

# **POSITIONAL CONTROL OF THE HARTRAO AND C-BASS RADIO TELESCOPES**

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

MASTER OF SCIENCE

of

RHODES UNIVERSITY

by

**CHARLES JUDD COPLEY**

JANUARY 2008

## **Abstract**

This thesis investigates control aspects of the Hartebeeshoek Radio Astronomy Observatory (HARTRAO) antenna as well as the newly acquired C-BASS antenna. The effect of thermal conditions on the the HARTRAO antenna pointing offset is investigated, and a new control system is developed to allow the C-BASS antenna to be transformed from a communications device into an astronomical instrument.

## Acknowledgements

I would first and foremost like to thank my supervisors for their help over the past two years. Prof. Justin Jonas has always taken time out of a very busy schedule to give both advice and assistance. Dr. Jonathan Quick has given valuable insights into the thermal pointing problem during my trips to HARTRAO and the late Dr. Paul Nathanson has been a model to aspire to throughout my Rhodes University career.

The staff at HARTRAO have always been willing to provide assistance. Mr. Pieter Stronkhorst, Dr. Mike Gaylard, Mr. Keith Jones, Dr. Adrian Tiplady and Mr. Paul Manners all helped a great deal while I was stationed at HARTRAO. Paul did an unenviable task of tracing a particular wire through a maze of others for which I am most grateful.

The technical support from Mr. Antony Sullivan at Rhodes has been superb. Without it there would have been many additional hours of pointless head banging. Andrew Youthed provided good experiential advice with board design, and the rest of the Rhodes University Masters students have always been there with a good sense of humour.

I am most grateful for the financial support from DAAD, HARTRAO and Rhodes University in the form of the Rhodes University Postgraduate Scholarship. These have been exceptionally generous and have made the time spent researching a great deal more comfortable. I would also like to thank John Gillam and the team at Postgraduate Funding for being both efficient and friendly.

I also acknowledge the data analysis facilities provided by the Starlink Project which is run by CCLRC on behalf of PPARC.

Last but not least I would like to thank my family for their unconditional support over the years. This has been of great comfort to me.

# Contents

<b>Contents</b>	<b>i</b>
<b>List of Figures</b>	<b>vi</b>
<b>List of Tables</b>	<b>ix</b>
<b>1 Overview of Thesis</b>	<b>1</b>
1.1 Astronomy as a Priority in South Africa . . . . .	1
1.2 Relevance of Thesis to <i>MeerKAT</i> . . . . .	1
<b>I Temperature Dependence of the Pointing Error of the 26 m HARTRAO Telescope</b>	<b>3</b>
<b>2 Introduction</b>	<b>4</b>
2.1 Background . . . . .	4
2.1.1 Required Pointing Accuracy . . . . .	4
2.1.2 Modelling the Pointing Offset . . . . .	4
2.2 The Thermal Environment of the HARTRAO Antenna . . . . .	5
2.2.1 Telescope Structure and Solar Illumination . . . . .	5
2.2.2 Support Temperatures . . . . .	6
2.3 Previous Models . . . . .	10
2.4 Overview . . . . .	11
<b>3 Case Studies</b>	<b>12</b>
3.1 Green Bank Telescope . . . . .	12
3.2 MERLIN Radio Telescope in Cambridge . . . . .	13
3.3 Thermal Effects on the Leighton 10 m antennas at Owens Valley Radio Observatory . . . . .	14
3.4 The 45 m Nobeyama Telescope . . . . .	16
3.5 The HARTRAO Study . . . . .	17
<b>4 Modelling the Problem</b>	<b>18</b>
4.1 VLBI Pointing Model . . . . .	18
4.2 Thermal Expansion . . . . .	18
4.3 Simple Physical Model of the Declination Supports . . . . .	19
4.4 Correlation Coefficients . . . . .	21
4.5 Principal Component Analysis . . . . .	22
4.6 Linear Fitting of Weighted Inputs . . . . .	23
4.7 Neural Networks . . . . .	24
4.7.1 Overfitting . . . . .	24
4.8 Finite Element Analysis . . . . .	27

<b>5 Data Capture and Reduction</b>	<b>28</b>
5.1 The HARTRAO Antenna . . . . .	28
5.2 Telescope Control . . . . .	28
5.3 Experiment Design . . . . .	30
5.3.1 3.5 cm Receiver . . . . .	30
5.3.2 Thermistors . . . . .	30
5.3.3 Description of Observations . . . . .	31
5.3.4 Choice of observed object . . . . .	31
5.4 Measuring Pointing Offsets Using Point Sources . . . . .	34
5.4.1 Typical Data from Point Source Observations . . . . .	34
5.4.2 Elliptical Gaussians . . . . .	34
5.4.3 Other Functions . . . . .	37
5.5 A Priori Pointing Corrections . . . . .	38
5.6 Data Reduction . . . . .	38
5.6.1 Step One- Extraction and Reduction of Data from Observation FITS Files . . . . .	38
5.6.2 Step Two- Examining Different Pointing Model Effectiveness . . . . .	40
5.7 Ensuring Data Quality . . . . .	43
5.8 Initial Observations and Refitting the VLBI Pointing Model . . . . .	44
<b>6 Results</b>	<b>47</b>
6.1 Comparison to the Existing Heuristic Model . . . . .	47
6.2 The Relationship Between Thermal Conditions and Pointing Offset . . . . .	49
6.2.1 Correlation Matrix . . . . .	49
6.2.2 Binning of the Pointing Offsets . . . . .	52
6.2.3 Linear Fitting of the Pointing Offset to the Temperature Differential .	54
6.3 Simple Physical Model of the Declination Supports . . . . .	54
6.4 Principle Component Analysis . . . . .	55
6.5 Linear Fitting of a Weighted Combination . . . . .	60
6.5.1 Weighted Temperatures . . . . .	60
6.5.2 Weighted Principle Components . . . . .	61
6.6 Using the VLBI Pointing Model and Temperature Binning as a Solution . .	64
6.7 Neural Network . . . . .	67
6.7.1 Temperature Inputs . . . . .	67
6.7.2 Neural Networks as a Substitute for the VLBI Pointing Model . . . . .	69
<b>7 Conclusion</b>	<b>71</b>
<b>II A Prototype Control System for the C-BASS Antenna</b>	<b>73</b>
<b>8 Introduction</b>	<b>74</b>
8.1 The C-BASS Experiment . . . . .	74
8.2 Proposed C-BASS System . . . . .	76
8.3 Existing NEC System . . . . .	77
8.3.1 Existing Control System . . . . .	77
8.3.2 Existing Hardware . . . . .	79
8.4 DC Motor Transfer Function . . . . .	80
8.5 Rate feedback . . . . .	82
8.6 Fundamental Frequency of an Antenna . . . . .	82
8.7 Requirements of the New Antenna Control System . . . . .	83
8.8 Proposed Control System . . . . .	84
8.8.1 Testing and Implementation Difficulties . . . . .	85

8.9	Overview of the Following Chapters . . . . .	87
<b>9</b>	<b>Architecture and Hardware</b>	<b>88</b>
9.1	The ATMega128 Microcontroller . . . . .	88
9.2	Ethernet . . . . .	88
9.2.1	Why Ethernet? . . . . .	88
9.2.2	Potential Problems Using Ethernet as the Interface Medium . . . . .	89
9.2.3	Ethernet Controller . . . . .	89
9.3	External SRAM . . . . .	90
9.4	Digital to Analogue Converters (DAC) . . . . .	90
9.5	Angle Encoders . . . . .	91
9.6	RS-485 Drivers . . . . .	91
9.7	Analogue to Digital Sampling . . . . .	92
9.8	RS232 and UART Settings . . . . .	93
9.9	Prototyping and Board Design . . . . .	93
9.10	Future Hardware Improvements . . . . .	97
9.10.1	DAC . . . . .	97
9.10.2	A/D . . . . .	97
9.10.3	Trim Resistors . . . . .	97
9.10.4	Grounding . . . . .	98
9.10.5	16 and 32 bit microcontrollers . . . . .	99
<b>10</b>	<b>Software</b>	<b>100</b>
10.1	C Programming Language . . . . .	100
10.2	Control Process and Data Flow . . . . .	101
10.2.1	Control Board Data Transfer . . . . .	103
10.2.2	Control Board Commands . . . . .	103
10.2.3	Command Angle Buffer . . . . .	105
10.2.4	PID Control Algorithm . . . . .	106
10.2.5	Failsafe Overrides . . . . .	107
10.3	Power Up Process . . . . .	108
10.4	Interrupt Driven Processes . . . . .	110
10.4.1	External Interrupt 4 - Emergency Override . . . . .	110
10.4.2	Timer Counter 3C Compare Interrupt - Servoloop Sampling . . . . .	110
10.4.3	External Interrupt 5 - 1-PPS . . . . .	111
10.4.4	External Interrupt 6 - Ethernet Interrupt . . . . .	112
10.4.5	Timer Counter 3B Compare Interrupt . . . . .	113
10.5	Ethernet Configuration . . . . .	114
10.5.1	Initialising . . . . .	114
10.5.2	Packet Filters . . . . .	115
10.5.3	MAC Address, IP Address, Port Settings and NTP and ARP . . . . .	115
10.5.4	Available Microcontroller Ethernet Software . . . . .	115
10.6	PC Code . . . . .	116
10.7	Existing bugs . . . . .	116
10.7.1	Sending Packets Errata . . . . .	116
10.8	Future Improvements . . . . .	117
<b>11</b>	<b>Performance and Reliability Tests</b>	<b>118</b>
11.1	Equipment . . . . .	118
11.2	PID tuning . . . . .	120
11.3	Adaptive PID Control . . . . .	122
11.4	Reliability Tests . . . . .	124

<b>12 Conclusion</b>	<b>125</b>
<b>Bibliography</b>	<b>126</b>
<b>III Appendix</b>	<b>129</b>
<b>A Elliptical Gaussian Uncertainties</b>	<b>130</b>
A.1 Error Matrix and Error Propagation . . . . .	131
A.2 Estimates of Uncertainties in this data . . . . .	133
<b>B VLBI Parameter listing</b>	<b>135</b>
B.1 Description of the model and parameters . . . . .	135
B.2 Discussion of Model . . . . .	136
<b>C DC Motor Transfer Function</b>	<b>137</b>
<b>D Frequency Analysis</b>	<b>140</b>
D.1 Bode Plots . . . . .	140
<b>E Ethernet Communication</b>	<b>143</b>
E.1 Ethernet Communication . . . . .	143
E.2 Data Frames . . . . .	144
E.2.1 Ethernet Frames (Datalink Layer) . . . . .	144
E.2.2 IP and ARP (Internet Layer) . . . . .	145
E.2.3 TCP/UDP Frames (Transport Layer) . . . . .	145
E.3 Time Keeping . . . . .	148
E.3.1 NTP/SNTP . . . . .	148
E.3.2 NTP Strata . . . . .	149
E.3.3 Different Time Stamping Options for SNTP . . . . .	149
E.4 ARP Frame Examples . . . . .	154
<b>F Description of the Existing System</b>	<b>155</b>
F.1 Layout . . . . .	155
F.1.1 Terminal Blocks . . . . .	155
F.2 Hardware . . . . .	163
F.2.1 DCPA CPU Board . . . . .	163
F.2.2 Relay Board . . . . .	163
F.2.3 Servopack . . . . .	163
F.2.4 Drive System . . . . .	163
F.2.5 Angle Encoders . . . . .	163
<b>G Prototype Control Board Screenshots</b>	<b>172</b>
<b>H Required External Jumpers for the Prototype Board</b>	<b>176</b>
<b>I Additional Prototype Board Information</b>	<b>184</b>
I.1 ENC28J60 SPI Commands . . . . .	184
I.2 ENC28J60 Initialisation . . . . .	184
I.3 ATMega128 Fuse Settings . . . . .	187
I.4 A/D Analogue Circuitry . . . . .	187
I.5 Commands Available to the Control Board . . . . .	189
I.5.1 Summary of Commands . . . . .	189
I.6 Compiling the code . . . . .	192

I.7	Code Layout . . . . .	192
I.7.1	AVRGCC . . . . .	192
I.7.2	C source files . . . . .	192
I.7.3	C header files . . . . .	193
<b>J</b>	<b>PID Tuning Examples using the prototype control board</b>	<b>195</b>
J.1	Step Response with no control . . . . .	195
J.2	Control and PID Tuning . . . . .	196
J.3	Adaptive PID Control . . . . .	200
<b>K</b>	<b>Datasheets</b>	<b>202</b>
K.1	ENC28J60 . . . . .	202
K.2	ROC 416 . . . . .	203
K.3	FRABA Encoder . . . . .	206

# List of Figures

2.1	Photos of the HARTRAO Antenna . . . . .	7
2.2	HARTRAO Temperatures throughout the year . . . . .	8
2.3	HARTRAO Temperatures throughout the year . . . . .	9
2.4	Original thermal pointing offset model . . . . .	10
3.1	MERLIN Telescope support structure . . . . .	13
3.2	OVRO 10 m Leighton Array . . . . .	15
3.3	Finite Element model of the Leighton Array backing structure . . . . .	16
3.4	Comparison between FEA of the OVRO telescope and temperature gradient model . . . . .	16
4.1	Structural models of the HARTRAO Telescope . . . . .	20
4.2	Neural Network Transfer Functions . . . . .	25
4.3	Neural Networks . . . . .	26
5.1	Typical cross scans . . . . .	33
5.2	Typical Raw Data from a Set of Cross Scans . . . . .	34
5.3	Typical Data from a Set of Cross Scans with Baselines Removed . . . . .	35
5.4	An Elliptical Gaussian . . . . .	35
5.5	Closer to the Elliptical Gaussian . . . . .	36
5.6	Flow diagram of the Fits2point Fortran code. . . . .	41
5.7	Screenshot of the MATLAB GUI interface used for the data analysis . . . . .	42
5.8	Refitting the VLBI Pointing Model . . . . .	46
6.1	Comparison of wrapped pointing offsets . . . . .	48
6.2	Binning the pointing offsets by AT5-AT6 . . . . .	52
6.3	Binning the data by AT9-AT10 . . . . .	53
6.4	Binning the data by ambient temperature . . . . .	53
6.5	Relationships between temperature differentials and pointing offsets . . . . .	54
6.6	Simple geometric model of Declination support thermal deformations . . . . .	55
6.7	The Principle Components of the Declination Support temperatures . . . . .	57
6.8	The Principle Components of the Hour Angle Support temperatures . . . . .	57
6.9	The second Principle Component plotted against the Declination temperature differential, AT5-AT6 . . . . .	59
6.10	Pointing offsets and the second Principle Component . . . . .	59
6.11	AT1 → AT10 and $T_{ambient}$ . . . . .	62
6.12	Weighting the Principle Components as a means of modelling the pointing offsets	63
6.13	VLBI Coefficients fitted to binned data . . . . .	65
6.14	AT5-AT6 Binning and resulting pointing offset corrections . . . . .	66
6.15	Predictions by a neural network model . . . . .	68
6.16	Neural network trained to replace the VLBI pointing model . . . . .	70
8.1	Photograph of the C-BASS 7.6 m antenna . . . . .	76

8.2	The layout of the proposed C-BASS system showing the receiver, controller and 1-PPS synchronisation signal . . . . .	77
8.3	Overview of the existing control system . . . . .	78
8.4	DCPA Control Board and Relay Board . . . . .	79
8.5	DC Motor servo systems . . . . .	81
8.6	Classic antenna control systems . . . . .	82
8.7	Antenna Fundamental Frequencies . . . . .	83
8.8	A functional block diagram of the proposed C-BASS control system. . . . .	85
8.9	A functional block diagram of the Antenna Control board . . . . .	86
8.10	Functional block diagram of the control board to be used for limit switches and clutches/brake control . . . . .	86
9.1	The signal conditioning circuitry used in the tachometer sampling . . . . .	92
9.2	Prototype Boards . . . . .	95
9.3	Functional block diagram of the prototype control board . . . . .	96
10.1	The layout of the servoloop system . . . . .	101
10.2	Low-level diagram of the servoloop . . . . .	102
10.3	Boot/idle status and interrupt driven processes of the control board . . . . .	109
10.4	Flow Diagram of processes triggered by Emergency Stop Interrupt (INT4) . .	110
10.5	Flow diagram of the processes triggered by a Counter 3C Interrupt on Compare Match . . . . .	111
10.6	Flow diagram of the processes triggered by a 1-PPS Signal (INT5) . . . . .	112
10.7	Flow diagram of the processes triggered by a ENC28J60 Interrupt Signal (INT6)	113
10.8	Flow diagram of the processes triggered by a Counter 3B Interrupt on Compare Match . . . . .	114
11.1	Photograph of control test equipment . . . . .	119
11.2	Diagram of the equipment used in the control testing . . . . .	120
11.3	Tuning the PID Coefficients . . . . .	121
11.4	Controller with different command signals . . . . .	123
A.1	Typical Position Uncertainties . . . . .	134
C.1	Schematic of a DC motor . . . . .	137
D.1	Bode Plots of second order transfer function . . . . .	142
E.1	OSI Reference Model Layers . . . . .	143
E.2	Data Encapsulation . . . . .	144
E.3	A typical Ethernet frame . . . . .	145
E.4	Ethernet Type II frame . . . . .	145
E.5	Comparison between UDP and TCP packets . . . . .	147
E.6	An overview of the NTP time synchronisation process . . . . .	149
E.7	An example of a typical NTP packet . . . . .	151
E.8	Time stamping at the application layer . . . . .	152
E.9	Time stamping at the Ethernet level . . . . .	152
E.10	Timestamp at the datalink/physical layer level . . . . .	153
E.11	A typical ARP request frame . . . . .	154
E.12	A typical ARP reply frame . . . . .	154
F.1	Terminal Block 104 TB . . . . .	156
F.2	Terminal Block 102 TB . . . . .	157
F.3	Terminal Block 103 TB . . . . .	158

