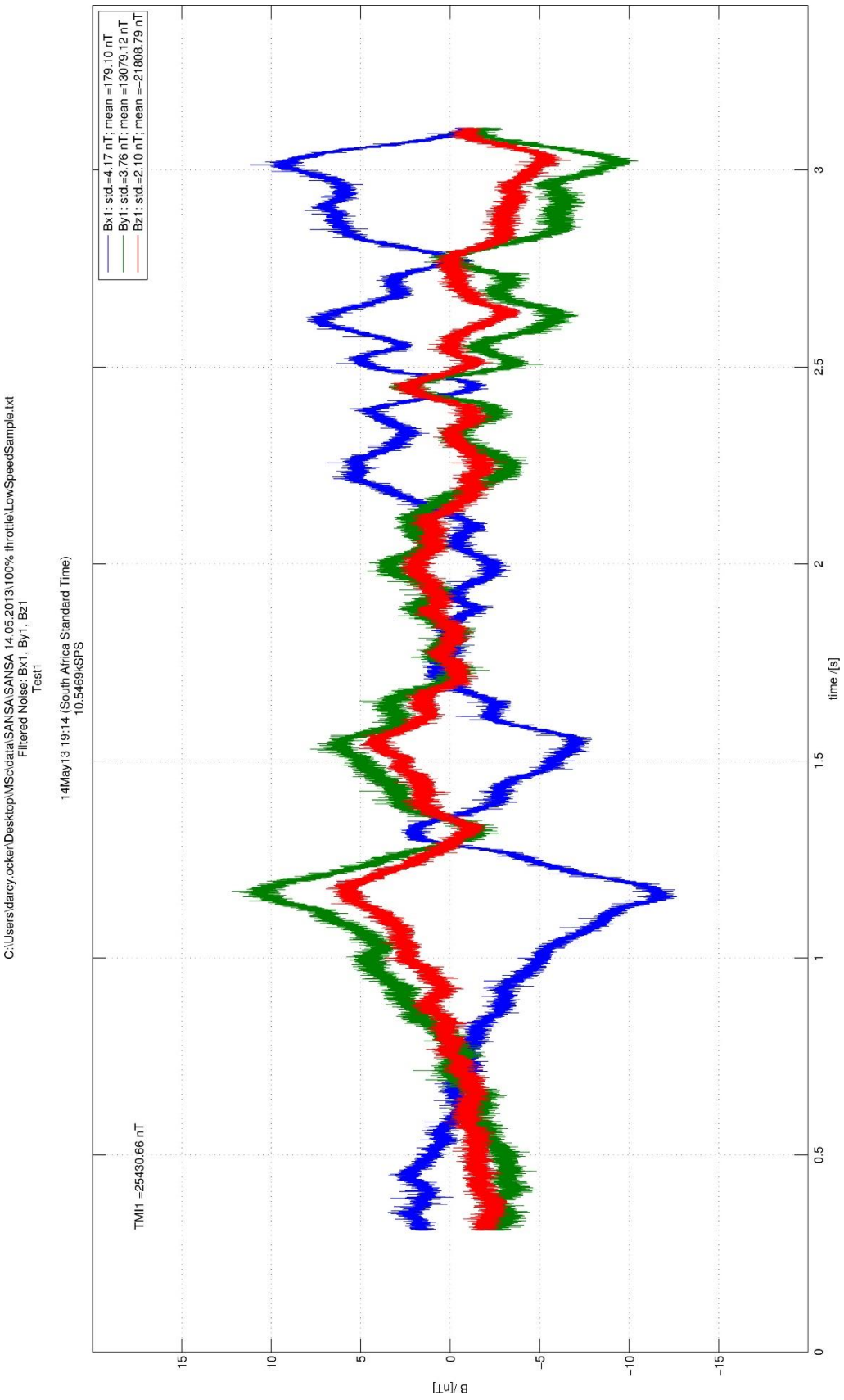


Figure I - 39: Noise vs. time (notch filters applied) of sensor 1 with DCBL motor at 100 % throttle.

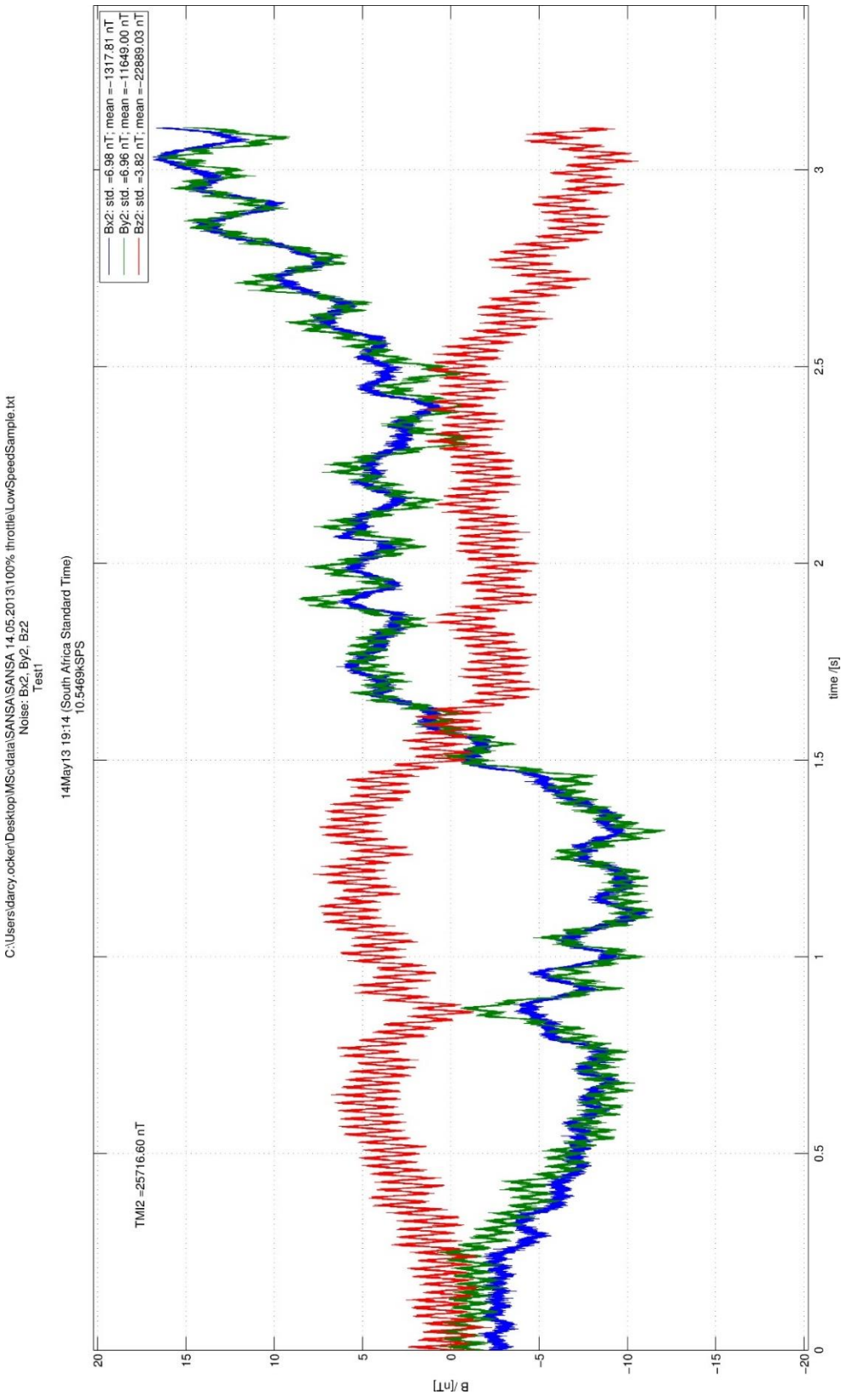


Figure I - 40: Noise vs. time of sensor 2 with DCBL motor at 100 % throttle. Note the 'slew' in the data due to the wingtip flapping movements due to turbulent airflow over the UAV during this test.