F.4	Terminal Block 108 TB . . . . .	159
F.5	Layout of the DCPA . . . . .	160
F.6	Overview of the existing control system . . . . .	161
F.7	DCPA Control Board and Relay Board . . . . .	162
F.8	The layout of the DCPA Control Board . . . . .	165
F.9	Diagram of the DCPA Relay Board I/O Ports . . . . .	166
F.10	Wiring Diagram for the AZ1 Servopack . . . . .	167
F.11	Wiring Diagram for the AZ2 Servopack . . . . .	168
F.12	Wiring Diagram for the EL1 Servopack . . . . .	169
F.13	Wiring Diagram for the EL2 Servopack . . . . .	170
F.14	Angle Encoder Layout . . . . .	171
G.1	Both sides of the control board . . . . .	173
G.2	Top side of the control board . . . . .	174
G.3	Bottom side of the control board . . . . .	175
H.1	External jumpers for prototype board . . . . .	176
H.2	External jumpers for <i>ETHERCHECK</i> . . . . .	177
H.3	External jumpers for <i>232A</i> . . . . .	177
H.4	External jumpers for 489A and 489B . . . . .	178
H.5	External jumpers for <i>INVERTER IN</i> and <i>INVERTER OUT</i> . . . . .	179
H.6	External jumpers for AD667(a) . . . . .	180
H.7	External jumpers for AD667(b) . . . . .	181
H.8	External jumpers for power connections . . . . .	182
H.9	External jumpers for A/D analogue circuitry . . . . .	183
I.1	ENC28J60 SPI Commands . . . . .	184
I.2	A schematic of the analogue input circuitry to the A/D. . . . .	188
J.1	Test equipment layout . . . . .	195
J.2	The motor 6 V step responses . . . . .	195
J.3	Layout of the equipment used in the obtaining these results. . . . .	196
J.4	Tuning the D coefficient . . . . .	196
J.5	P tuning-Settled on P = between 4 and 8 . . . . .	197
J.6	I tuning-Settled on I =2 . . . . .	198
J.7	Final Tuning . . . . .	199
J.8	Command = 10000 . . . . .	200
J.9	Command = 5000 . . . . .	201
K.1	Schematic for the ENC28J60 board used . . . . .	202
K.2	ROC 416 Interface . . . . .	203
K.3	ROC 416 wiring diagram Part 1 . . . . .	204
K.4	ROC 416 wiring diagram Part 2 . . . . .	205

# List of Tables

2.1	Abbreviations used to indicate thermistor positions . . . . .	6
4.1	Dimensions of the HARTRAO support structure . . . . .	19
5.1	Data pre-processing output files . . . . .	39
5.2	VLBI Pointing Model Parameters and the results of refitting the parameters using these observations. . . . .	45
6.1	Correlation Matrix (R) of the support temperatures, the ambient temperature and the pointing offsets. . . . .	51
6.2	Correlation Matrix (R) of the temperature differential between axis supports and the pointing offsets. . . . .	51
6.3	Correlation matrix between the Principle Components of all thermistor readings and the pointing offsets. . . . .	58
6.4	Correlation matrix between the Principle Components of only the Hour Angle thermistor readings and the pointing offsets. . . . .	58
6.5	VLBI Pointing Model Coefficients . . . . .	64
9.1	RS232 format used for communications in the final version of the code. . . . .	93
B.1	VLBI Pointing Model Parameters . . . . .	136
F.1	Description of the DCPA Relay Board I/O Cables . . . . .	164
I.1	Fuse settings of the ATMega 128 . . . . .	187
I.2	Analogue input circuitry resistor values used . . . . .	188
K.1	Fraba Angle Encoder wiring diagram . . . . .	206

# Chapter 1

## Overview of Thesis

### 1.1 Astronomy as a Priority in South Africa

Astronomy has been listed as a priority for South African science as a result of the Astronomy Geographic Advantage Programme. Recent years have seen South Africa emerging as a leading contender to host a number of prestigious astronomical instruments. In the optical domain, the Southern African Large Telescope (SALT) has been built at the South African Astronomical Observatory (SAAO) site in Sutherland, Northern Cape. The world has not lost sight of South Africa's numerous advantages with regards radio telescopes either and South Africa has been shortlisted, alongside Australia, to host the world's most ambitious radio telescope, the Square Kilometre Array (SKA). The Hartebeeshoek Radio Astronomical Observatory (HARTRAO) 26 m antenna is also situated in the country and is a partner in the C-BAND All Sky Survey (C-BASS) project which requires the transformation of a communications antenna into a radio telescope.

As a precursor to the announcement of the winning SKA bid, South Africa is currently building a smaller array called *MeerKAT*, an SKA science and technology technology demonstrator. The aim behind the project is to build a world-class scientific instrument, develop local expertise and to demonstrate South Africa's ability and commitment to hosting the SKA.

### 1.2 Relevance of Thesis to *MeerKAT*

Aspects that need to be investigated and developed for *MeerKAT* are models of thermally induced pointing offsets and a suitably generic control system for use with the individual antennas. It is well known that pointing offsets dramatically reduce the dynamic range of mosaic images produced by array telescopes as well as effecting the sensitivity (and thus consistency) of single dish observations (Ukita, 1999).

Part I of this investigation will be to investigate thermally induced pointing offsets seen

on the HARTRAO antenna. A recent receiver upgrade means that modelling these effects is now necessary at HARTRAO and gaining an understanding of the thermal effects on antenna pointing with regards *MeerKAT* will be useful. Part II of this thesis examines the suitability of a simple, cost effective control board, initially for use with the newly acquired C-BASS antenna, but designed with implementation on *MeerKAT* in mind.

## **Part I**

# **Temperature Dependence of the Pointing Error of the 26 m HARTRAO Telescope**

# Chapter 2

## Introduction

### 2.1 Background

In February 2007, Hartebeeshoek Radio Astronomy Observatory (HARTRAO) commissioned and installed a new 1.3 cm (22 GHz) receiver. Previously, the highest observing frequency had been 12 GHz (2.5 cm). Since the approximate beamwidth is given by  $\theta_{HPBW} = \frac{1.2\lambda}{D}$  (where  $\lambda$  is the observation wavelength and  $D$  is the telescope diameter), this upgrade will result in nearly halving the smallest beamwidth from 80 mdeg to approximately 40 mdeg.

#### 2.1.1 Required Pointing Accuracy

In general, 'blind' radio telescope pointing needs to be accurate to 1/10th of the beamwidth, with array telescopes making interferometric measurements (such as *MeerKAT*) requiring even greater accuracies. This requirement maintains pointing to well inside a window of 1/5th of the beamwidth. Anything larger than this results in a greater than 10% loss of sensitivity of the receiver (Ukita, 1999), causing inconsistent observations which should be avoided. The consequence of this is that the HARTRAO telescope pointing accuracy will need to be improved to approximately 4 mdeg (or 1/10 of the beamwidth) as a requirement for consistent observations using the new receiver.

#### 2.1.2 Modelling the Pointing Offset

A number of potential causes of pointing offsets need to be considered when pointing a telescope. These effects include:

- Gravitational deformation of the antenna structure and surface
- Axis Encoder Offsets
- Axis misalignments
- Atmospheric Refraction

- Wind Loading on the antenna structure and surface
- Thermal deformations of the structure and surface

The telescope at HARTRAQ uses the VLBI pointing model (further detail available in Appendix B) to remove the non-temporal effects such as gravitational deformations, axis misalignments and encoder offsets together with a standard refraction model to account for atmospheric refraction effects, however neither of these models include corrections for temporal effects such as thermal and wind loading.

Increasing the required pointing accuracy means that thermally induced pointing offsets can no longer be considered insignificant and need to be modelled and incorporated into the current pointing correction models. This thesis deals with an examination of the temperatures of the Declination and Hour Angle axis support struts, and the role that these temperatures may play in causing pointing offsets.

The HARTRAQ Antenna is equatorially mounted and it should be noted that Hour Angle and Right Ascension are a left-handed and right-handed system related to each other through the local sidereal time. Hour Angle will be used when referring to the local coordinate system of the HARTRAQ antenna e.g Hour Angle supports, while Right Ascension will be used when referring to observations e.g Right Ascension pointing offset.

The investigation includes using simple physical models, linear fitting, adaptation of the VLBI pointing model, and neural networks to create a model for the pointing offsets using thermal inputs.

## 2.2 The Thermal Environment of the HARTRAQ Antenna

An examination of the thermal environment of the HARTRAQ antenna is critical to begin understanding the thermally induced pointing offsets experienced at HARTRAQ. Throughout this thesis, abbreviations will be used to indicate the position of thermistors recording temperature data. These abbreviations are included in Table 2.1. Figures 2.1 (a) → (c) give an idea of the positions of the supports, the orientation of the supports and approximate positions of the temperature sensors.

### 2.2.1 Telescope Structure and Solar Illumination

Figures 2.1a, 2.1b and 2.1c are photographs taken of the HARTRAQ structure in July 2006, from orthogonal viewpoints at approximately the same height in an effort to obtain the dimensions of various beams. A simple structural diagram is included in Figure 2.1d. It is important to note the solar illumination on the Declination support beams. Since HARTRAQ

Table 2.1: Abbreviations used to indicate thermistor positions.

Label	Description
AT1	Top North West Declination Support Temperature
AT2	Top South West Declination Support Temperature
AT3	Bottom North West Declination Support Temperature
AT4	Bottom South West Declination Support Temperature
AT5	Top North East Declination Support Temperature
AT6	Top South East Declination Support Temperature
AT7	Bottom North East Declination Support Temperature
AT8	Bottom South East Declination Support Temperature
AT9	East Hour Angle Support Temperature
AT10	West Hour Angle Support Temperature

is in the Southern Hemisphere the Sun always transits across the Northern part of the sky being highest above the horizon in the Summer, and closest to the horizon in the Winter. We see from Figure 2.1a that the Sun illuminates the lower half of the North facing Declination support in July (Winter). Examination of Figure 2.1b shows that the Sun does not illuminate the South Declination support at all at this time. It is clear from this that the North facing beam will generally be warmer than the South facing beam during the day when solar illumination is greatest. The Hour Angle supports do not show similar differences in solar illumination.

### 2.2.2 Support Temperatures

Temperature data, shown in Figure 2.2 and Figure 2.3, suggests that while the North facing Declination supports are warmer during the day (as expected due to the Solar illumination), the South facing Declination supports may actually be warmer than the North facing side, particularly in the early evening. This may be due to the North facing side undergoing faster radiative cooling after sunset, since it isn't shielded by the antenna support buildings.

Figure 2.2 shows that the temperature differential between the North facing Declination supports and the South facing Declination supports varies both diurnally and, to a lesser extent, with season. Figure 2.3 shows the Hour Angle supports temperatures which do not show this same seasonal variation.

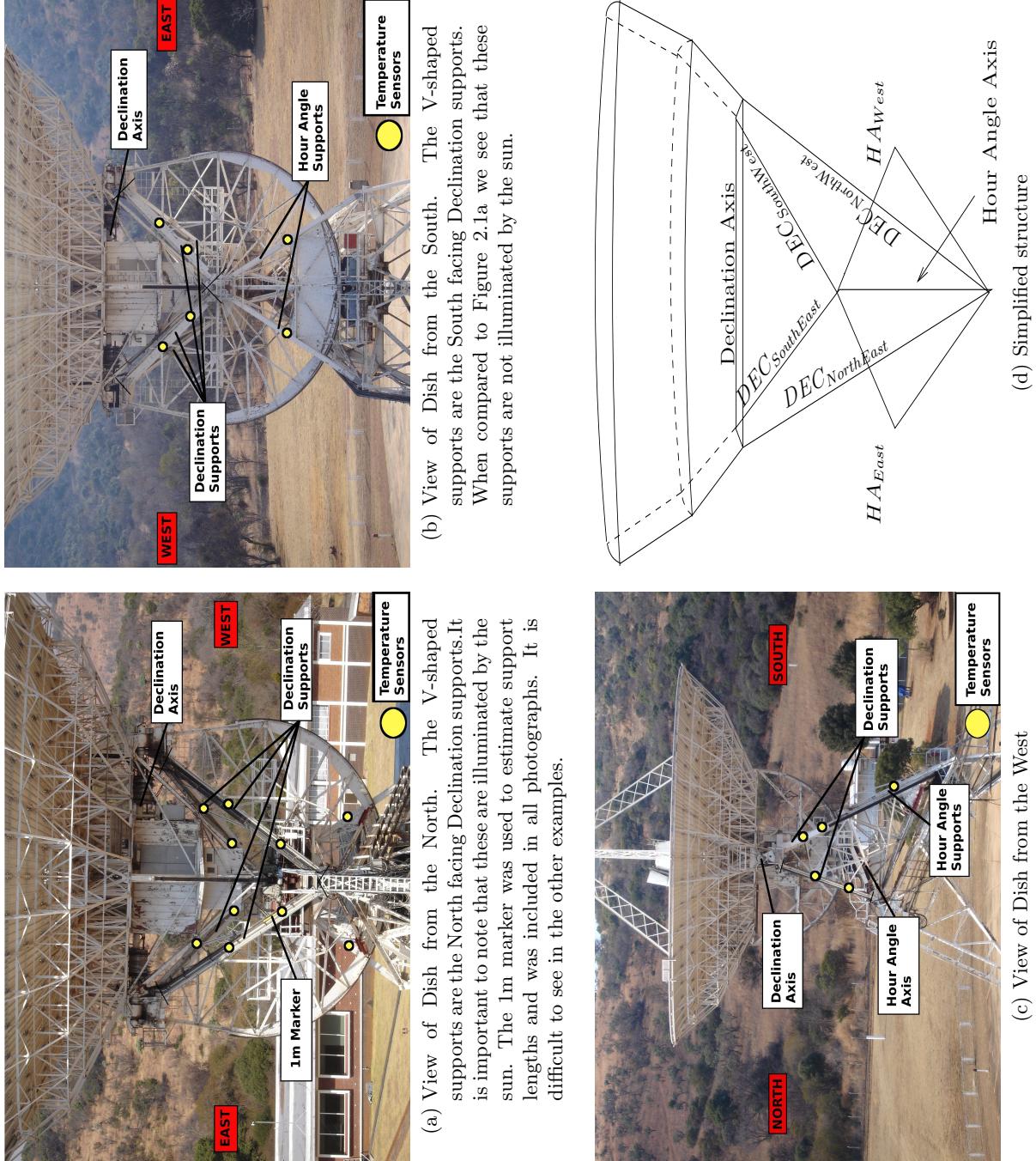


Figure 2.1: Orthogonal photographs of the HARTRAO dish taken in July 2006. These show the pattern of the solar illumination experienced by the North facing Declination support beams in Figure 2.1a. They indicate the orientation of the different supports in relation to each other and show the positions of temperature sensors.