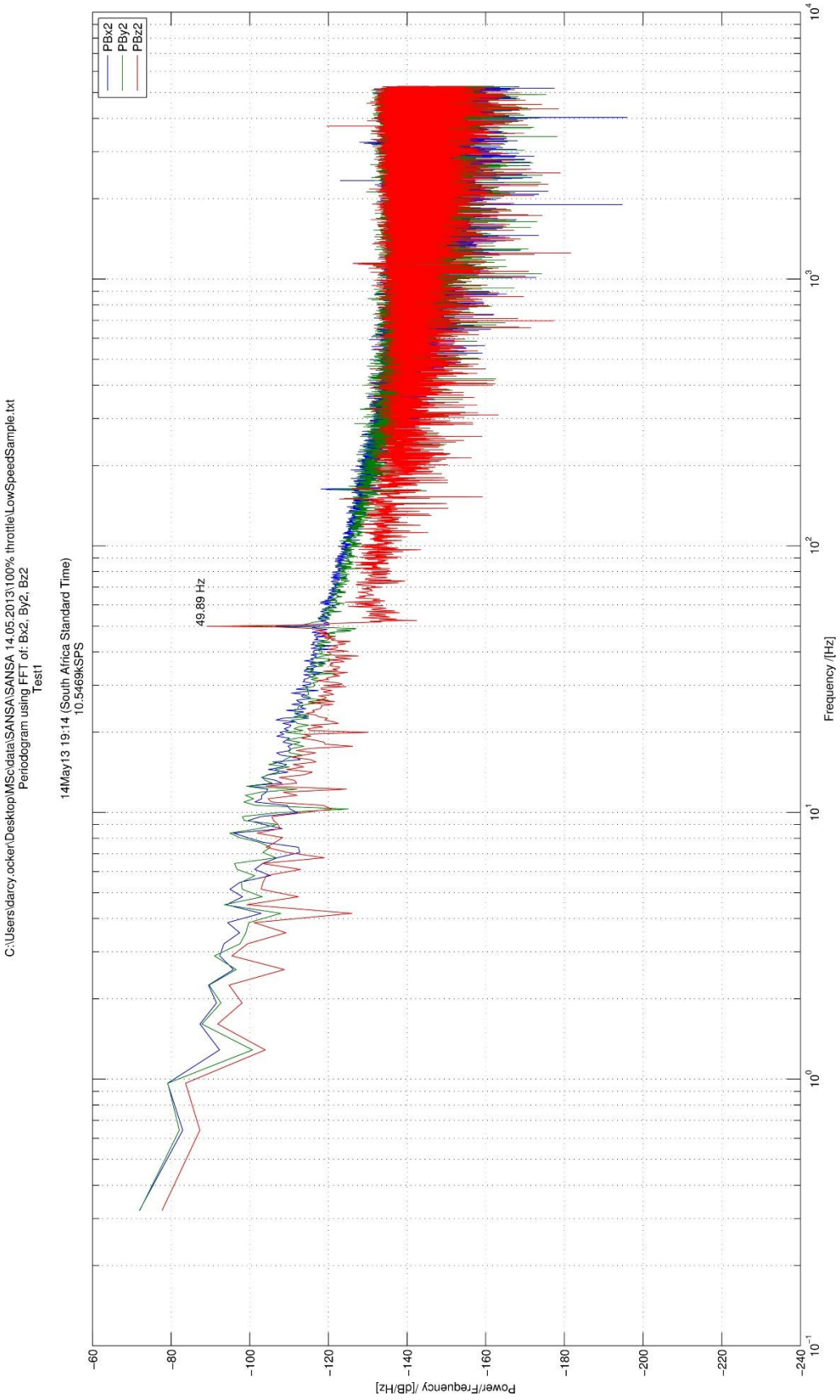


Figure I - 41: Spectral analysis of sensor 2 with DCBL motor at 100 % throttle.

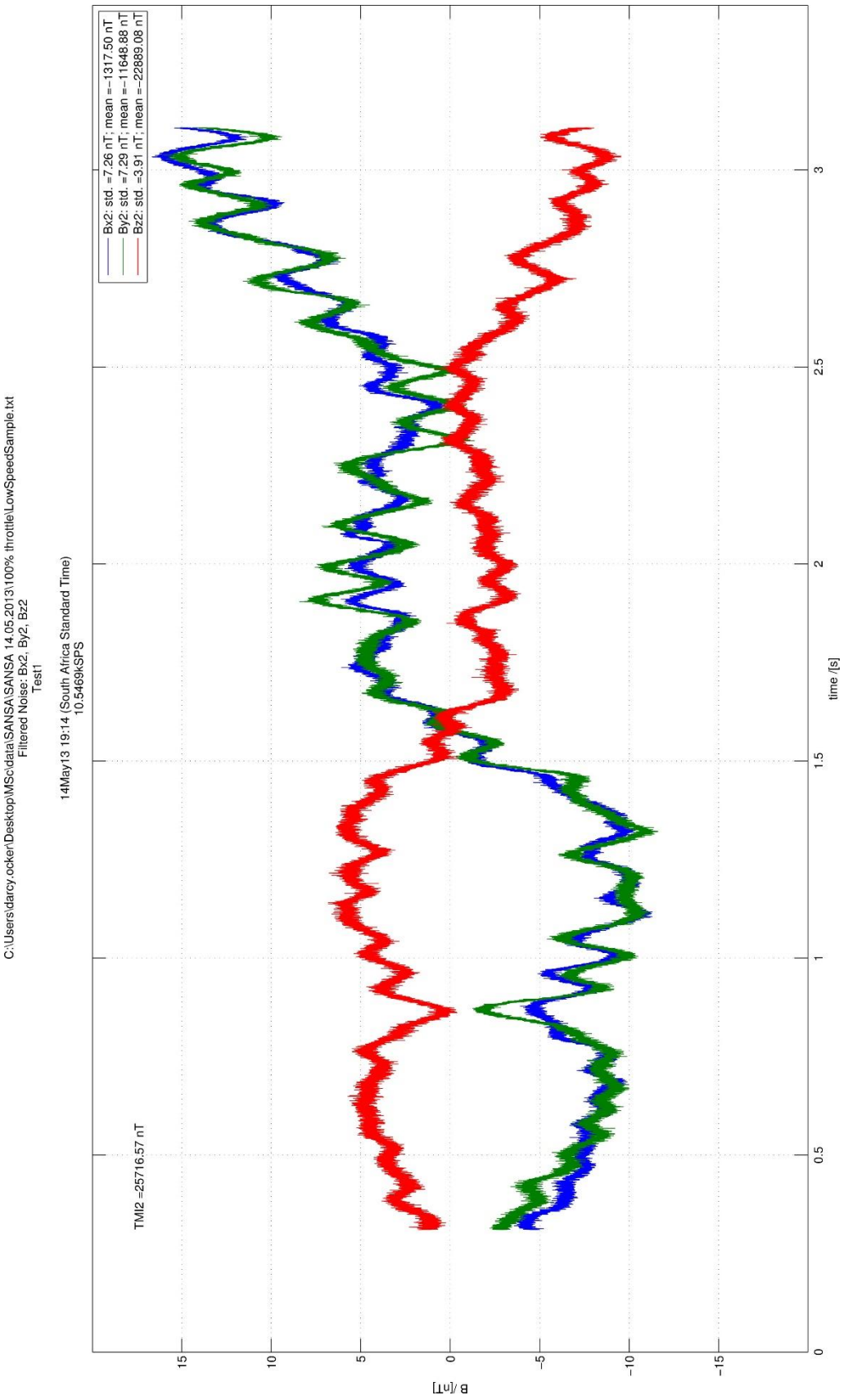


Figure I - 42: Noise vs. time (notch filters applied) of sensor 2 with DCBL motor at 100 % throttle.

Appendix J – Magnetised Metal Pole Characterisation

The characterisation of the magnetic dipole due to the magnetisation process of the 3 m metal pole is shown below. Note the magnetometer placed 73 cm above the magnetised pole and digitisation equipment for the magnetometer output in Figure 9-5 above. The output was recorded for five points along the metal pole. The distances for which measurements were taken along the magnetised pole length are:

1. 0 mm
2. 760 mm
3. 1500 mm
4. 2260 mm
5. 3000 mm

Individual plots are shown in Figure J - 1 through Figure J - 4 in Appendix J. The plots show the normalised (geomagnetic field removed) data of the magnetised pole. Note the classic (asymmetrical) dipole shape along some of the magnetic axes as well as the TMI plot.

Figure J - 1: The B_x measurement results for the magnetised metal pole at 5 points recorded 73 cm above the pole. . Please note the vertical axis displays the values in nT.