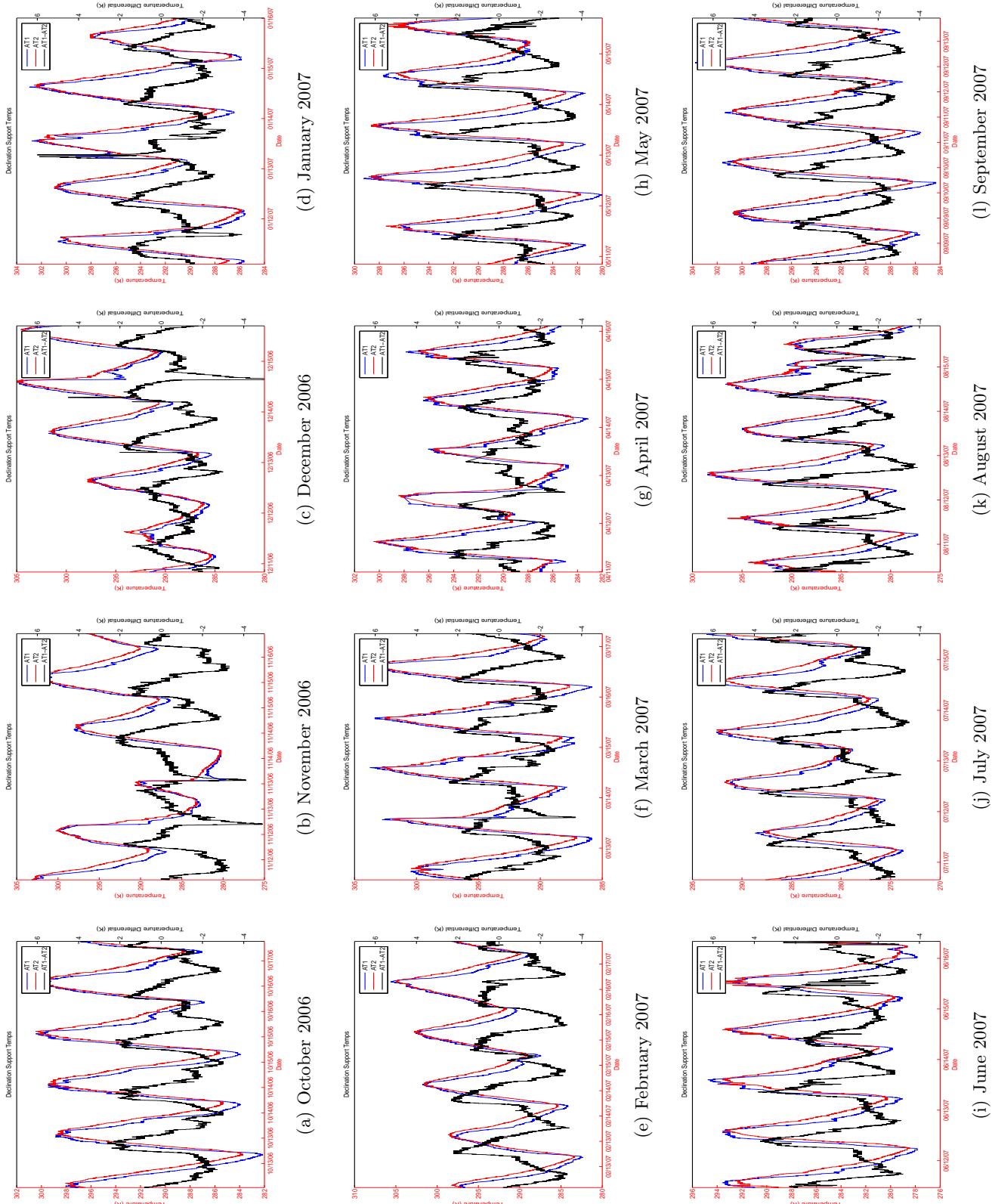


Figure 2.2: Temperature readings (5 days in each month) from the Declimation Support mounted thermocouples obtained throughout a complete annual cycle. These figures show the variation in the thermal loading of the supports during the different seasons. AT1 is the temperature at the top of the North West Declimation support and AT2 is the corresponding temperature at the top of the South West Declimation support.

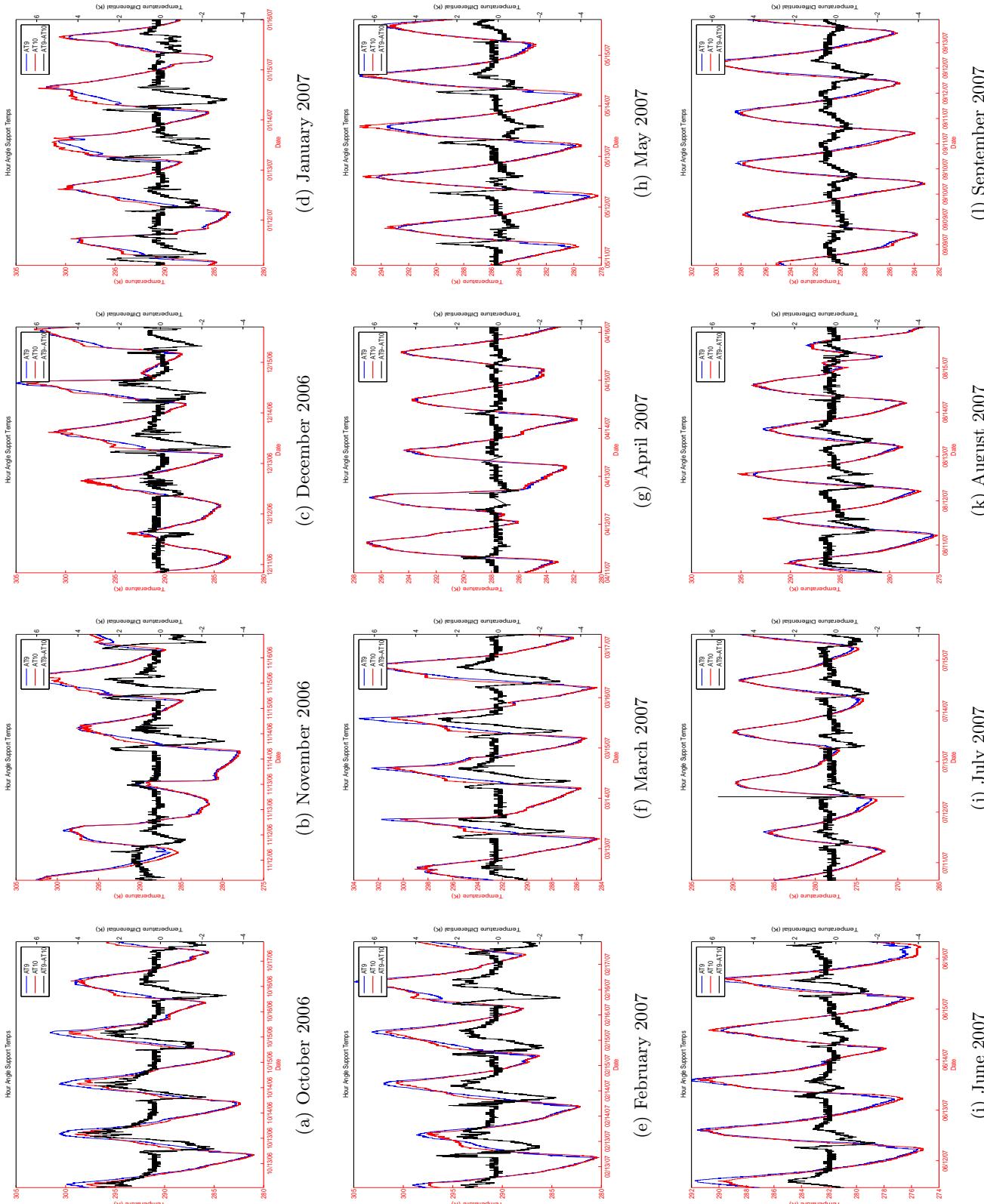


Figure 2.3: Temperature readings from the Hour Angle Support mounted thermocouples obtained throughout the year. AT9 is the East Hour Angle support temperature and AT10 is the West Hour Angle support temperature.

## 2.3 Previous Models

Previous measurement of the Declination and Hour Angle pointing offsets were done by Dr. George Nicolson prior to this project, and resulted in the development of the existing heuristic thermal model. The model is a predictive thermal correction that is dependent on day of year and time of day and is shown in Figure 2.4. The model was derived heuristically by examining pointing offsets throughout the year, however it did not use temperature measurements of the structure. These pointing offset measurements were made before extensive modifications were undertaken on the dish surface and counterbalance, which resulted in a changed mass distribution. Depending on the cause of the thermal pointing offsets they may have been effected by the changed mass distribution.

The predictive model in Figure 2.4 shows that the thermal Declination pointing offset was more pronounced in Winter than in Summer. It shows that the greatest Declination offsets were likely to be found during the day time between 07h00 and 15h00. This is similar to the pattern observed with the maximum temperature differential between North and South Declination supports (see Figure 2.2), both diurnally and seasonally.

This similarity provides a basis to investigating temperature differentials as a starting model of the thermal pointing offset experienced by the HARTRAO antenna.

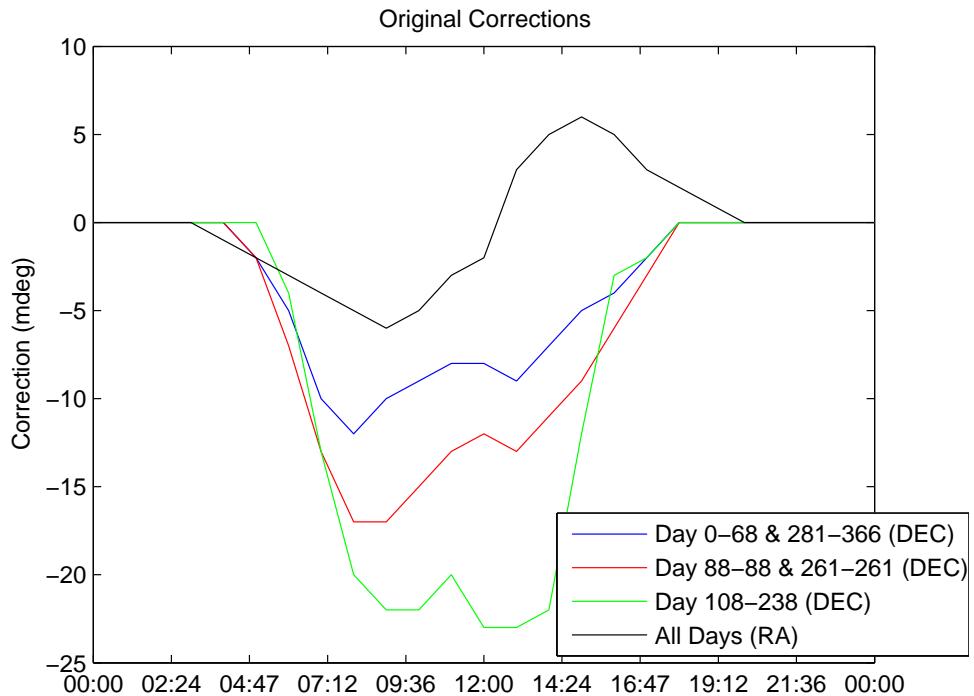


Figure 2.4: The original pointing error correction model derived by Dr. George Nicolson.

## 2.4 Overview

This thesis begins with a presentation of case studies in Chapter 3. These are all applicable to modelling the thermal pointing offset of the HARTRAO telescope. The case studies explore the different modelling methods used in the literature, and explain the relevance of these modelling techniques to the HARTRAO investigation.

Chapter 4 presents a theoretical overview of the different modelling techniques used in this study, as well as the general theory behind Finite Element Analysis (FEA). FEA would have been expensive and costly to undertake and, as such, was not investigated as a possible methodology, however, it is the most commonly used method in similar previous studies and therefore deserves mentioning.

The data capture and reduction techniques are explained in Chapter 5. It begins by giving an overview of the instrumentation used as well as an outline of the observation strategy. The derivation of pointing offset measurements from radio telescope observations of point sources is explained, and the data reduction strategy is outlined in terms of flow diagrams of the code. Chapter 5 concludes by presenting the results of the unprocessed pointing offset measurements.

Chapter 6, the penultimate chapter of this part of the thesis, describes the results of the modelling techniques outlined earlier in the thesis. The effectiveness of different models are examined in terms of both Right Ascension and Declination pointing offsets.

Conclusions are then drawn in Chapter 7, and ideas for future research are presented. I touch on the relevance of these results for *MeerKAT*.

# Chapter 3

## Case Studies

Detailed discussion of the effect of telescope pointing error compensation for thermal structural distortion is not commonly found in the mainstream scientific literature. It is rather found in technical reports (Condon and Constantikes, 2003; Nothnagel et al., 1997) and telescope viability studies (Von Hoemer, 1975). These technical memoranda are fairly brief overviews, aimed primarily at minimising the thermal effect on pointing from the outset by means of material choice, enclosing antennas and structural design. There are a few examples of actual implementation of a correction post-construction (Condon, 2003a; Condon and Constantikes, 2003; Condon et al., 1993; Condon, 2003b). The following sections represent relevant case studies for a selection of telescopes. The unexpected similarity between Finite Element Analysis<sup>1</sup> (FEA) and simpler models is emphasised.

### 3.1 Green Bank Telescope

A large number of technical reports from the 100 m Green Bank Telescope (GBT) are available (Condon, 2003a; Condon and Constantikes, 2003; Condon et al., 1993, 1989; Condon, 2003b). These explore attempts made at the GBT to allow observations to be made at up to 117 GHz. Condon (2003a) mentions that if the telescope is much larger than the observing wavelength (i.e  $D > 10^4\lambda$  resulting in a small beam), *time-dependent mechanical distortions* are likely to degrade its performance. The reports note that astronomers may have to observe calibration sources periodically in order to maintain the required pointing and tracking tolerances (dictated by the scientific requirements) for aperture efficiency and gain accuracy and consistency (Condon et al., 1993). It is noted that the high sensitivity of the GBT means that a suitable calibrator is typically never more than 0.03 radians from any suitably strong point source. Since HARTRAO is a comparably small antenna, with low sensitivity at 22 GHz, there are few nearby calibrators. Periodic calibration is thus not an option and suitably accurate dead reckoning is required.

---

<sup>1</sup>See Section 4.8 for a detailed discussion of this technique

### 3.2 MERLIN Radio Telescope in Cambridge

Bayley et al. (1994) studied the 25 m MERLIN Radio Telescope based in Cambridge. The telescope design is different to HARTRAQ, being an ALT-AZ mounting, however a number of principles are transferable. Of primary importance are the structural supports of the telescope as shown in Figure 3.1. The elevation configuration is structurally similar to the Declination axis of the HARTRAQ antenna shown in Figure 2.1d.

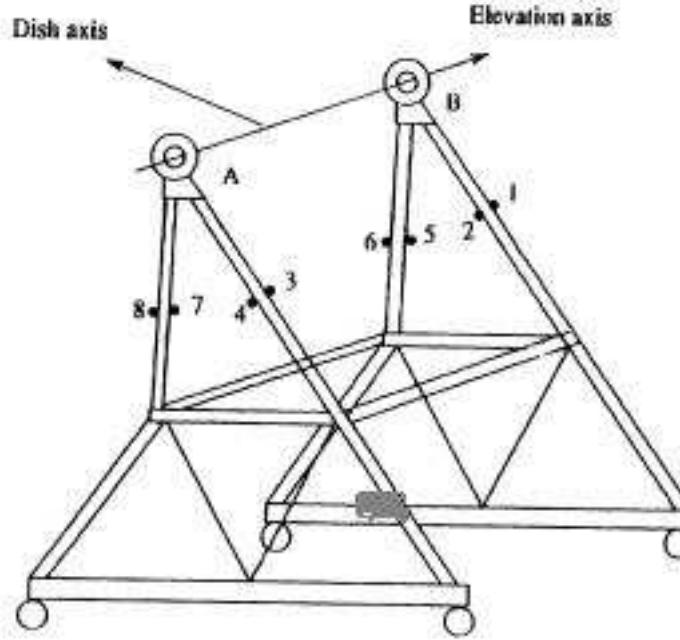


Figure 3.1: The support structure of the MERLIN telescope based at Cambridge together with the positions of the thermistors (Bayley et al., 1994).

The magnitude of the pointing offset observed for the Altitude and Azimuth axes differs, an effect that is observed on the HARTRAQ antenna Declination and Right Ascension pointing offsets. This paper was able to investigate the use of FEA modelling of the thermal effects on structural distortion, due to the availability of mechanical and structural data (the investigation was carried out immediately after the telescope's commissioning). The result of using FEA predictions in the pointing model, was an improvement of the pointing deviation to approximately 2 mdeg RMS. A simpler model based only on temperature differentials agreed with the FEA predictions to within 0.5 mdeg. A similar agreement between FEA and a simpler model is found on the OVRO Telescope discussed in Section 3.3. This suggests that FEA on simple frame structures such as this may not be necessary. This is a promising result for this study of the HARTRAQ antenna, as FEA was not investigated as a methodology due to cost and implementation difficulties.

### 3.3 Thermal Effects on the Leighton 10 m antennas at Owens Valley Radio Observatory

A detailed study of the backing structure (using FEA) of the Owens Valley Radio Observatory (OVRO) 10 m antennas (Lamb and Woody, 2002) brought attention to the fact that temperatures of the backing structure can effect the focus of the telescope, contributing to other related problems. The investigation is primarily aimed as a pathfinder towards potential problems and solutions for sub-millimetre array telescopes and the emphasis is on using the Leighton 10 m antennas as potential models for these future antenna arrays.