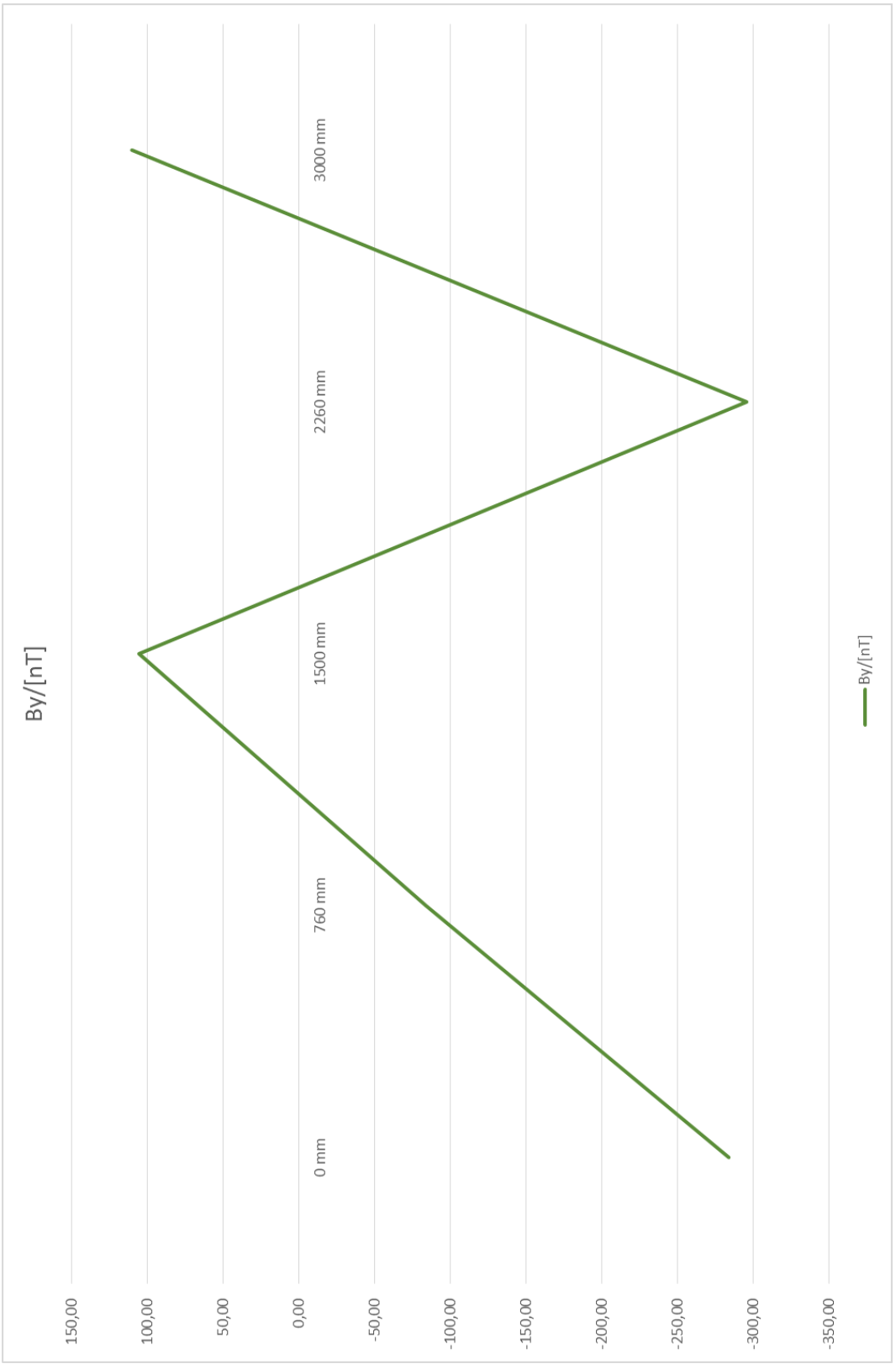


Figure J - 2: The B_y measurement results for the magnetised metal pole at 5 points recorded 73 cm above the pole. . Please note the vertical axis displays the values in nT.

Figure J - 3: The B_z measurement results for the magnetised metal pole at 5 points recorded 73 cm above the pole. . Please note the vertical axis displays the values in nT.

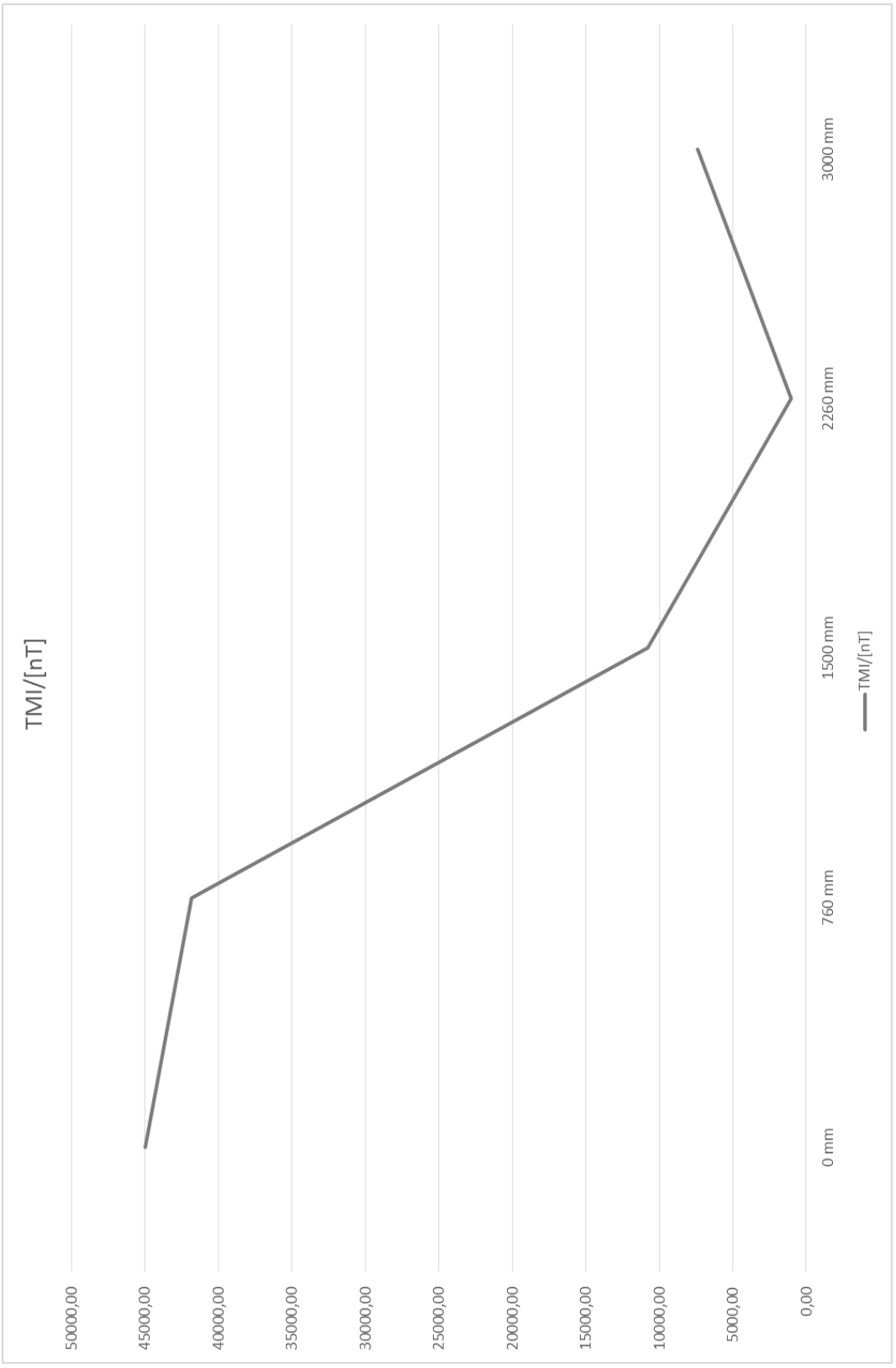


Figure J - 4: The TMI measurement results for the magnetised metal pole at 5 points recorded 73 cm above the pole. Note the characteristic dipole shape. . Please note the vertical axis displays the values in nT.

Appendix K – Magnetic Dipole Detection Results

The figures for the magnetic dipole detection results for the two experiments are presented in full below.

Detection results for the 600 mm above the ground experiment: Figure K - 1 to Figure K - 6.

Detection results for the 600 mm below (on the ground) experiment: Figure K - 7 to Figure K - 12.