The paper discusses temperature gradients in materials stating that they are a function of the material and the optical and infrared emissivities of the surfaces. This is not relevant to the HARTRAO thermal problem, however these are important aspects to be taken into consideration during the construction of *MeerKAT*. Reference is made to dependence on solar illumination, wind and ambient temperature variations which is what this study will focus on.

The problem of an accurate temperature model for the entire structure means that detailed temperature measurements are required. OVRO used 76 thermistors (YSI 44007) mounted on aluminium blocks using thermally conductive epoxy, to obtain temperature readings for modelling the structural deformation. The temperature measurements from the thermistors had an uncertainty of 0.07 °C with a maximum uncertainty of 0.2 °C. The acquisition of temperature data is done in sets of 1000 samples which are averaged to reduce noise and results in measurements that are better than 1 bit on the 12-bit Analogue to Digital converter. The data is recorded every 2 minutes. The large number of thermistors alone is indicative of the possible expense and difficulty involved in implementing FEA as a method of modelling the pointing offsets.

The structure was approximately isothermal at night, but during the day significant gradients developed with a maximum thermistor temperature difference of 12 °C observed. Temperature change may occur as quickly as  $1.5 \text{ C}.\text{min}^{-1}$ . The conditions at HARTRAO are similar with maximum temperature differences of approximately 10 K and maximum temperature changes of  $0.5 \text{ C}.\text{min}^{-1}$  as shown in Figure 2.2. It was noted that the lowest temperatures on the structure are generally close to ambient air temperature, however they are occasionally lower. This was explained as being partly due to radiative cooling to the sky, and partly due to the 200 m distance to the air temperature sensor from the telescope.

The paper mentions that other telescopes have shown strong astigmatism under solar heating but that this does not appear to be a problem at OVRO. This was not investigated



Figure 3.2: Picture of the OVRO 10m Leighton Array

at HARTRAO in this study.

A beam-element model (i.e no rotational degrees of freedom at the structural nodes) was constructed using *Algor* (a FEA program) and is illustrated in Figure 3.3a. The results of the FEA are shown in Figure 3.4. This shows good correlation between the simple linear model (using mean temperature gradients) and the FEA model, although the conclusion is that since the complete FEA solution takes only a few seconds, it is preferable to use the full FEA model. Bayley et al. (1994) notes a similar agreement between FEA and a much simpler model for the MERLIN telescope as described in Section 3.2. This is a promising result for HARTRAO since the models explored here do not include FEA. However since these results are based on modelling the backing structure it does suggest that the pointing offset at HARTRAO may be caused by thermal gradients in areas other than the Declination and Hour Angle support struts.

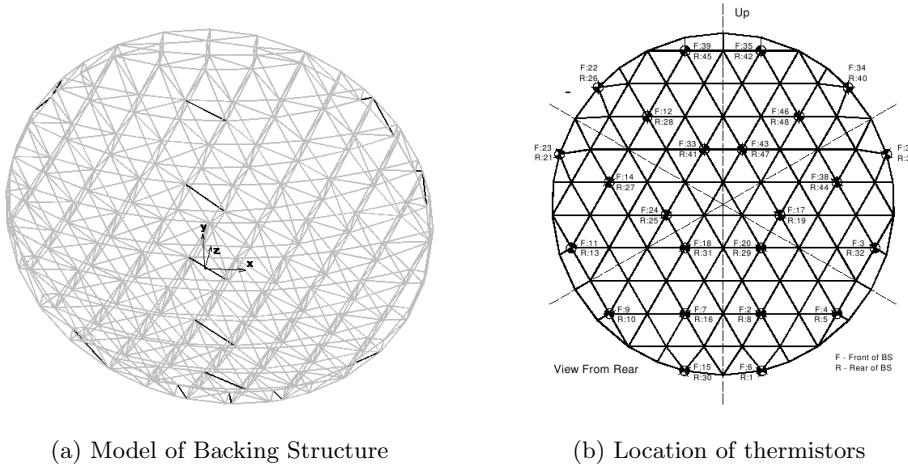


Figure 3.3: Model of the backing structure used in the FEA calculations for the OVRO telescope as well as the location of the thermistors. The dark struts are ones which break the lateral symmetry of the dish (Lamb and Woody, 1998)

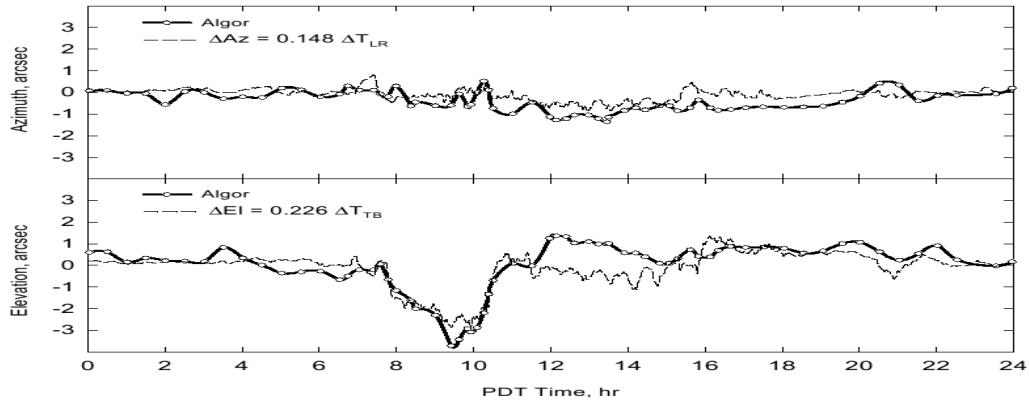


Figure 3.4: Comparison between the full FEA prediction for the pointing offset and an estimate based on a linear dependence on the temperature gradient (Lamb and Woody, 2002).

### 3.4 The 45 m Nobeyama Telescope

Ukita (1999) deals extensively with the pointing model of the 45 m Nobeyama telescope in Japan, which uses 16 parameters. This model takes into account well understood pointing offset effects such as encoder error, axis misalignments and gravitational deformation and is similar to the VLBI pointing model used at HATRAO. The Nobeyama thermal pointing model was calibrated using sources which had a signal-to-noise ratio larger than 10. A similar criteria was used to separate out poor data in this study of HATRAO. The paper explored the effect of wind on the pointing and separated data using wind speed as a criteria in certain cases. While wind effects were considered negligible in the HATRAO study, it is interesting to note that Ukita (1999) estimated that the wind contribution to the pointing offset was

roughly proportional to the square of the wind velocity.

Since the Nobeyama telescope has an ALT-AZ mount, the results were not directly applicable to HARTRAO, however they provided insight into the issues. The paper provides a good argument for the importance of good, reliable pointing. The example given is that a pointing error of only 1/5 of the Half Power Beam Width of a Gaussian beam results in a loss of sensitivity of 10 % (Ukita, 1999). This particular telescope operates with a maximum observation frequency of 115 GHz, corresponding to a beam width of 15 arcseconds (or  $\approx 4.1$  mdeg). This makes it necessary to reduce errors to below 0.83 mdeg to maintain a loss of sensitivity of less than 10 %. For HARTRAO the high end is 22 GHz so we would expect a less challenging pointing requirement since the beamwidth would be larger. The 2.5 cm receiver has an estimated beamwidth of 80 mdeg (or  $\approx 290$  arcseconds) so we would require the pointing errors to be below 16 mdeg to avoid a loss of sensitivity of more than 10 %. The 1.3 cm (22 GHz) receiver, with a HPBW of 40 mdeg, would need slightly better pointing still ( $<8$  mdeg) to avoid the 10 % loss of efficiency. Subsection 2.1.1 mentions that the generally acceptable pointing offset is 1/10th of the beamwidth. For the new 22 GHz receiver, this dictates that the pointing offset at HARTRAO be less than 4 mdeg.

### 3.5 The HARTRAO Study

In general the literature attempts have been remarkably successful which is promising for this study. Even relatively simple models have produced surprisingly good results.

A particularly reassuring aspect of the past attempts to model thermal effects, is the general agreement between simple models and the more comprehensive FEA. Since we were not able to carry out a FEA on the HARTRAO structure we focused on investigating the effectiveness of simple models as a starting point.

## Chapter 4

# Modelling the Problem

This chapter provides general descriptions of the modelling techniques used in this investigation. A brief summary of Finite Element Analysis is included, despite not being used in this study, since the technique is used in the majority of studies in the scientific literature.

### 4.1 VLBI Pointing Model

A thorough description of the physical effects modelled by the VLBI Model is provided in Appendix B. The VLBI pointing model is parameter based and has the form shown below.

$$\begin{aligned}\Delta X = & P_1 - P_2 \cos \phi \sin X \sec Y + P_3 \tan Y - P_4 \sec Y + P_5 \sin X \tan Y \\ & - P_6 \cos X \tan Y + P_{12}X + P_{13} \cos X + P_{14} \sin X + P_{15} \cos 2X \\ & + P_{16} \sin 2X\end{aligned}\tag{4.1}$$

$$\begin{aligned}\Delta Y = & P_5 \cos X + P_6 \sin X + P_7 - P_8(\cos \phi \cos X \sin Y - \sin \phi \cos Y) \\ & + P_9Y + P_{10} \cos Y + P_{11} \sin Y\end{aligned}\tag{4.2}$$

where  $X$  and  $Y$  represent the Hour Angle and Declination respectively.

Each term models a different pointing effects such as gravitational deformations, axial misalignments, encoders offsets and other static causes of pointing offsets.

### 4.2 Thermal Expansion

A fundamental relationship that is needed for a physical model of the structure, is how metal expands under different thermal loads. This is given by

$$\alpha_{Steel} = \frac{1}{l} \frac{dl}{dT}\tag{4.3}$$

where  $l$  is the length of the metal.

The fractional expansion of metal is thus related to the temperature difference ( $\Delta T$ ) through the expansion coefficient  $\alpha$ . The equation can be rearranged into

$$dl = \alpha_{Steel} \times \Delta T \times l \quad (4.4)$$

### 4.3 Simple Physical Model of the Declination Supports

I decided that a simple model would be worthwhile after reading a technical report of the Greenbank Telescope (Condon and Constantines, 2003) where a simple isosceles triangle model is used to predict the approximate magnitude of the pointing errors. This required dimensions of the support structure itself. Unfortunately these were no longer available at the observatory, and previous attempts to obtain blueprints had proved unsuccessful.

I decided that the best way of proceeding would be to get rough dimensions by taking photographs of the structure together with a 1 m ruler for perspective and distance estimates. The photographs were taken by raising ourselves up on the cherry picker to (roughly) the same height as the dish (as determined by a builders level) and obtaining photographs from the North, South and West of the dish with the 1 metre measuring stick in each of the photographs. These photographs (from the various orthogonal views) and the 1 m ruler were then used determine angles and lengths of the various beams in question. A selection of the photographs is shown in Figure 2.1a, 2.1b and 2.1c.

It is immediately obvious that the measurements will be strongly influenced by human perspective as it is difficult to judge accurately from a photograph where a joint ends. For certain beams it is possible to obtain dimensions from the North view and to compare them with similar measurements from the West view. These independent measurements did agree to within approximately 25 cm, however this required careful examination to correlate joints between the two photographs. The estimated dimensions are included in Table 4.1 and simple structural drawings of the structure are shown in Figure 4.1.

Table 4.1: Dimensions of the HATRAO support structure relevant to the simple physical model.  $\alpha_{Steel}$  is the expansion coefficient of steel.

Dimension	Value
$l_{North}$	8.9m
$l_{South}$	7.5m
c	1.2m
$\alpha_{Steel}$	$1.2 \times 10^{-5} \text{m.K}^{-1}$

Examination of Figure 4.1c and using Equation 4.4 leads us to Equation 4.5 and

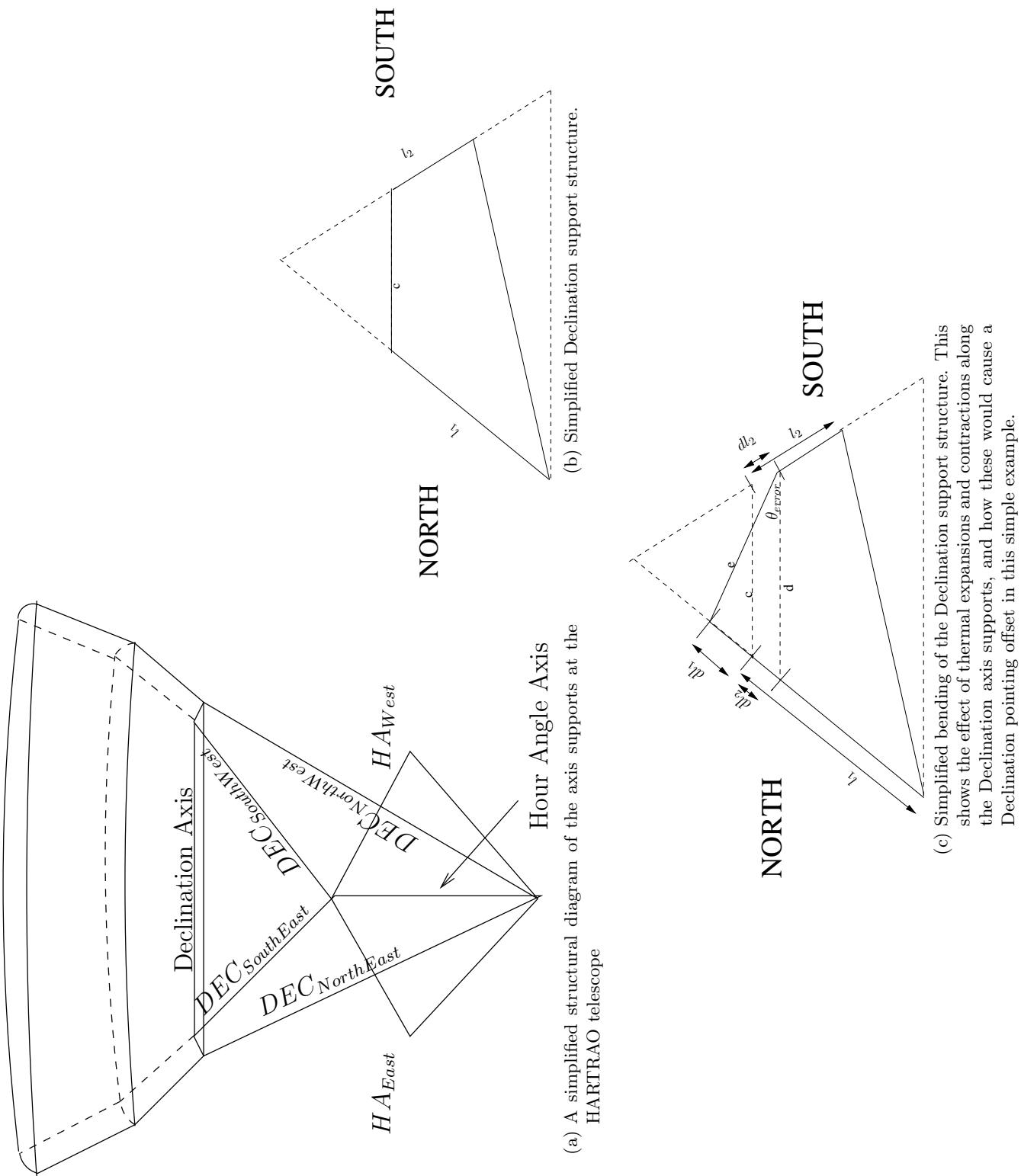


Figure 4.1: Structural models of the HARTRAO Telescope

Equation 4.6.

$$dl_1 = \alpha_{Steel} \times dT_1 \times l_1 \quad (4.5)$$

$$dl_2 = \alpha_{Steel} \times dT_2 \times l_2 \quad (4.6)$$

1.  $l_1$  and  $l_2$  are the lengths of the support beams on either side
2.  $dT_1$  and  $dT_2$  are the difference in temperature of the support beams from the standard reference temperature i.e  $T_1 - T_{ref}$  and  $T_2 - T_{ref}$
3.  $dl_1$  and  $dl_2$  are the relative expansion/contraction of the beams relative to their state at the standard temperature.

On examining Figure 4.1c we see that for small angles (such as those expected here) the following approximation is valid ( provided  $c \approx d \approx e$ )

$$\theta_{error} = \frac{dl_1 + dl_2}{c} \quad (4.7)$$

We can use Equation 4.7 (together with Equation 4.5, Equation 4.6 and Table 4.1) to create a simple model of the Declination pointing offset.