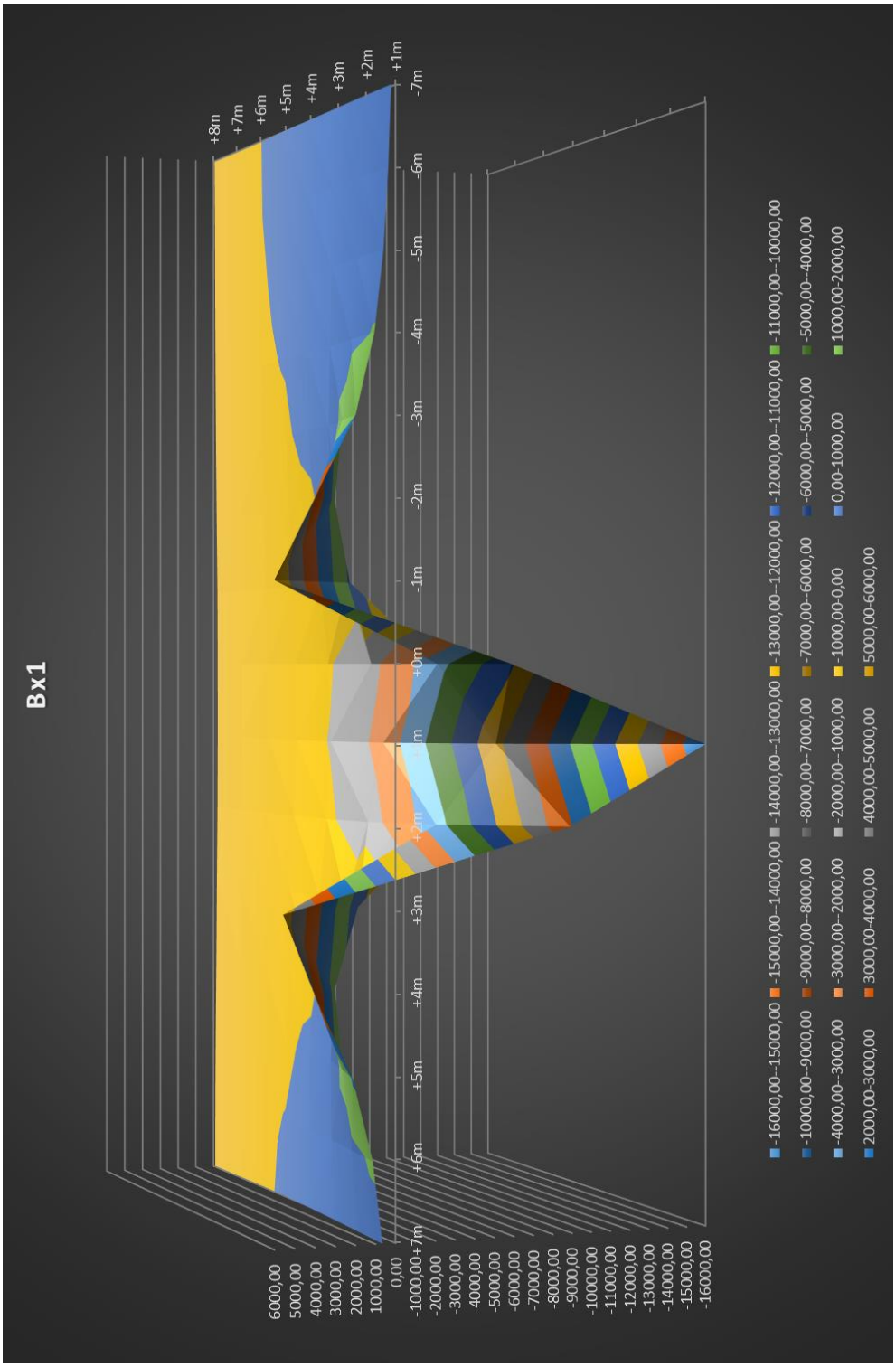


Figure K - 1: B_x data for sensor 1 with the 3 m magnetised pole held at the same height (600 mm above the ground) as the LEMI-011b magnetometers. . Please note the vertical axis displays the values in nT.

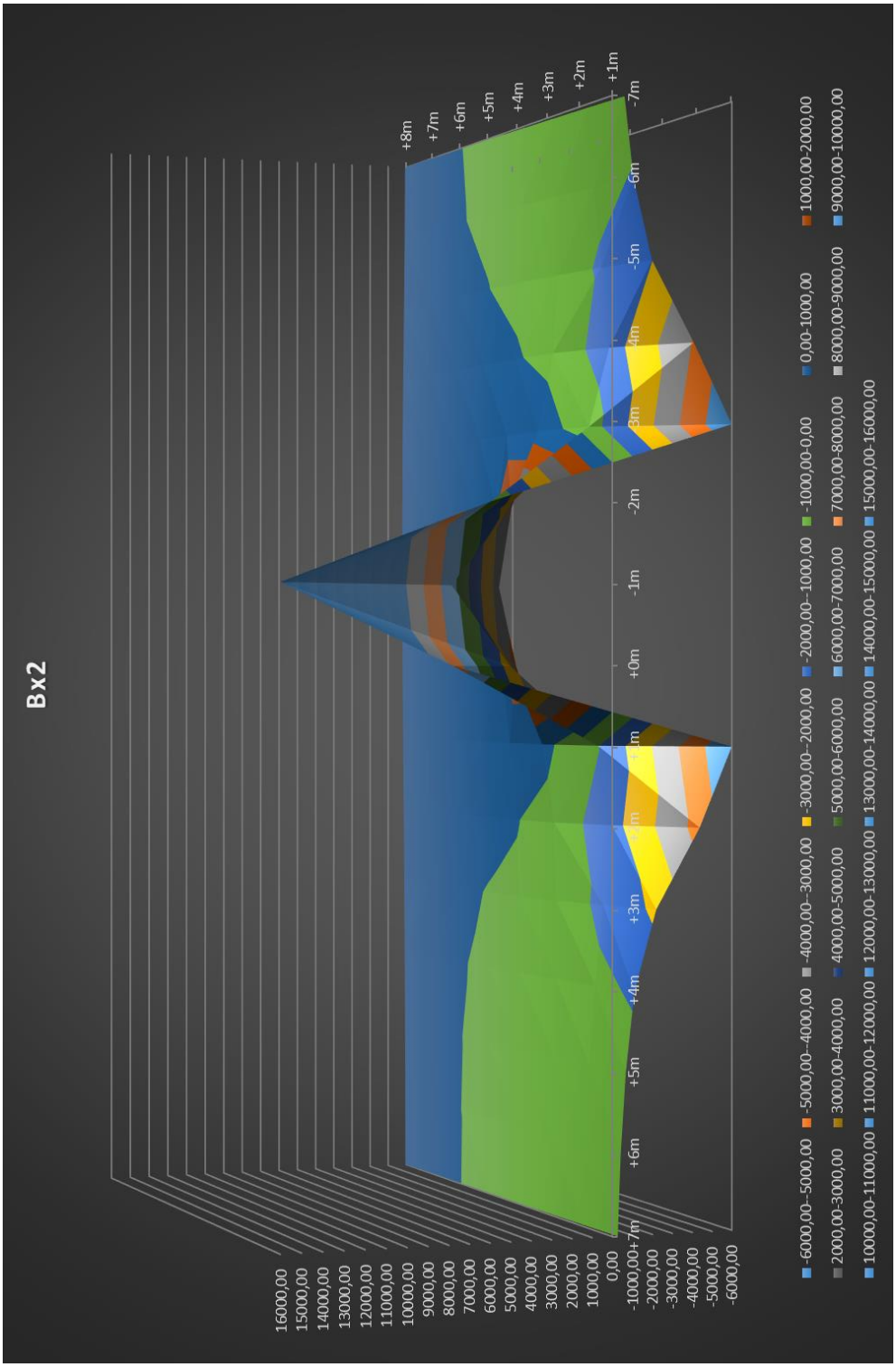


Figure K - 2: B_x data for sensor 2 with the 3 m magnetised pole held at the same height (600 mm above the ground) as the LEMI-011b magnetometers. . Please note the vertical axis displays the values in nT.

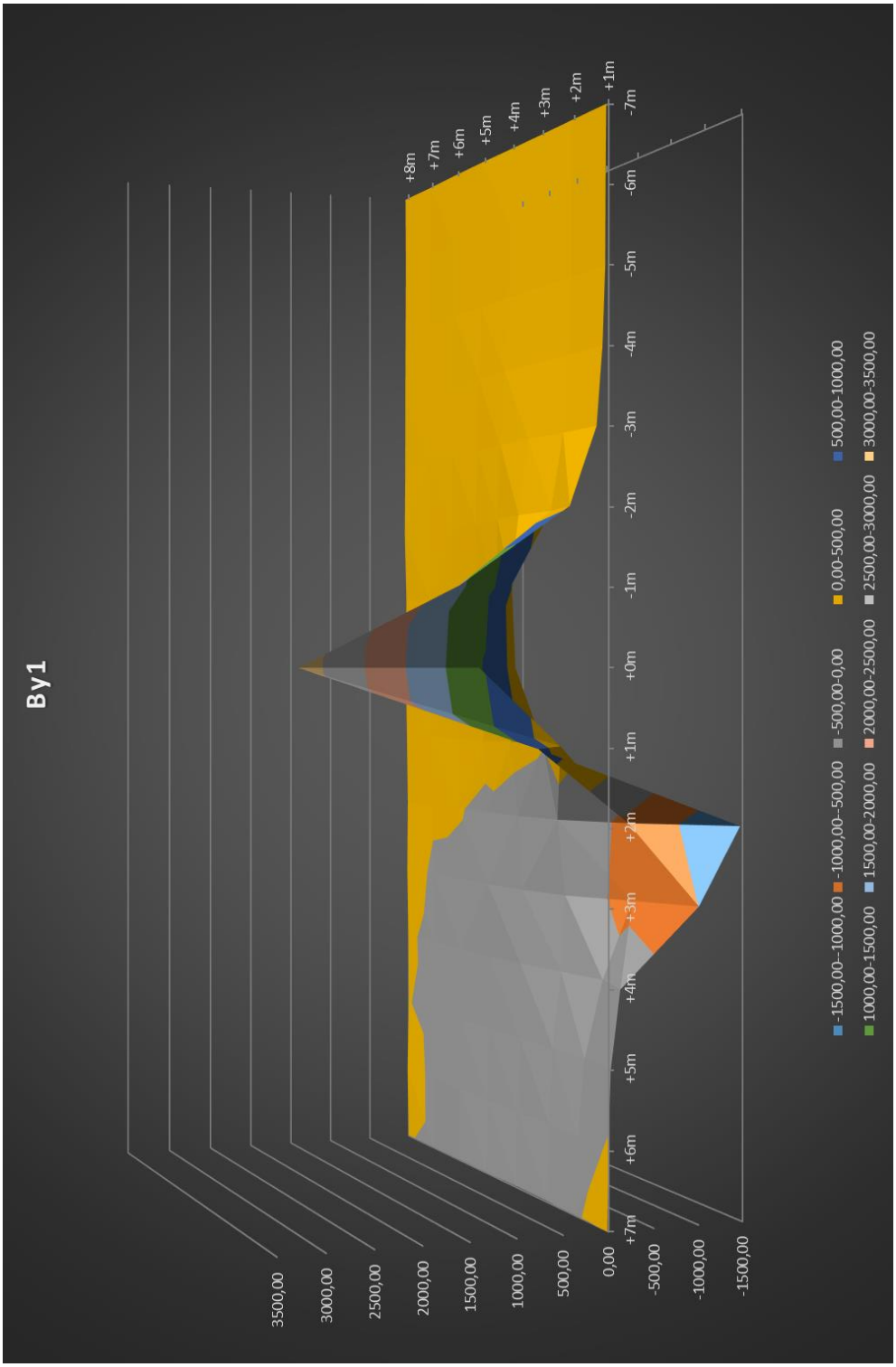


Figure K - 3: B_y data for sensor 1 with the 3 m magnetised pole held at the same height (600 mm above the ground) as the LEMI-011b magnetometers. . Please note the vertical axis displays the values in nT.