## 4.4 Correlation Coefficients

Correlation coefficients are a measure of the degree of correlation between two variables, calculated using the following equation

$$R = \frac{N \sum xy - \sum x \sum y}{\sqrt{[N \sum x^2 - (\sum x)^2] [N \sum y^2 - (\sum y)^2]}} \quad (4.8)$$

This equation produces a value in the interval  $(-1, +1)$ . A value of  $+1$  indicates a perfect positive linear relationship between the variables and a value of  $-1$  indicates a perfect negative linear relationship between the variables. A value of  $0$  indicates that there is no correlation between the variables.

If there are  $M$  variables, an  $M \times M$  correlation matrix can be produced made up of the correlation coefficients of the  $M$  variables. Element  $R_{ij}$  of the matrix is the correlation coefficient between variable  $i$  and variable  $j$ . The correlation coefficient between a variable and itself will be  $1$  (since it is a perfect positive linear relationship) and thus the diagonal of a correlation matrix contains only  $1$ 's.

Typically,  $R > 0.5$  (or  $R < -0.5$ ) shows a good correlation, although this depends on the instrumentation used to collect the data, with higher correlation coefficients expected from higher precision instrumentation.

Correlation matrices are often used as a starting point to investigating relationships between variables. The related covariance matrix serves as a basis for other statistical techniques such as *Principal Component Analysis* and *Factor Analysis*. These techniques can be used to reduce the number of variables to a new set of variables that still contain most of the information.

## 4.5 Principal Component Analysis

Principal Component Analysis (PCA) is used as a method of reducing a data set to a lower dimension (Schlens, 2005). This is often used to reveal underlying relationships between sets of variables.

A full description of the mathematics behind PCA is not relevant to this study, and is readily available in the literature (Jolliffe, 2002; Schlens, 2005), however I will provide a brief summary. PCA takes observations of a set of variables (expressed in a vector space) and changes the basis of this vector space. The new basis is chosen so as to minimise noise by maximising the *Signal to Noise Ratio* (SNR) and to minimise the data redundancy.

The *variance* is defined as

$$\sigma_A^2 = \langle a_i a_i \rangle_i \quad (4.9)$$

where  $\langle \rangle_i$  denotes the average of the values over the  $i$  indexed values. The variance relates to the SNR.

The *covariance* between A and B is given by

$$\sigma_{AB}^2 = \langle a_i b_i \rangle_i \quad (4.10)$$

with high values indicating high redundancy between variables. An important result from the definition of covariance is that  $\sigma_{AB}^2 \geq 0$  and  $\sigma_{AB}^2 = 0$  only when there is no correlation between the variables.

The *covariance matrix* is a square symmetric matrix where the diagonal terms are the variance of the variables, and the off-diagonal terms are the covariance between the variables. It is used as a starting point for many statistical investigations (including the correlation matrix described in Section 4.4). We can define the covariance matrix of the data as  $C_X$ . As mentioned previously, PCA seeks to find a new basis so as to maximise the SNR (maximise the variance  $\sigma_A^2$ ) and to minimise the data redundancy (minimise the covariance  $\sigma_{AB}^2$ ). A correlation matrix ( $C_Y$ ) constructed from a perfectly optimised data set would therefore show zero covariance (since zero is the minimum achievable covariance) between variables and therefore the off-diagonal terms of  $C_Y$  would be zero. This is the definition of a diagonal

matrix, and minimising the covariance can then be redefined to the problem of diagonalising  $C_X$ .

PCA assumes that all basis vectors  $(p_1, \dots, p_m)$  are orthonormal (i.e  $p_i \cdot p_j = \delta_{ij}$ ) and thus that the matrix  $P$  is orthonormal. PCA secondly assumes the directions with the greatest variance, which maximises the SNR for reasons outlined earlier. The algorithm used by PCA is as follows

1. Select a direction in the  $m$ -dimensional space along which the variance of the variables is maximised. This is saved as  $P_1$ , the first principal component
2. Find another direction along which the variance is maximised, however this direction should be perpendicular to the first direction due to the requirement of orthonormality
3. Repeat this process until the  $m$  vectors are completed

From the above it is important to notice the following restrictions on PCA

- PCA will only reveal linear relationships between variables
- The Principal Components are orthonormal
- The Principal Components will be ordered in decreasing variance

## 4.6 Linear Fitting of Weighted Inputs

Simple models for a phenomenon can be constructed if the cause is thought to be understood. In the case of the HATRAO pointing offset, we suspect that differential heating of the axis supports may cause deformations in the supports, resulting in axis movements which cause the pointing offsets.

If temperatures of the axis supports are given by  $AT_1, \dots, AT_n$ , then a simple model involves assigning variable weights to these temperatures and adding these weighted temperatures.

$$\text{Model Pointing Offset} = \sum_{i=1}^{i=n} a_i AT_i \quad (4.11)$$

A suitable fitting technique can be used to obtain the best-fit weighting vector  $a_i$ . If the support temperatures do play a role in causing pointing offset, this would produce a model of the pointing offset which uses the support temperatures as inputs.

This method is often recommended as a replacement for FEA (see Section 4.8) when the complexity of FEA is not justified or when FEA is difficult to implement, as is the case at HATRAO.

## 4.7 Neural Networks

Neural Networks are loosely derived from biological systems, with the system being defined by the relationships between all the elements. They have certain advantages over other modelling systems in that non-linear effects are easily taken into account, the development of the model is relatively quick and easy to understand, and a thorough physical understanding of the entire system is unnecessary. All that is required to roughly model the HARTRAO pointing offset using neural networks, are pointing offset measurements and corresponding axis support temperatures. The disadvantage is that since little physical understanding is required, any physical change in the structure requires the model to be retrained. This should not be a significant disadvantage when compared against most other methods, apart from a Finite Element Analysis (see Section 4.8) of the structure.

The fundamental building block of a neural network is the neuron which is introduced in Figure 4.3a. As shown in Figure 4.3a, a scalar input,  $p$ , is multiplied by a weighting matrix and a bias is added. This is then used as the argument to a transfer function. A number of different transfer functions can be utilised depending on the application and the training method. MATLAB provides a wide variety but as an introduction, shows the transfer functions in Figure 4.2.

A typical neural network will involve an input matrix, a neural network consisting of one or more neurons as well as possibly more than one layer of neurons, and the corresponding output vector.

In the simplest case, the output of the transfer function is fitted to observations (using non-linear fitting techniques) by changing the weighting matrix. More complicated systems may involve using more neurons before the output as in Figure 4.3e. The actual fitting occurs through a minimisation of the difference between predicted and observed errors as shown in Figure 4.3b.

### 4.7.1 Overfitting

One of the potential problems associated with neural networks is *overfitting*. This results in the neural network fitting extremely well to the training data set, but not when applied to new (unseen) data.

Two major techniques are used to overcome this problem (Doan and Liong; The Mathworks, 2005). These are

1. Early stopping method
2. Bayesian Regularisation

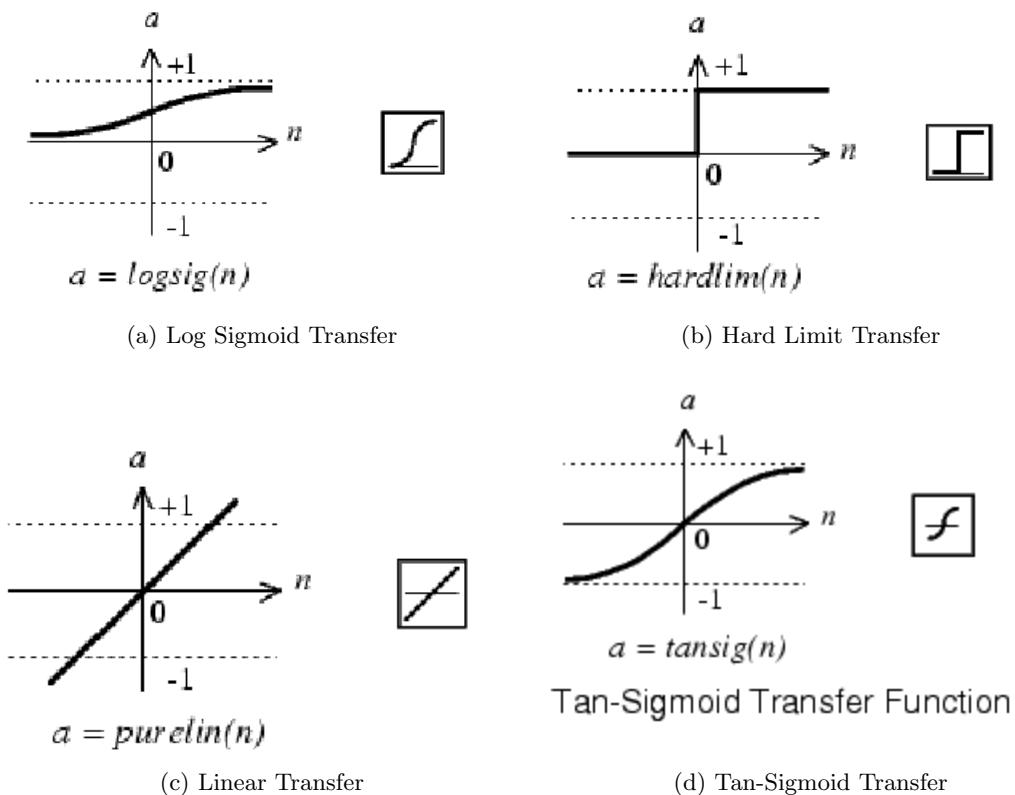
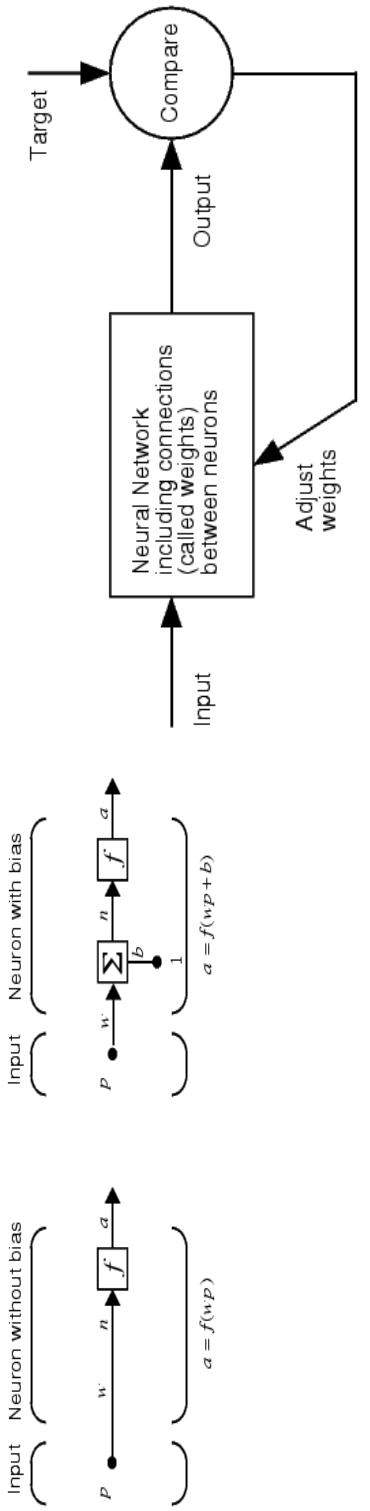


Figure 4.2: Different transfer functions used in neural networks (MATLAB R14)

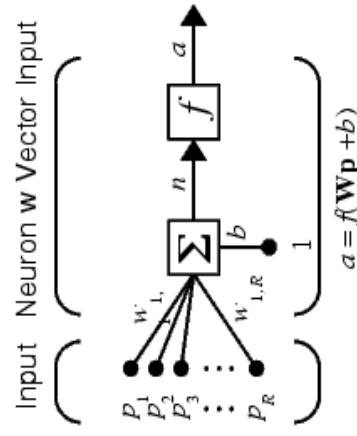
The early stopping method was employed in this data analysis. This requires that the data be divided into 3 sets. These are known as the Validation, Testing and Training sets respectively. The neural network is trained on the Training set and updates the weights and biases. The neural network monitors the error between the model predictions and observations for the Validation set throughout the training. In the beginning of the training, the error will decrease as the fitting improves. When the network begins to overfit to the Training data however, the Validation set error will typically begin to rise. If this occurs for a specified number of training iterations then the training is stopped.

The Testing data set is not used in the process but its error can be compared to the error of the Validation training set. If the data has been divided poorly, the two errors will not be consistent with each other. This can be remedied by redividing the data to contain evenly spread points across all the sets.



(a) The figure on the left shows one without bias input while the one on the right includes bias input.

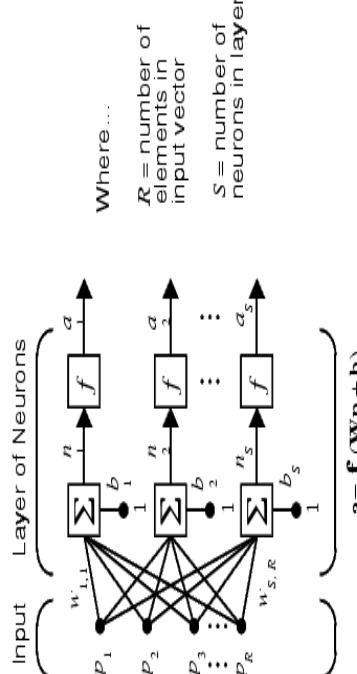
(b) Depiction of neural network training method. This shows the feedback of the output and the minimisation routine.



(c) An input vector being used as the input to a neuron.



(d) A one layer neuron neural network.



(e) A multi layer neuron neural network

$$\begin{aligned}
 a^1 &= \mathbf{f}^1(\mathbf{L}\mathbf{W}_{1,1}\mathbf{p} + \mathbf{b}_1) \\
 a^2 &= \mathbf{f}^2(\mathbf{L}\mathbf{W}_{2,1}\mathbf{f}^1(\mathbf{L}\mathbf{W}_{1,1}\mathbf{p} + \mathbf{b}_1) + \mathbf{b}_2) \\
 a^3 &= \mathbf{f}^3(\mathbf{L}\mathbf{W}_{3,2}\mathbf{f}^2(\mathbf{L}\mathbf{W}_{2,1}\mathbf{f}^1(\mathbf{L}\mathbf{W}_{1,1}\mathbf{p} + \mathbf{b}_1) + \mathbf{b}_2) + \mathbf{b}_3)
 \end{aligned}$$

Figure 4.3: Diagrammatic examples of neural network nomenclature and structures (MATLAB R14).

## 4.8 Finite Element Analysis

Finite Element Analysis is the primary tool used in the majority of thermal correction models. The idea is that the antenna can be represented by linked elements, each of which has an associated set of equations. Examples of these elements include triangles, hexagons, cubes etc.. The finite element procedure involves three steps:

- Pre-processing
- Solving
- Post-processing

In pre-processing the finite element model is constructed. This model should replicate the important parameters of the antenna, such as material and structure, and is generally obtained from existing blueprints.

A meshing procedure is then applied, where the model is broken up into smaller constituent elements, each with their own associated set of equations. Nodes are then defined as positions of important points on these elements, and are used to calculate displacements and other useful features of the model. Constraints are then placed on particular nodes to simulate the real model more accurately. Certain nodes may, for example, have movement restricted in certain axes to simulate fixed points on the structure.

The solving step follows. External loads, such as forces, temperatures, pressures etc. are applied to the finite element model and resulting displacements of nodes are then iteratively updated. This iterative process continues until a minimum output is reached and the nodes are consistent within themselves. The final displacements are then noted.

Post-processing is then applied to the resulting displaced nodes. This allows the user to visualise the effects of the initial loads placed on the model. In the case of thermal deformations of an antenna, it would be possible to place a set of temperature loads on the model, and then to visually inspect the effect this would have on the structure as a whole.

This study does not use FEA primarily due to a lack of accurate telescope dimensions. I did investigate various options to obtain the necessary dimensions using measurement techniques. Prof. Heinz Ruther of the University of Cape Town Geomatics Department suggests that the most suitable technique would be to use a laser-scanner. This would allow dimensions accurate to 6mm RMS to be obtained, but would be expensive to implement.

At this stage I decided not to pursue this option further as the HARTRAO pointing requirements could probably be satisfied with a less sophisticated model which would be easier and less expensive to realise.

## Chapter 5

# Data Capture and Reduction

The calibration of pointing models requires the observation of the position of a catalogue of strong point sources.

This chapter describes the instrumentation used during these observations and outlines the experimental design. It includes a description of how pointing offsets were obtained by fitting different beam profile functions to point source scan data. I outline the layout of the code that reduces the observations to pointing offset measurements and explain how data quality is ensured. I conclude the chapter by including the resulting pointing offsets and describing the refitting of the VLBI pointing model using these new results.