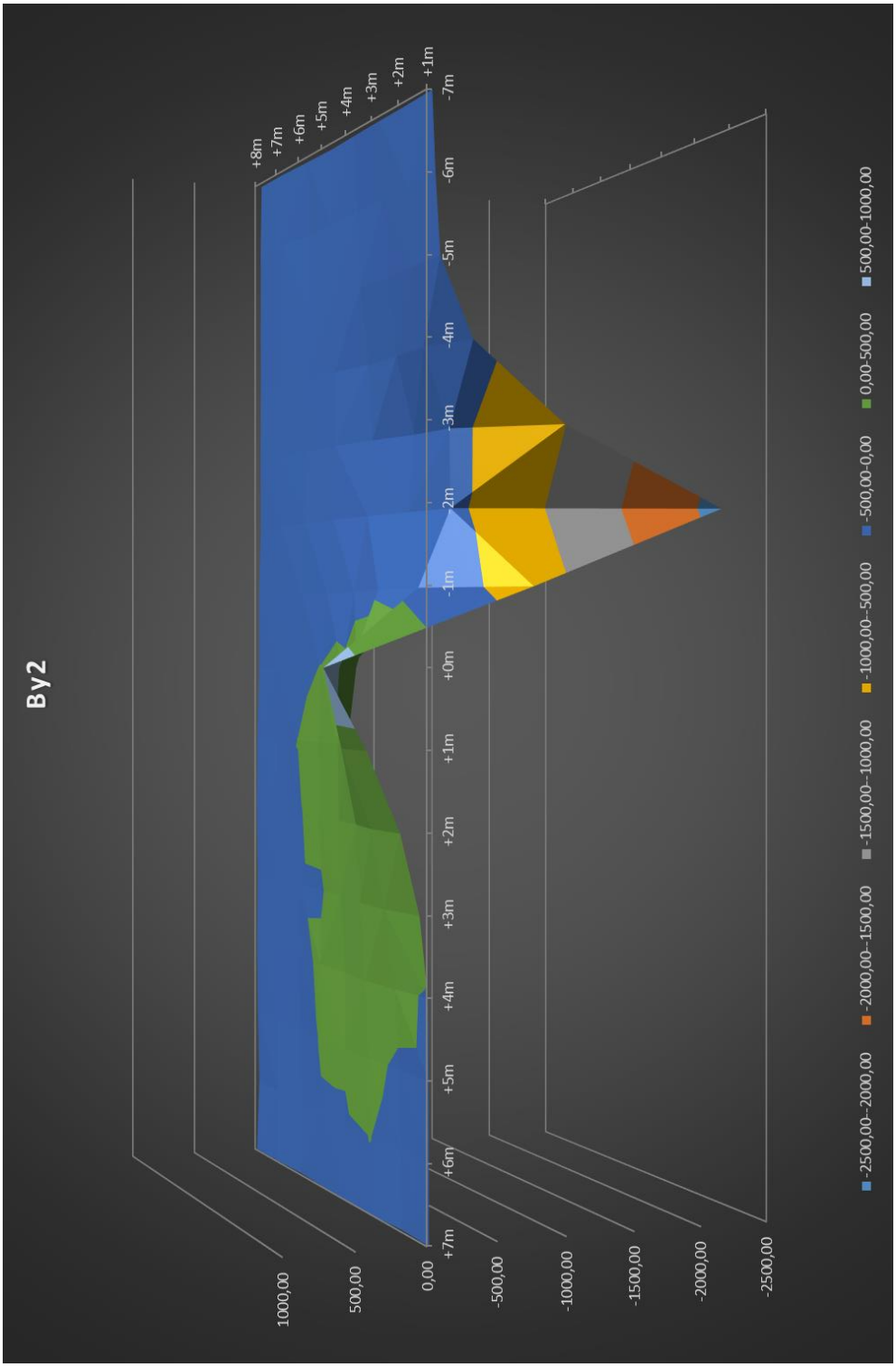


Figure K - 4: B_y data for sensor 2 with the 3 m magnetised pole held at the same height (600 mm above the ground) as the LEMI-011b magnetometers. . Please note the vertical axis displays the values in nT.

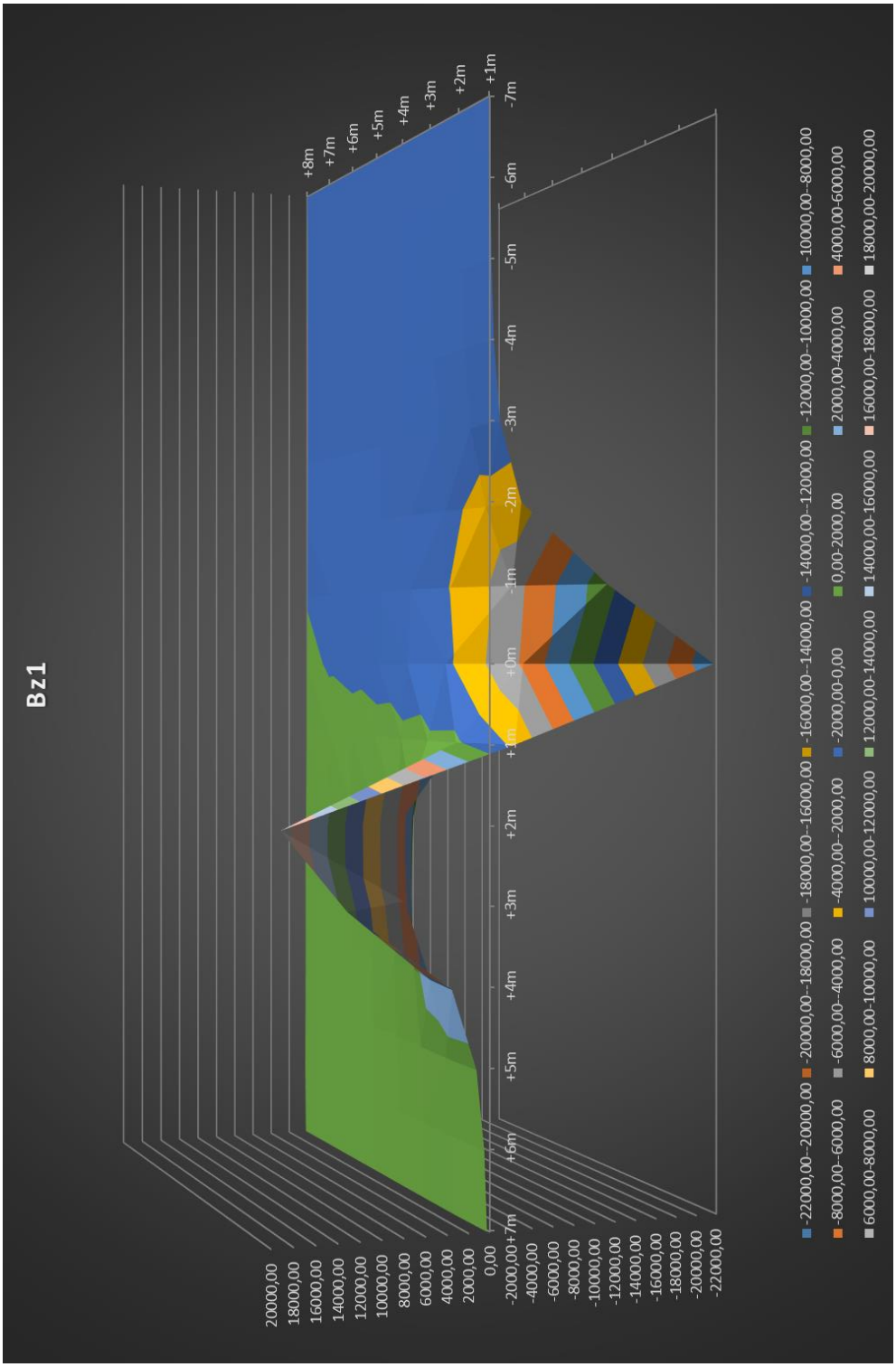


Figure K - 5: B_z data for sensor 1 with the 3 m magnetised pole held at the same height (600 mm above the ground) as the LEMI-011b magnetometers. . Please note the vertical axis displays the values in nT.

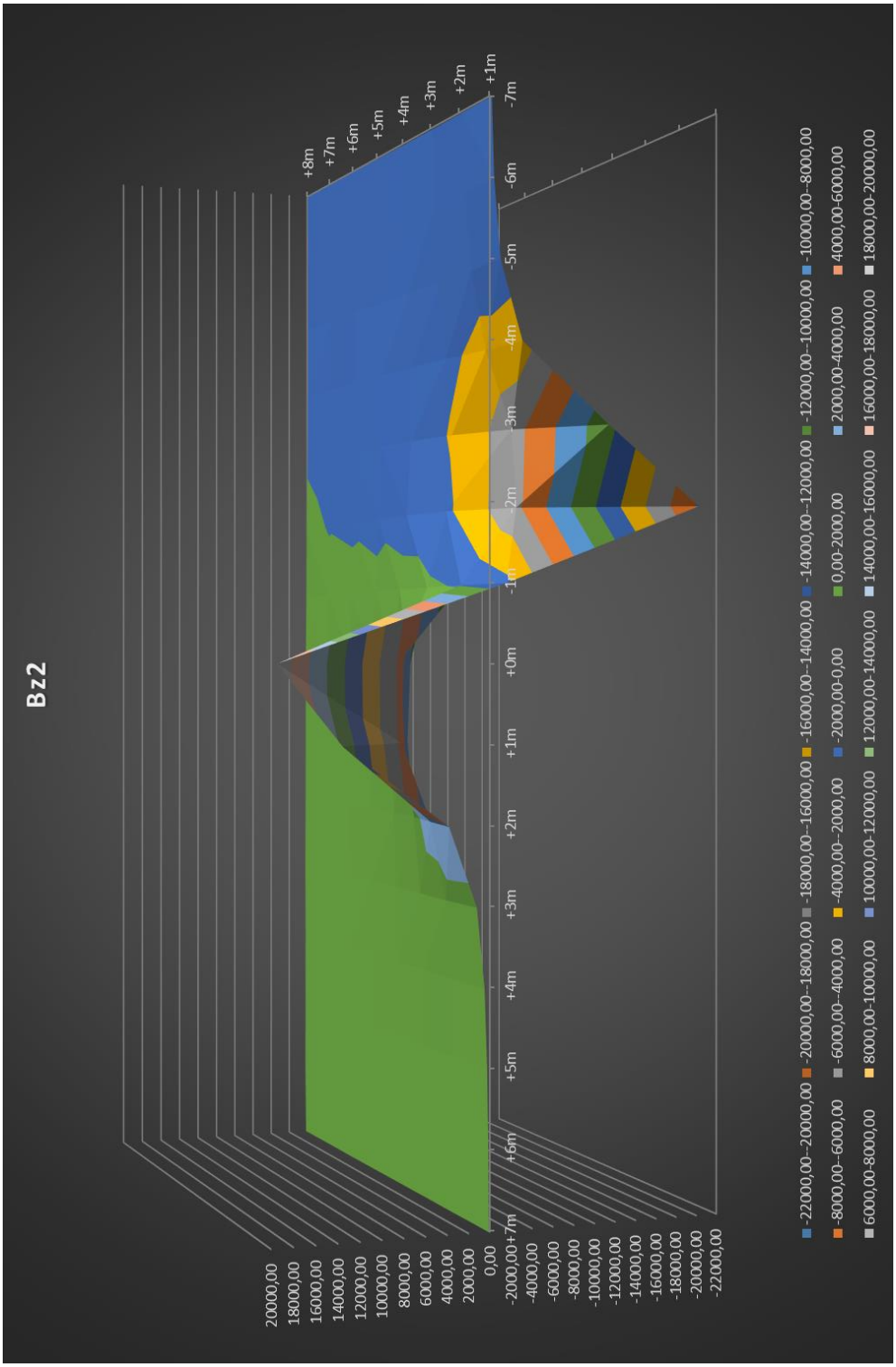


Figure K - 6: B_z data for sensor 2 with the 3 m magnetised pole held at the same height (600 mm above the ground) as the LEMI-011b magnetometers. . Please note the vertical axis displays the values in nT.

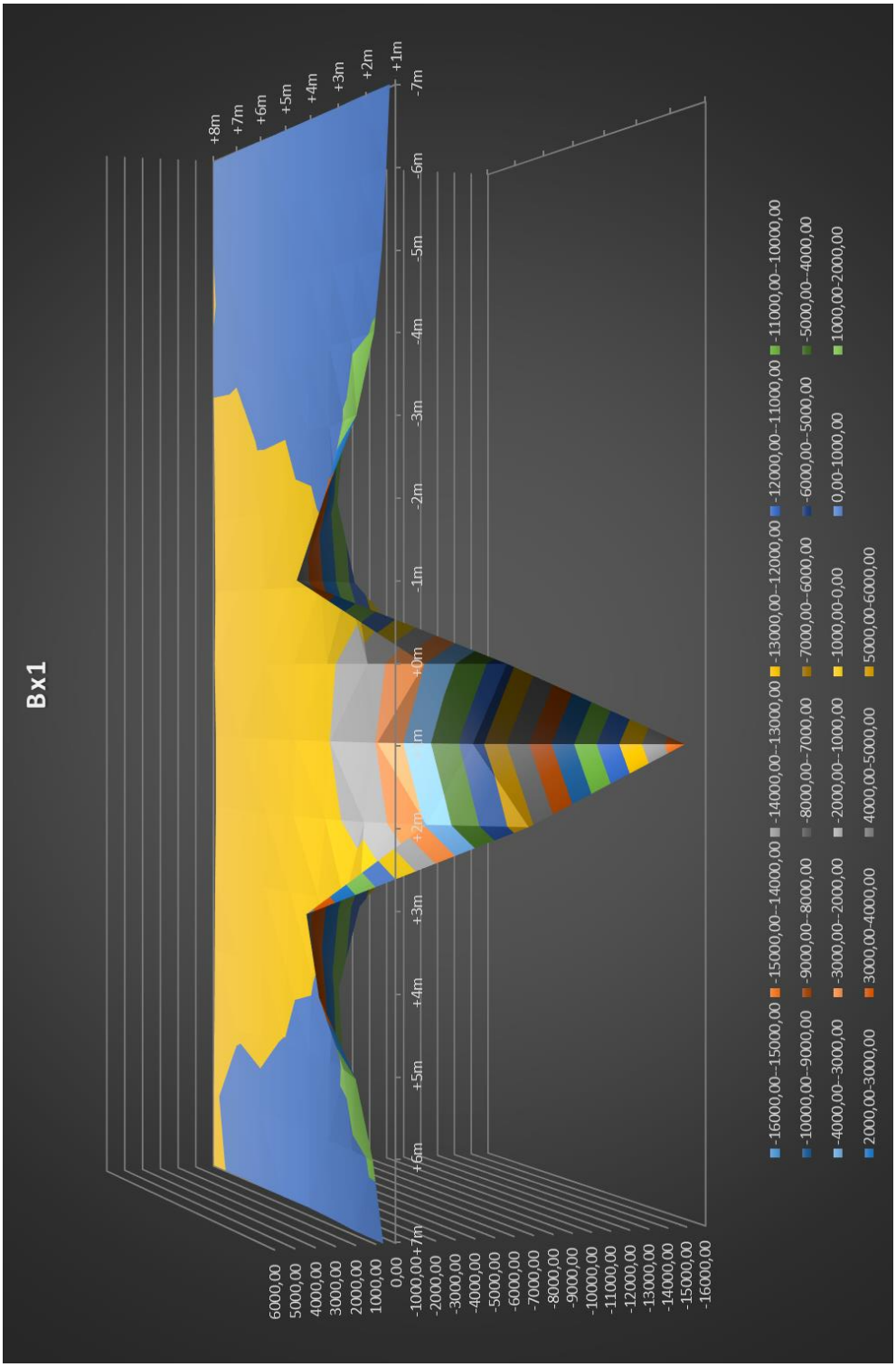


Figure K - 7: B_x data for sensor 1 with the 3 m magnetised pole 600 mm below (on the ground) the LEMI-011b magnetometers. . Please note the vertical axis displays the values in nT.