### 5.1 The HARTRAO Antenna

The Hartebeeshoek Radio Astronomy Observatory (HARTRAO) dish is located 80 km west of Johannesburg, South Africa. It was initially built by NASA as a tracking station for probes sent beyond the Earth's orbit. The initial surface was made of aluminium mesh and operated at a frequency of 960 MHz (30 cm). In the mid-1960's the surface was upgraded to aluminium panels, and the frequency changed to 2300 MHz (13 cm). The antenna was handed over to the South African government in 1974, and was transformed into a radio telescope. Subsequent upgrades of receiver systems and dish surface, have led to the current configuration with receiver systems at 18 cm, 13 cm, 6 cm, 5 cm, 4.5 cm, 3.5 cm, 2.5 cm and 1.3 cm.

The dish has an equatorial mount, and features a Cassegrain optical system with a 25.9 m parabolic primary reflector. The receiver feeds are mounted in a feed cabin on the primary reflector, close to the optical axis of the secondary reflector.

### 5.2 Telescope Control

The automatic observation system at HARTRAO accepts a list of objects for a particular observation schedule. The objects are typically entered together with their J2000 (or B1950)

mean coordinates and observation strategy, and the observations are given a priority rating. It was anticipated that we would need to experiment with this priority rating in order to obtain sufficient observations without 'locking' other observations out of the scheduler. Fortunately this did not materialise as an issue.

An example of a typical object input to the observation file is given in Listing 5.1. This is an extract from *pointing.cat*. The keywords defined in the catalogue are referenced in the scheduler file (included in Listing 5.2) which sets up other observation properties such as wavelength, start and end time and the type of scan. The HARTRAO website (HARTRAO, b) includes more information on different setup options.

```
OBJECT 0023-26, B0023-263
// Good for pointing at X-band, but weakish
COORDSYS EQUATORIAL
EQUINOX 1950.0
RA      00 23 18.914
DEC     -26 18 49.25 // VLA
ENDOBJ
```

Listing 5.1: Typical Observation object in the observation catalogue *pointing.cat*. The layout is self-explanatory, defining the coordinate system and equinox used as well as the position of the source in the defined equinox. It specifies a keyword (0023-26) used by the scheduler file (see Listing 5.2) to identify which objects will be included in observations

```
SETUP
OBSERVER Jonathan Quick // observer name
PROJECT Refining the New Pointing Model using X-band // project \
title
PROPOSAL 2005.99 // proposal number
OBSLOCAL jon // local \
observed username
STRTDATE 2006 07 18 // date on which observations should \
start
STRTTIME 00 00 00.0 // time at which observations begin
ENDDATE 2007 12 31 // date on which observations end
ENDTIME 00 00 00.0 // time at which observations end
CATALOG pointing.cat // reference to the catalogue containing \
objects that should be observed
PRIORITY FOREGROUND // priority level of the survey
ORDER RANDOM // order of observations \
through the catalogue
OUTFILE Copley
CONF 3.5CM // observation \
wavelength
RESTFREQ 8400.000E6
INSTRUME DICKE // Dicke switching to remove \
atmospheric contaminations
SCANTYPE CROSSED // Cross scan observation
SCANDIST SN
SCANS 8
SCANANGL 0.0
ALTLIMIT 60.0 // angular elevation above which \
observations are made
```

```

WEATHER          CLEAR           //worst allowable weather conditions \
during which observations are made
ENDCONF
ENDSETUP

// Good for pointing at X-band
OBJECT 0316+41 // aka 3C84
USECONF 3.5CM
ENDOBJ
.....

```

Listing 5.2: Typical scheduler file used during these observations. This sets the observation wavelength, dates, objects and other conditions related to the observations. Important to note is the ALTLIMIT keyword which specifies that observations will only be made at elevations greater than 60°. This means that all observation will be within 30° of zenith minimising other pointing offset effects such as gravitational deformation. We limit observations to only occur during clear weather conditions. Keywords of objects to be used from the pointing catalogue are included below the ENDSETUP keyword- in this example only 0316+41 would be used. Further details are available on the HARTRAO website (HARTRAO, b)

In the scheduler input file shown in Listing 5.2 we see that the observation file is *pointing.cat* and the priority is given as *BACKGROUND*. We see the the scantype is defined as *CROSSED* and that Dicke Switching is being utilised.

## 5.3 Experiment Design

### 5.3.1 3.5 cm Receiver

The pointing observations were conducted using the 3.5 cm receiver. This receiver has a dual feed which permits Dicke-switched radiometry. I have concentrated on the left circularly polarised feed in this investigation.

### 5.3.2 Thermistors

Temperature was recorded using the AD592 IC. These IC's provide a current which is proportional to the absolute temperature. These have an accuracy of 0.5°C at 25°C (Analog Devices, a). Two were mounted on each of the four primary Declination axis supports at two different places approximately 2 m apart and at approximately the same position on each support. One thermocouple was mounted on each of the Hour Angle supports. The exact positions were not considered to be important at this stage.

The SCADA data collection unit recorded temperature readings every 1 minute for all ten of the IC's.

### 5.3.3 Description of Observations

The type of observations used are described by the *CROSSED* keyword in the observation file. A total of 4 scans along each axis are made for each observation to give a total of 8 scans per object. For each scan, coordinates are recorded in a multitude of systems. Each scan comprises 460 signal samples, with the associated time/position. These 8 scans are stored in a FITS file, a file format commonly used for astronomical observations.

During each scan one of the coordinates is kept fixed, while the other is varied to produce a cross-scan of the object. This is shown in Figure 5.1a. I have included the Right Ascension scan and the Declination scans separately in Figure 5.1b and Figure 5.1c, to highlight the difference in the movement between the two scans.

Hour Angle and Right Ascension are related through sidereal time. The local coordinate system of the HARTRAO antenna is the Equatorial Hour Angle and Declination. To maintain a fixed Right Ascension position the antenna is forced to move through Hour Angle as the sidereal time changes. This leads to inevitable drifts while attempting to keep the Right Ascension position fixed. We see the slight drift in the Right Ascension in 5.1b. For the Right Ascension scan the antenna does not need to move to maintain a fixed the Declination. The changes in Declination that we see in 5.1c are only quantisation errors of the encoder.

As each scan is completed the coordinates of the maximum signal are recorded and used for the following scan so that the telescope can 'centre in' on the observed object and obtain a maximum signal. This overcomes the problem that the the initial scan is often not on the source. I have effectively treated the first scan along each axis as a 'calibration' observation as a result.

In the initial stages of the reduction program development I only used Scan 4 since this should be on-source. I was flagging data manually at that stage and dealing with the additional scans would have been difficult. I subsequently increased the scope to include Scan 2,3 and 4 along each axis by imposing a set of criteria outlined in Section 5.7. This set of criteria simplified the automatic flagging procedure considerably and allowed me to fit elliptical Gaussians rather than quadratics (see Section 5.4).

### 5.3.4 Choice of observed object

Our typical observation requires that the object conform to the following specifications:

- Strong point source
- Close to Zenith during observation (to minimise gravitational and atmospheric effects on the pointing)

- Accurately known mean position

Fortunately the observation scheduler (which is outlined in Section 5.2) allows constraints to be placed on a list of suitable objects. These constraints included zenith distance which allowed a limit to be easily placed on the zenith distance. A suitable zenith distance of  $30^{\circ}$  was decided on. This prevented observation of any objects not within  $30^{\circ}$  of zenith. We removed any objects not within the range of  $55^{\circ} \rightarrow -5^{\circ}$  Declination as they would never pass within the required distance of the zenith (since HATRAO is at approximately  $25^{\circ}$  latitude). We only used bright sources with mean positions determined by VLBI measurements.

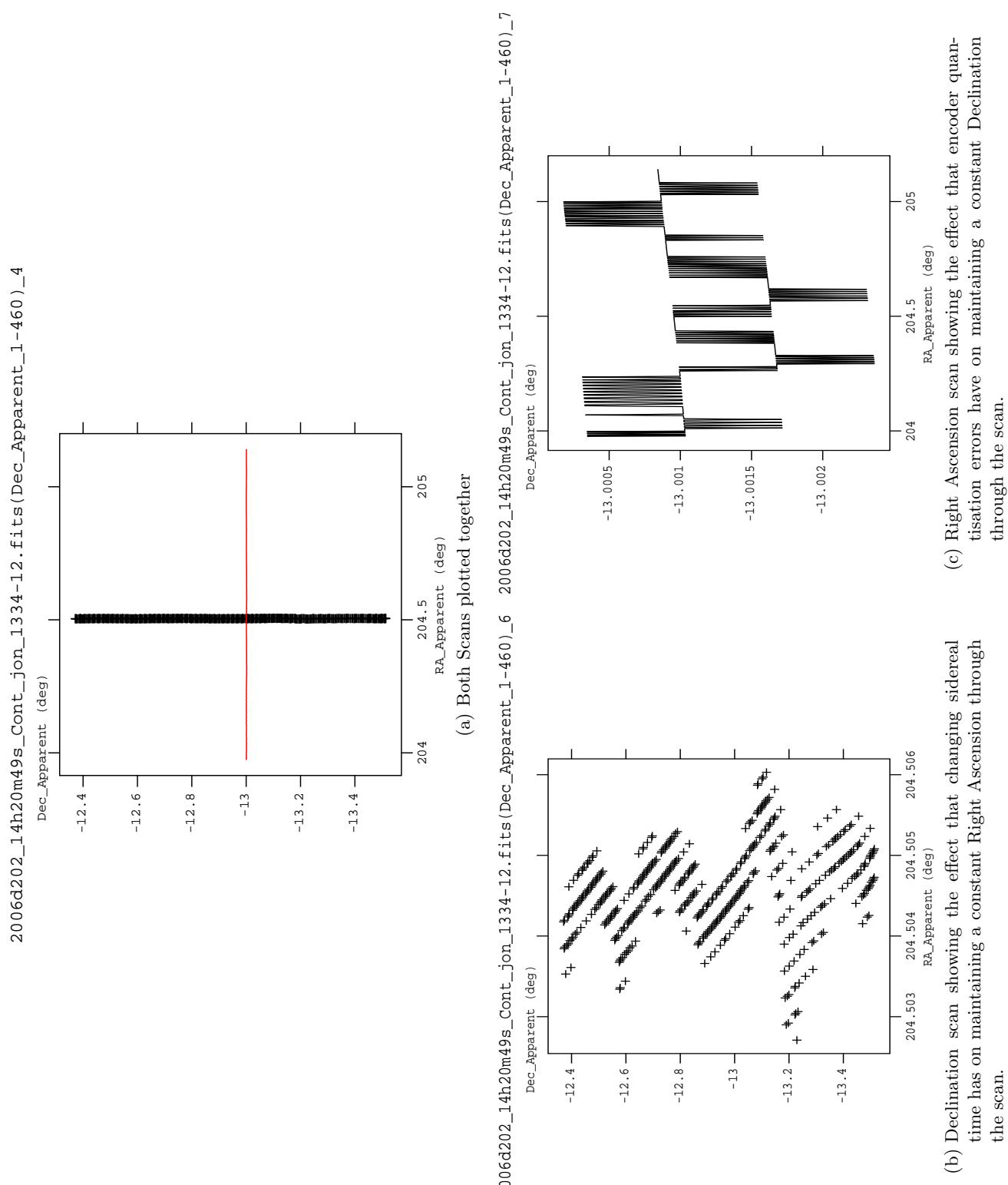


Figure 5.1: A typical cross scan trajectory of an object showing the Right Ascension and Declination movement of the telescope beam.

## 5.4 Measuring Pointing Offsets Using Point Sources

This section describes data obtained at HATRAO from point source observations, and introduces analytical functions that can be used to obtain the observed position from the observed signal distribution. Pointing offsets can be obtained by comparing the observed position of a point source against an accurately known VLBI catalogue position.

### 5.4.1 Typical Data from Point Source Observations

Data typical of a cross scan of a point source is included in Figure 5.2. Observations require baseline correction as illustrated by Figure 5.2. It is used to remove any slope or offset in the baseline due to atmospheric/electrical changes or variations of the response of the antenna during the scan. In these examples a quadratic baseline is fitted using the first and last 50 data points in the scan and is then subtracted from the data. The *baseline corrected* data is shown in Figure 5.3.

There is a noticeable difference between the two cross scans, with the Declination scans only showing one negative peak and the Right Ascension scans showing both a negative and positive peak. The negative peak occurs due to the beam differencing operation of the dual-beam dicke switched radiometer. The positive peak in the Right Ascension cross scan is due to the Hour Angle movement of the dual-beam *across* the point source. Declination cross scans do not result in the both beams moving across the point source and only show the negative peak.

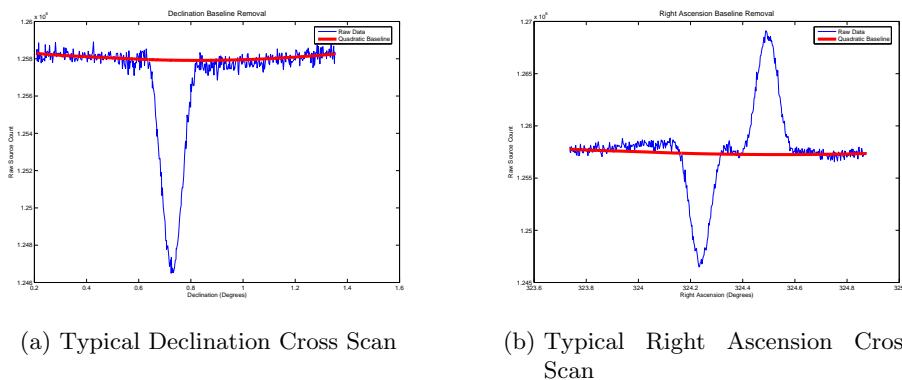


Figure 5.2: Typical Raw Data from a Set of Cross Scans

### 5.4.2 Elliptical Gaussians

The main beam lobe of a radio telescope can be approximated by an elliptical Gaussian distribution given by Equation 5.1

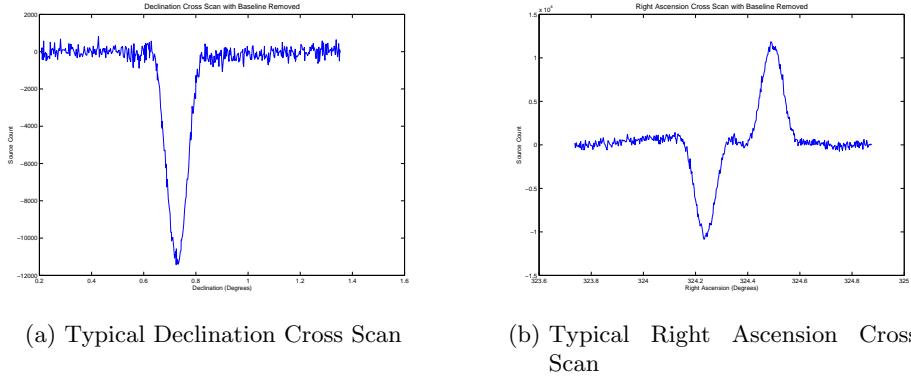


Figure 5.3: Typical Data from a Set of Cross Scans with Baselines Removed

$$T_{source}(x, y) = A_1 \exp \left[ -\frac{(x - x_0)^2}{2\sigma_x^2} - \frac{\beta(x - x_0)(y - y_0)}{\sigma_x \sigma_y} - \frac{(y - y_0)^2}{2\sigma_y^2} \right] \quad (5.1)$$

An observation of a point source will result in a signal distribution given by Equation 5.1. Combining the cross scans and fitting an elliptical Gaussian to the signal distribution, as shown in Figure 5.4 and Figure 5.5, allows the position of the point source to be obtained. If the object's celestial coordinates are accurately known from VLBI measurements, these observed positions can be used to obtain pointing offsets.