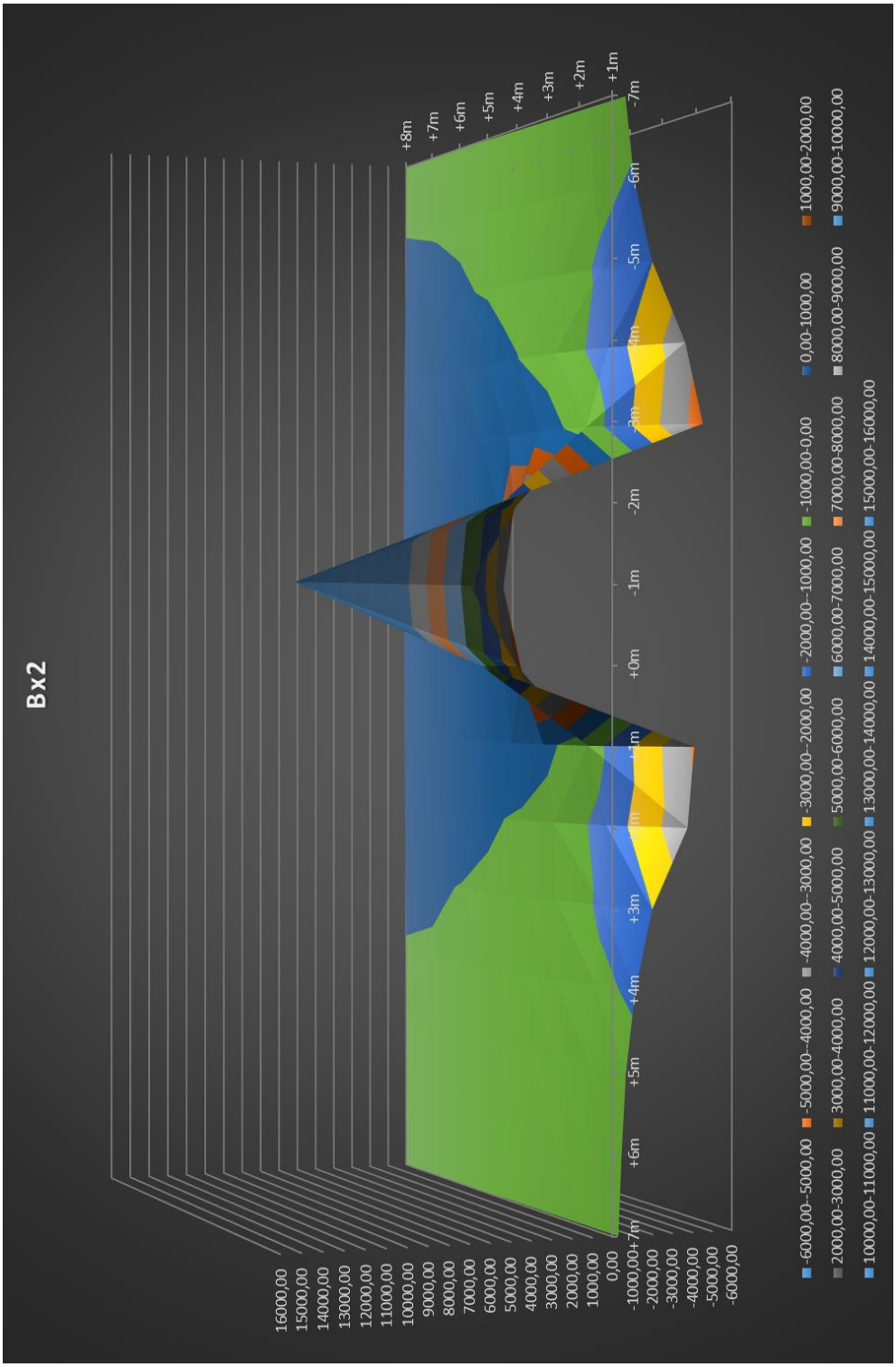


Figure K - 8: B_x data for sensor 2 with the 3 m magnetised pole 600 mm below (on the ground) the LEMI-011b magnetometers. . Please note the vertical axis displays the values in nT.

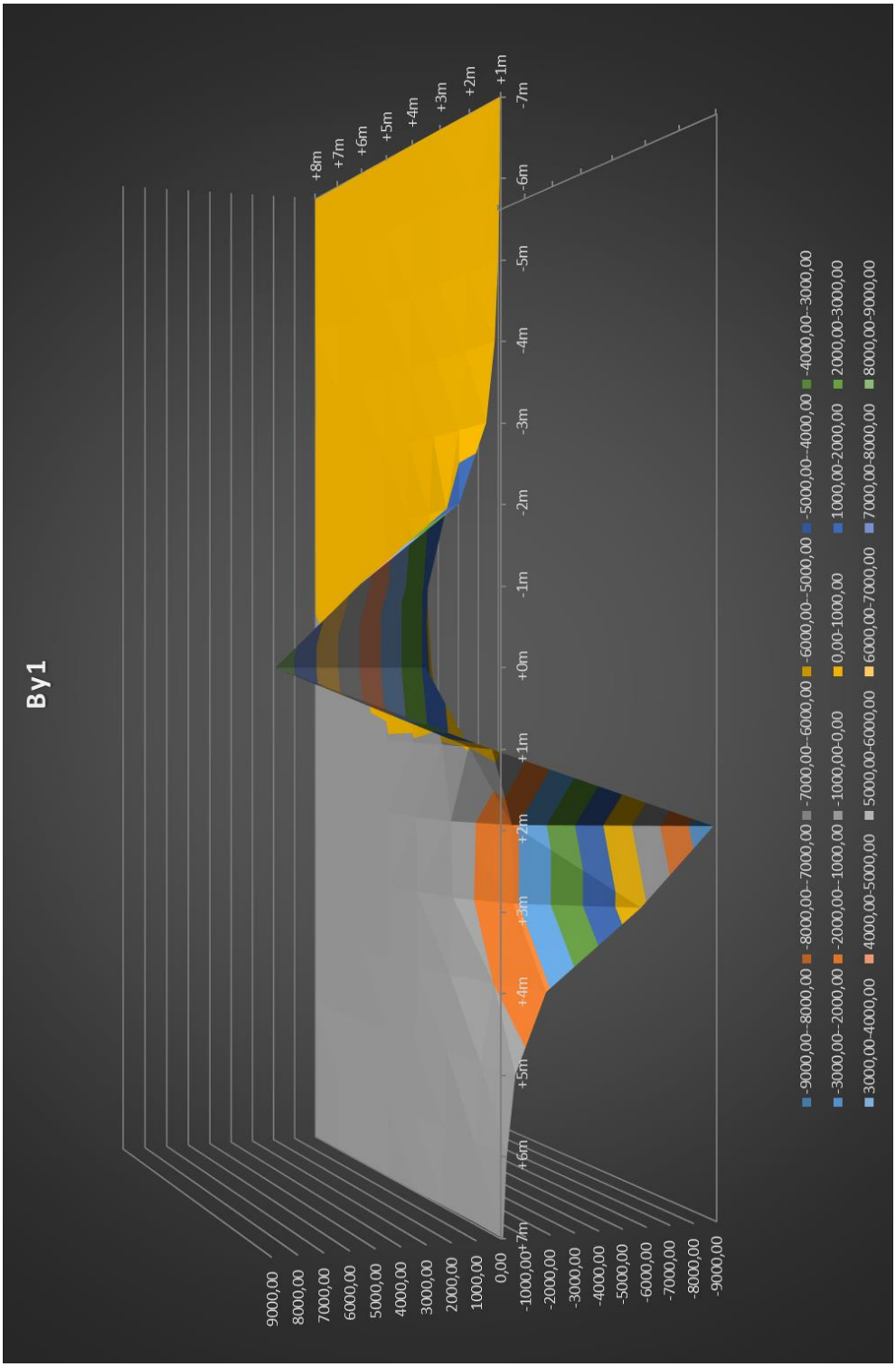


Figure K - 9: B_y data for sensor 1 with the 3 m magnetised pole 600 mm below (on the ground) the LEMI-011b magnetometers. . Please note the vertical axis displays the values in nT.

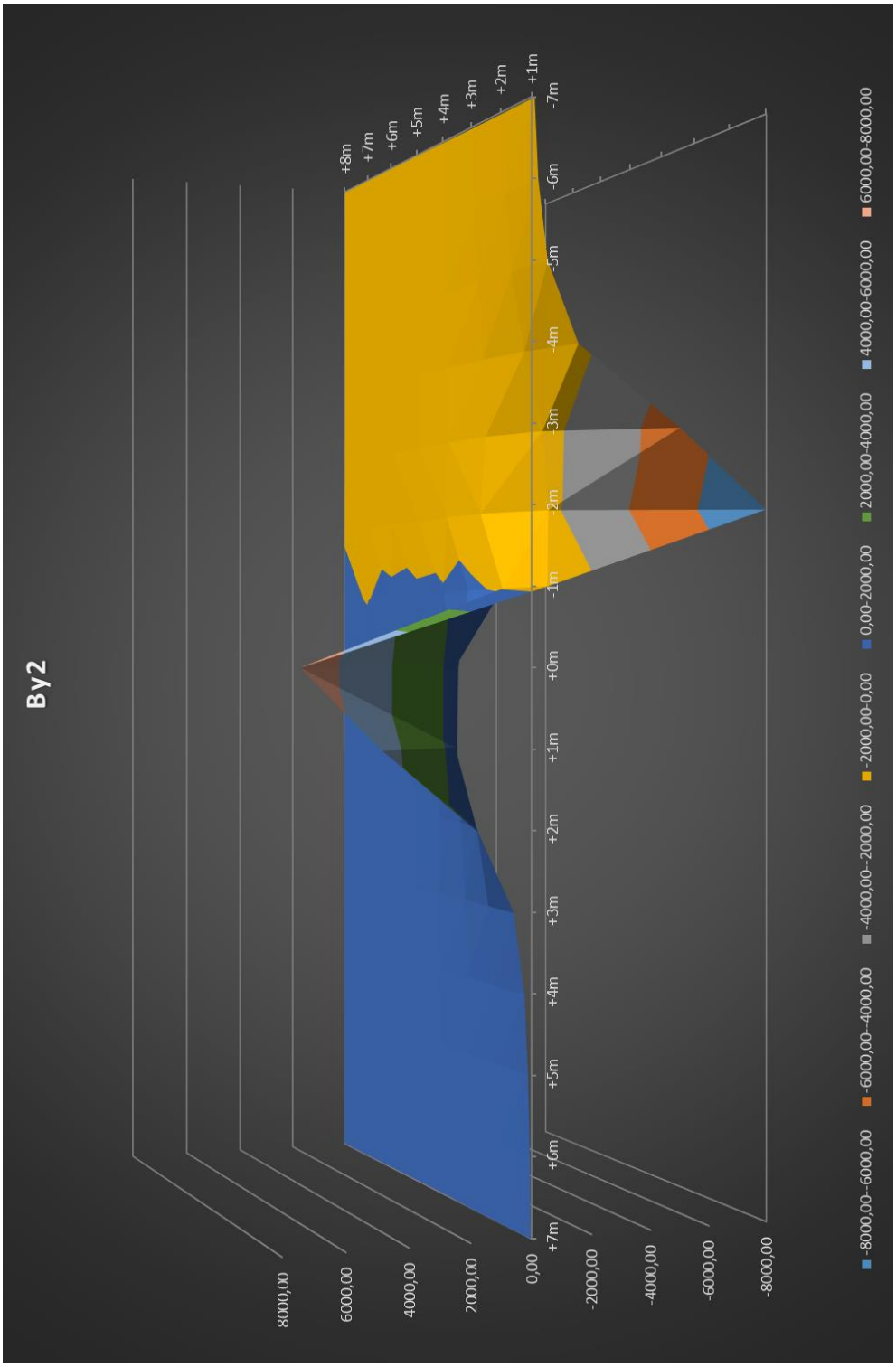


Figure K - 10: B_y data for sensor 2 with the 3 m magnetised pole 600 mm below (on the ground) the LEMI-011b magnetometers. . Please note the vertical axis displays the values in nT.