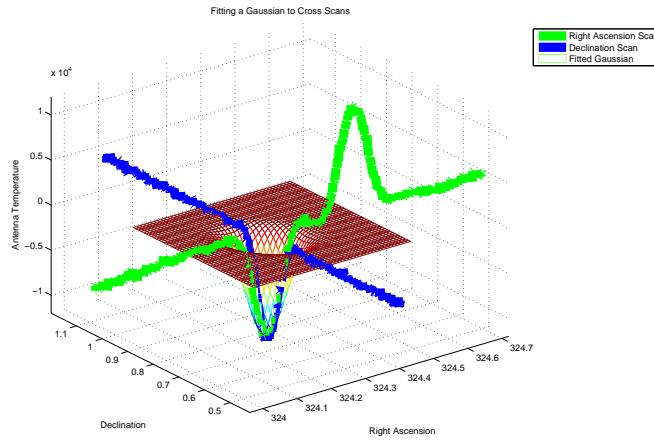


Figure 5.4: An elliptical Gaussian fitted to the combined Right Ascension and Declination data

In the final data reduction I combined the Declination and Right Ascension cross scans introduced in Subsection 5.4.1 and fitted an elliptical Gaussian to this combined data. The second Declination scan was combined with the second Right Ascension scan and a Gaussian was fitted to this combined data. I continued by using the third Declination and Right Ascension scans to fit a second Gaussian and did the same with the fourth Declination and

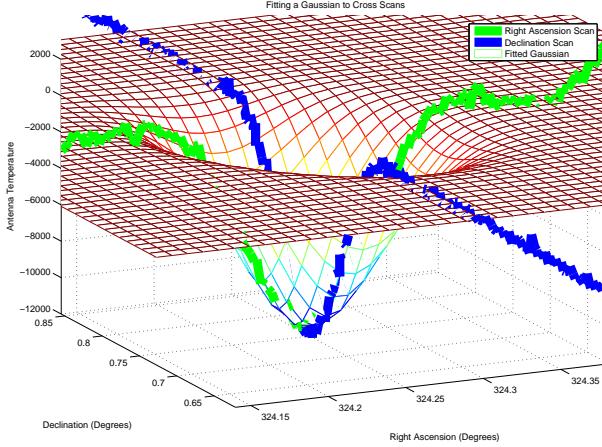


Figure 5.5: A closer view of the elliptical Gaussian shown in Figure 5.4

Right Ascension scans to produce a third Gaussian fit. I did not use the first scan for reasons outlined in Subsection 5.3.3.

### Elliptical Gaussian Uncertainties

Estimating uncertainties for the observation parameters is not trivial. Condon (1997) explains a general method for estimation used when fitting an elliptical Gaussian to images and estimating uncertainties in terms of pixels. In this case I used the following common astronomical definitions

$$\Theta_M = \sqrt{8\ln 2}\sigma_x \quad (5.2a)$$

$$\Theta_m = \sqrt{8\ln 2}\sigma_y \quad (5.2b)$$

$$\rho^2 = \frac{\pi}{8\ln 2} \frac{\Theta_M \Theta_m A^2}{h^2 \mu^2} \quad (5.2c)$$

The error estimates of these values is given by

$$\frac{2}{\rho^2} \approx \frac{\mu^2(A)}{A^2} \quad (5.3a)$$

$$= 8 \ln 2 \frac{\mu^2(x_0)}{\Theta_M^2} \quad (5.3b)$$

$$= 8 \ln 2 \frac{\mu^2(y_0)}{\Theta_m^2} \quad (5.3c)$$

$$= \frac{\mu^2(\Theta_M)}{\Theta_M^2} \quad (5.3d)$$

$$= \frac{\mu^2(\Theta_m)}{\Theta_m^2} \quad (5.3e)$$

A note here is that the  $h^2$  represents the area of each 'pixel' and the  $\mu$  represents the RMS of the noise in the scan. To use this method, I estimated the RMS as the RMS of the first 50 data points in both the Declination and Right Ascension scans. I estimated the 'area' of each pixel (in square radians) by multiplying the Declination step size by the Right Ascension step size. An explanation of the derivation of the equations given in Equations 5.3 is included in Appendix A. I found the estimated uncertainties in position estimated from the Gaussian fitting to be negligible ( $\ll 1\text{mdeg}$ ) compared to the desired pointing accuracy of approximately 4 mdeg, and so I haven't focused much attention on them in this study. Typical values are included in Appendix A.2, Figure A.1a and Figure A.1b to substantiate this decision.

### 5.4.3 Other Functions

Simpler functions such as 2-D Gaussians or quadratics are often used instead of the elliptical Gaussian to obtain the observed position. This is because it can prove difficult to automate the fitting process of the elliptical Gaussian due to the complexity and non-linearity of the function.

#### 2-D Circular Gaussian

It is often computationally easier to fit a 2-D Gaussian to a signal, and to use this to obtain positions of objects from pointing scans along one axis. A 2-D Gaussian can be expressed as

$$T_{source}(x) = P_i \exp\left(-\frac{(x - X_i)^2}{2\sigma_i^2}\right) \quad (5.4)$$

A Gaussian of this form is characterised by 3 parameters. These are the central abscissa  $X_i$ , the central ordinate  $P_i$  and the dispersion  $\sigma_i$  (Kaper et al., 1966). Uncertainties can be estimated (Kaper et al., 1966; Gaylard, 2005b) such that (where  $a$  = Number of Data points across FWHM)

$$\sigma_{P_i} = \left[ \frac{\text{RMS}_{noise}}{\sqrt{a}} \right] \quad (5.5)$$

$$\sigma_{width} = \text{Width} \times \frac{\sigma_{P_i}}{P_i} \quad (5.6)$$

$$\sigma_{X_i} = \frac{\sigma_{width}}{2} \quad (5.7)$$

#### Quadratics

A quadratic is even less computationally difficult to fit than a 2-D Gaussian. This is often used as a first approximation to the lower quarter of a Gaussian signal, and positions can be estimated from this. For the initial purposes of my investigation, a Gaussian fitting was not required since I was only interested in the position of the peak and I began this study

by using a quadratic approximation. This has the advantage of being considerably simpler to implement and more robust than Gaussian fitting since we can work with a linear function.

## 5.5 A Priori Pointing Corrections

There are three possible corrections that may have been applied to the observed coordinates before they were written to the FITS file. These may need to be removed from the data before data analysis can be done. The pointing corrections are:

1. Refraction
2. VLBI Pointing Model Corrections
3. Dr. Nicolson's original thermal pointing correction model

The first and second of these models was applied to all of the data that I collected, while the third model was only applied to data collected before January 2007, after which a keyword was added to the scheduler file parser (see Listing 5.2) to allow the application of this pointing correction to be selected. The two sets of observations were stored separately and it was easy to note whether the thermal correction model had been applied or not. It is necessary to remove the thermal pointing correction model from the observations before beginning any analysis. Depending on the type of analysis, it may be necessary to remove the VLBI Pointing Model corrections.

## 5.6 Data Reduction

The data reduction is done in two steps. The first involves extracting the data, pre-processing it and returning initial values of the pointing offset and the second analysing this data and fitting various models to it.

### 5.6.1 Step One- Extraction and Reduction of Data from Observation FITS Files

I used Fortran77 code primarily because of the availability of the established FITSIO library which allowed easy manipulation and reading of FITS file data.

Code was written to read in data from each FITS file sequentially. The observations contained in the FITS files are detailed in Subsection 5.3.3 and flow charts of the basic process are shown in Figure 5.6. The program first subtracts a baseline from each of the scans (as shown in Figure 5.2) and then proceeds to obtain initial estimates of the Gaussian parameters using a quadratic fit across the Gaussian peak. These estimates are then used to provide reasonable starting values for the non-linear fit of the elliptical Gaussian.

Thereafter the scans are grouped into sets of two comprising a Declination scan and a Right Ascension scan. The data sets are combined and a elliptical Gaussian is fitted to the combined data with the peak position recorded. The program records whether the thermal correction model has been applied to the data as well as the heuristic thermal correction model value for the time of the scan. It removes the VLBI pointing model from each of the scans and writes the Declination, Right Ascension, Hour Angle and signal counts to file. Uncertainties are estimated for each of fit parameters and then saved to file. The data with the VLBI model removed is used later in refitting the VLBI model (results shown in Figure 5.8), to ensure that there is minimal contamination from other causes of pointing offset besides the thermal effects. The file names containing the data are summarised in Table 5.1.

Table 5.1: Summary of the contents of the *analysis.txt* files output in the first stage of the data analysis.

File	Description
<i>analysis.txt</i>	Primary Data from fitting to the data before removing the VLBI pointing model. Includes time, the catalogue position of the object and whether Dr. Nicolsons original temperature correction had been applied together with the correction that needed to be removed.
<i>analysis1.txt</i>	Further information on the object such as ambient temperature and the epoch used for catalogue positions
<i>analysis2.txt</i>	The Left Circularly Polarised Scans from each FITS file. These are the scans with the VLBI pointing model removed from the coordinates.
<i>analysis3.txt</i>	The Right Circularly Polarised Scans from each FITS file with the VLBI pointing model removed from the coordinates
<i>analysis4.txt</i>	Uncertainty estimates for the Left Circularly Polarised Data fits
<i>analysis5.txt</i>	Uncertainty estimates for the Right Circularly Polarised Data fits

A note should be added to the above. The current system derives all coordinates from apparent Hour Angle and Declination. This is the Hour Angle and Declination once corrections for refraction and diurnal aberration are made. The conversion from Hour Angle to Right Ascension requires the local apparent sidereal time which can be calculated from UT1. UT1 is not the same as UTC and differs from the latter by less than a second at all times (this is maintained by adding and subtracting leap seconds from UTC). The conversion from Hour Angle to Right Ascension at HARTRAO ignored this small difference between UT1 and UTC and assumed UT1=UTC in the calculation of the sidereal time. I have compensated for this by retroactively applying a UT1-UTC correction and recalculating the Right Ascension values listed in the FITS files. This change has subsequently been incorporated into the HARTRAO control system.

### 5.6.2 Step Two- Examining Different Pointing Model Effectiveness

The rest of the data analysis is performed in MATLAB. This is done to take advantage of the greater versatility provided by this high level programming environment. I initially wrote a number of scripts to perform the analysis, however it soon became difficult to keep track of what had been done and I found myself redoing many time consuming calculations far too many times for my sanity. I decided to create a GUI (screenshot shown in Figure 5.7) which allowed previous results to be saved to file and allowed me to keep a better track of what I was doing.

The GUI allows me to proceed easily through the following options:

- Removing thermal pointing corrections if necessary.
- Recalculating VLBI Pointing Model coefficients in an effort to remove any remaining systematic pointing errors .
- Correlating temperature readings with scans. Both instantaneous thermal conditions and an average of temperature readings over a previous period were calculated since temperature effects do not have an instantaneous effect on the support structures and might occur as a result of earlier temperature conditions.
- Binning data according to thermal properties if necessary.
- Attempting to model the pointing offsets using the models outlined in Chapter 4.

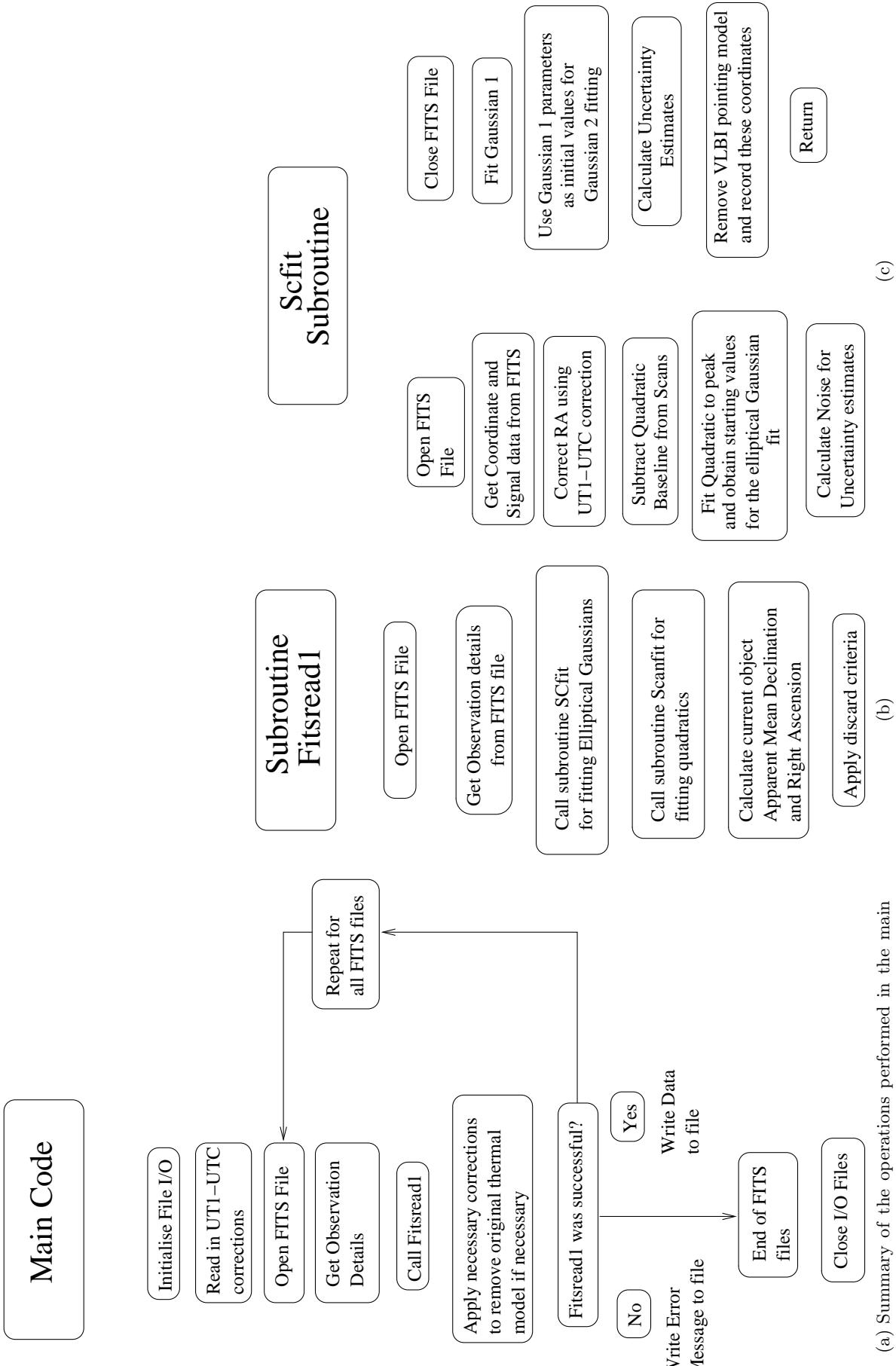


Figure 5.6: Flow diagram of the Fits2point Fortran code.

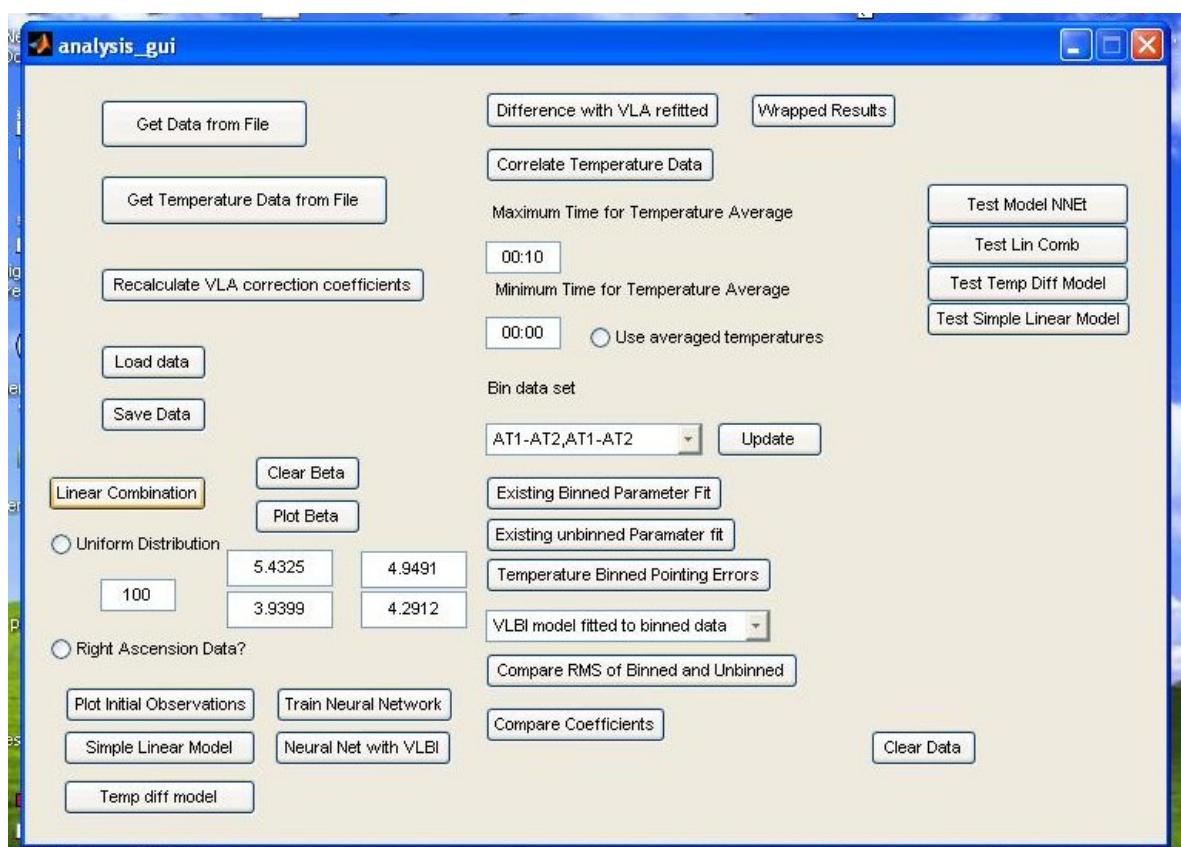


Figure 5.7: Screenshot of the MATLAB GUI interface used for the data analysis

## 5.7 Ensuring Data Quality

After beginning data collection, it became evident that a relatively large portion of the observations (approximately 10%) were entirely unsuitable for use as pointing offset measurements. Many had low signal to noise ratios and certain scans appeared to have missed the source completely.