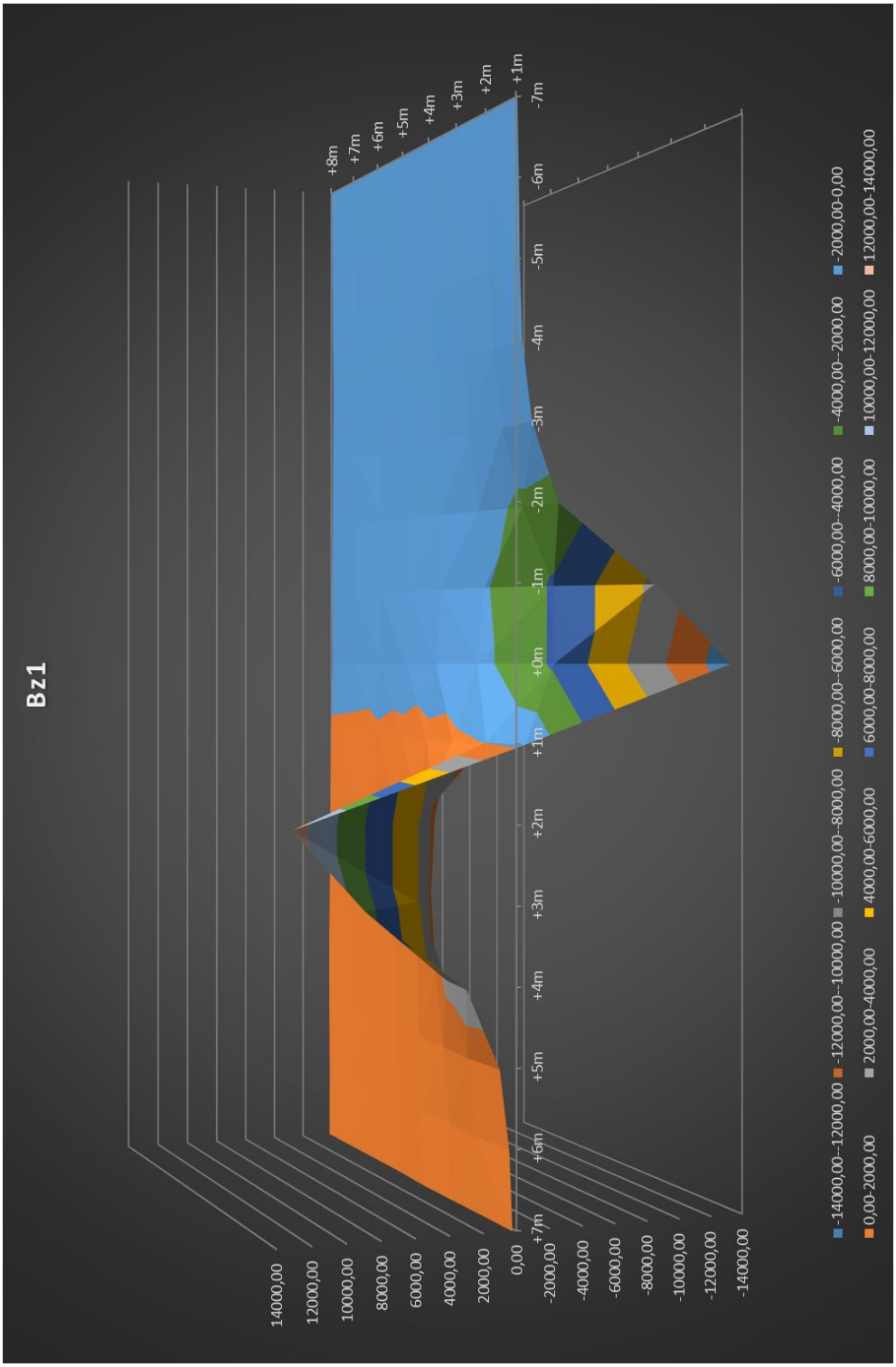


Figure K - 11: B_z data for sensor 1 with the 3 m magnetised pole 600 mm below (on the ground) the LEMI-011b magnetometers. . Please note the vertical axis displays the values in nT.

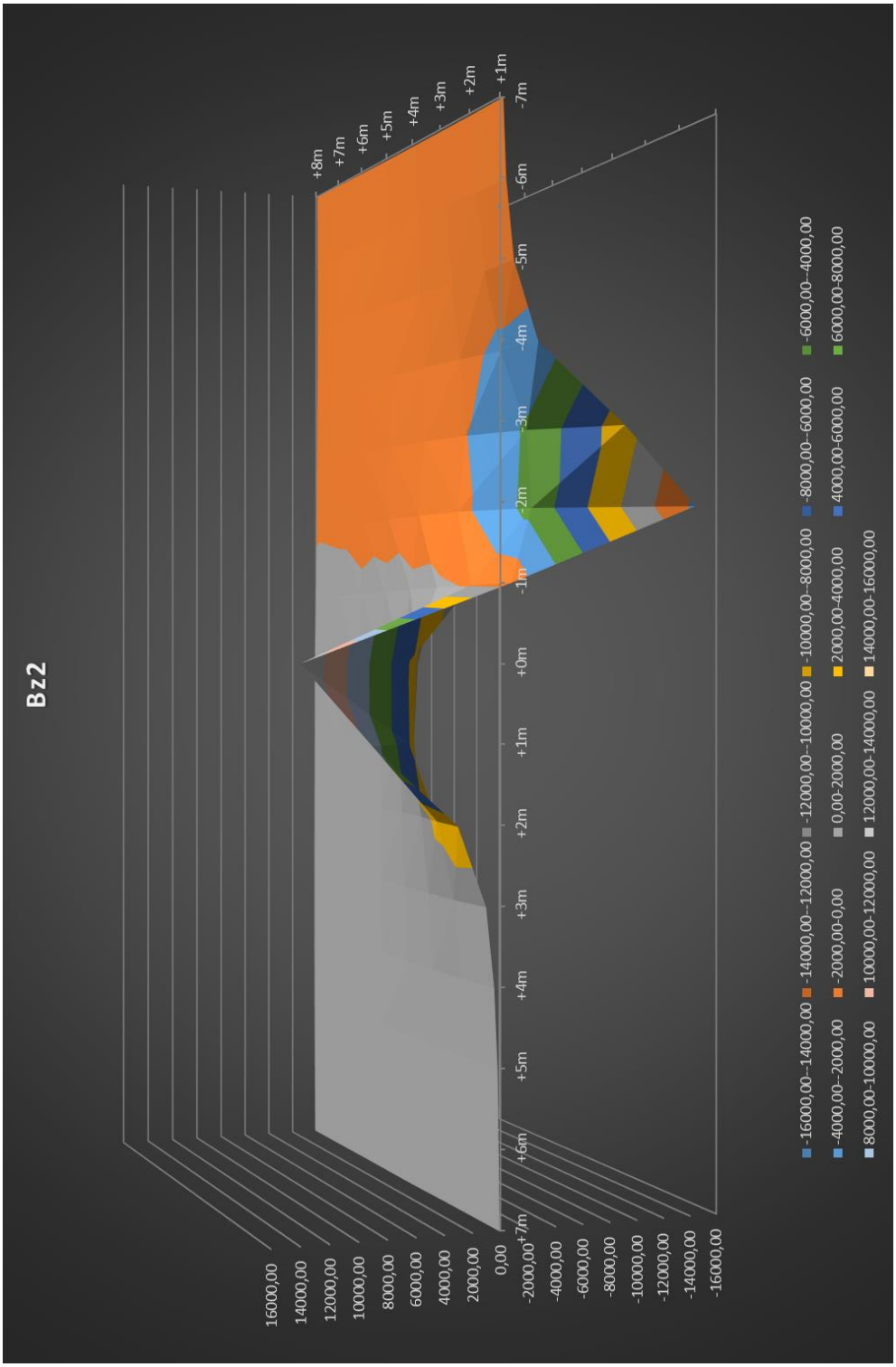


Figure K - 12: B_z data for sensor 2 with the 3 m magnetised pole 600 mm below (on the ground) the LEMI-011b magnetometers. . Please note the vertical axis displays the values in nT.

The End.