In an effort to minimise manual requirements I imposed a set of criteria on the observations which needed to be met before the observation was used in the data reduction. The criteria was similar to that used in Ukita (1999) on the Nobeyama Telescope (see Section 3.4). The selection criteria are given below:

- Signal to noise ratio greater than 10
- Pointing offset of less than 40 mdeg for both axes (estimated from quadratic fitting to the source) to discard obviously faulty scans. This assumed that thermally induced pointing errors were less than 40 mdeg which I believe to be a reasonable assumption since this is the beamwidth size. Any automatically calculated errors larger than this must be as a result of faulty scan data since the object would probably not have been within the beam at all.
- Of the 3 scans used along each axis, at least 2 should give measurements agreeing within 5 mdeg. Any not within this band were discarded. This allowed for one of the scans to be useless without discarding the other two.
- Quadratic fitting should be 'successful' on at least 2 of the scans. This means that the quadratic fitting should provide initial Gaussian parameter estimates for 2 corresponding Declination and Right Ascension scans which are used to fit the Gaussian. The process is explained briefly in Subsection 5.4.2.
- At least 2 out of the possible 3 Gaussian fittings should be successful.

This selection criteria is applied during Stage 1, and works well. Without the criteria a significant number of outlying data points are allowed into the data which require manual removal. The second criteria has obvious limitations if fitting to questionable scans still produces pointing errors within 40 mdeg, however I felt that the first and third criteria would take care of these scans.

## 5.8 Initial Observations and Refitting the VLBI Pointing Model

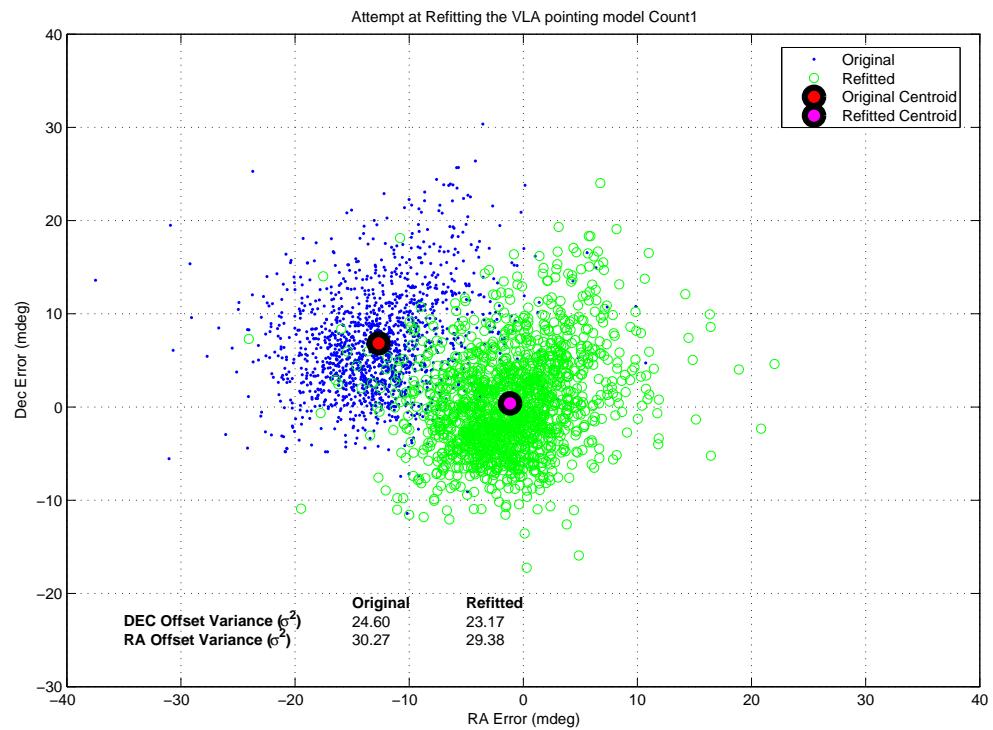
Data was collected from July 2006 to July 2007 and the pointing offset observations are summarised in Figure 5.8. We would expect that the pointing offsets should centre around (RA,DEC)=(0,0) with scatter due to pointing offset effects not considered in the VLBI pointing model.

As more data was obtained it became increasingly obvious that the pointing offset had an inherent non-zero offset as shown in Figure 5.8. This implies that other sources of pointing offset modelled in the VLBI pointing model may be influencing the pointing offset. Before trying to model the thermal effect on the pointing offsets, these other sources of contamination need to be removed as best as possible. To achieve this I removed the existing VLBI pointing model from observations and then refitted the parameters. The effectiveness of this refitting was potentially compromised by the 30° zenith angle limit (since the pointing model should be fitted with observations from as many different directions as possible), however it would at least allow the pointing offsets to be centred about zero for our purposes. The remaining offset would hopefully be due to effects not included in the model such as thermal effects.

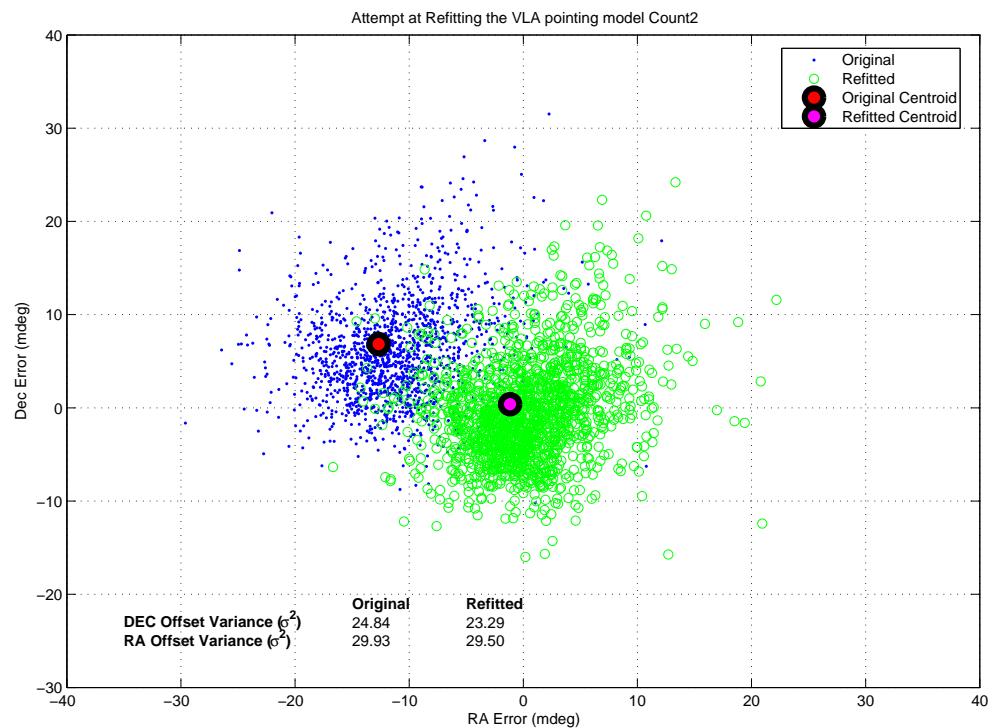
The results of this process are promising, as shown in Figure 5.8. The centroid of the original pointing offsets was at approximately (RA,DEC)=(-15,5) mdeg in the original pointing offset observations. This reflects poorly on the current parameters of the VLBI pointing model. The new coefficients result in reducing the systematic offset and moving the centroid to approximately (RA,DEC)=(0,0) mdeg. The variance of the pointing offsets is only marginally improved. The old and new parameters are shown in Table 5.2

Table 5.2: VLBI Pointing Model Parameters and the results of refitting the parameters using these observations.

Description	Original	Refitted	% Change
$P_1$	-0.0074075498	-0.01049454780367	29.4153
$P_2$	-0.0063556978	-0.00686492941186	7.4179
$P_3$	-0.0047638724	-0.00517999928007	8.0333
$P_4$	-0.0024362272	-0.00279710700103	12.9019
$P_5$	0.0544945337	0.05452390127463	0.0539
$P_6$	-0.0207822602	-0.02080699817014	0.1189
$P_7$	-0.0595969707	-0.05996367310289	0.6115
$P_8$	-0.0234767608	-0.02450930739531	4.2129
$P_9$	0	-0.00110319602355	100.0000
$P_{10}$	0	-0.00023846410765	100.0000
$P_{11}$	0.0173795968	0.01756346900581	1.0469
$P_{12}$	0	0.04962180835857	100.0000
$P_{13}$	0	0.00339941490416	100.0000
$P_{14}$	-0.0355144888	-0.10342461981940	65.6615
$P_{15}$	0	-0.00085846903721	100.0000
$P_{16}$	0	0.00893371292985	100.0000



(a) Left Circularly Polarised Data



(b) Right Circularly Polarised Data

Figure 5.8: The effect of recalculating the parameters of the VLBI pointing model on the centroid of the observed pointing offset.

# Chapter 6

## Results

This chapter deals with the results of using the previously described modelling techniques to predict and thus remove pointing offsets. I begin by demonstrating that the pointing offsets obtained during this study are similar to those obtained by Dr. George Nicolson during a previous investigation. I then show conclusively that there is a relationship between the support temperatures and the pointing offset before describing the results of the different modelling techniques.

### 6.1 Comparison to the Existing Heuristic Model

As mentioned in Section 2.3, a previous study of the effect of temperatures on pointing offsets was carried out by Dr. George Nicolson. The results of a comparison between model and data are summarised in Figure 6.1a. Section 2.3 mentioned that substantial alterations were performed to the dish subsequent to Dr. Nicolson's study and that this may have effected to thermal pointing offsets. If the pointing offset is caused by changes in the support structure, it is likely that there will be a good agreement between the existing model and the data since the supports have not been altered. The original pointing correction model is shown in Figure 6.1a and the Declination and Right Ascension pointing offsets obtained during this study are shown in Figure 6.1b and Figure 6.1c respectively. Since Figure 6.1a shows the correction model, we would expect the observed pointing offsets to be a reflection of this correction model about the time axis. In Figure 6.1b and Figure 6.1c we see the Declination and Right Ascension pointing correction models mirrored by both the observed Declination pointing offset and the Right Ascension pointing offset. It appears that the two studies agree reasonably well.

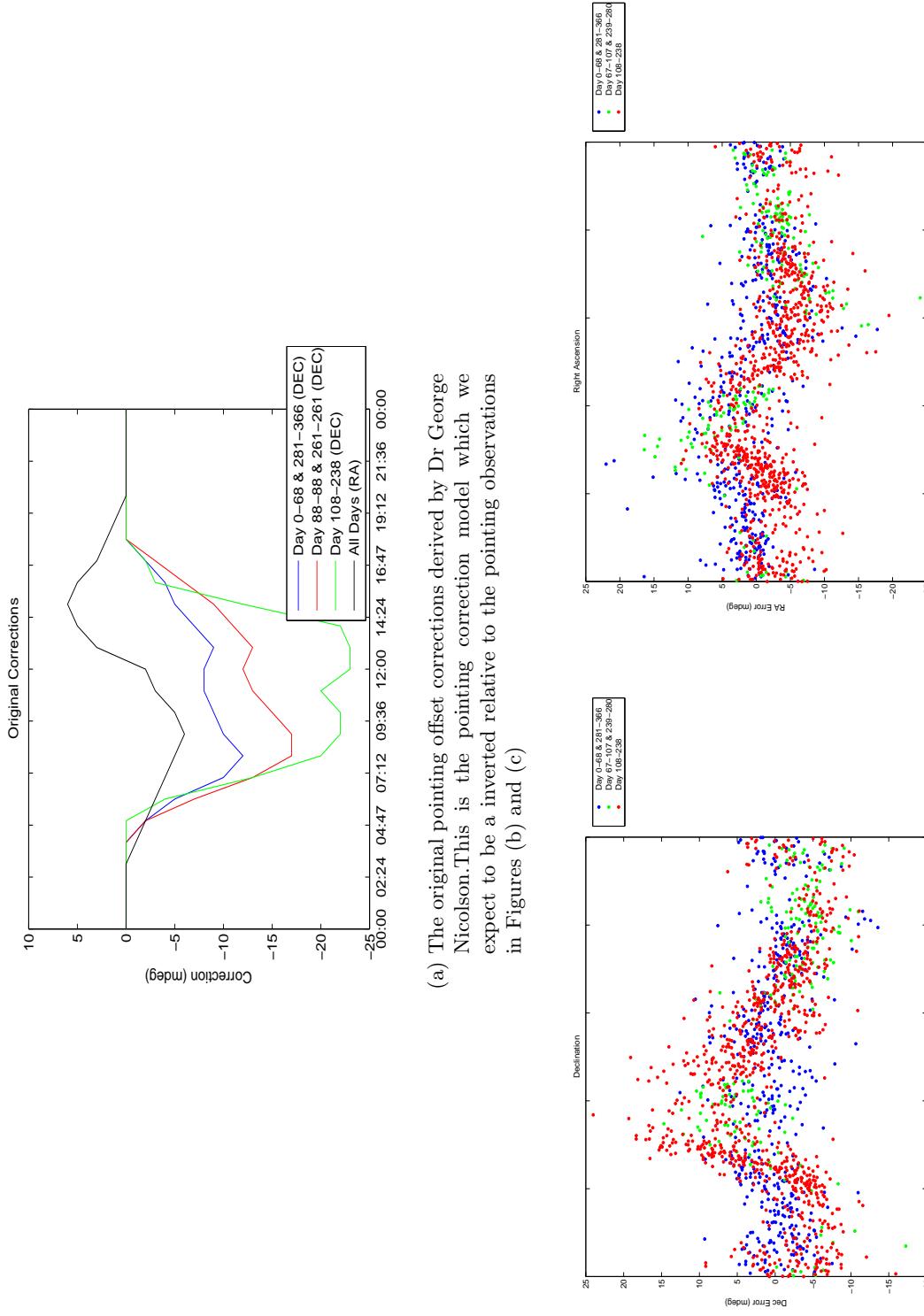


Figure 6.1: A comparison of the original thermal pointing model and the pointing offsets obtained during these observations. Since we expect (a) to be the inverse of (b) and (c), we note that there is a general agreement between the previous study by Dr. Nicolson and the observations obtained during this investigation.

## 6.2 The Relationship Between Thermal Conditions and Pointing Offset

Two different techniques were used to demonstrate that there is a relationship between the support thermal conditions and the pointing offset.

The first was the use of correlation matrices. Correlation matrices are used as a means of seeing linear relationships between the different variables as described in Section 4.4. They were used to examine the relationship between various thermal properties of the supports and the Declination and Right Ascension pointing offsets.

The second method was to bin the pointing offsets according to different thermal properties and to see if the centroid of the binned pointing offsets moved as the thermal properties changed.

Both methods show that there is a relationship between the temperature difference across the Declination supports and the Declination pointing offset, however the Right Ascension pointing offset does not have a clear relationship to the recorded thermal conditions. This is demonstrated graphically in Subsection 6.2.3 when the Declination and Right Ascension pointing offsets are plotted against the temperature difference between Declination and Hour Angle supports respectively.

### 6.2.1 Correlation Matrix

Two correlation matrices were formed. The first correlation matrix used the antenna support temperatures, the ambient air temperature and the pointing offsets of the two axes. The second used the temperature differential between opposing supports (i.e AT1-AT2, AT3-AT4 ... AT9-AT10) and the pointing offsets of the two axes. The correlation matrices are included in Table 6.1 and Table 6.2. Table 6.2 shows that there is a correlation between the temperature differential between Declination supports and the Declination pointing offset.

The highest correlation coefficients ( $R_{ij} > 0.99$ ) highlighted in Table 6.1 show that there is a strong correlation between the North facing Declination supports (AT1, AT3, AT5 and AT7), and a strong correlation between the South facing Declination supports (AT2, AT4, AT6 and AT8). This is to be expected since there should be a strong relationship between thermistors with the same North-South orientation relative to the Sun.

The correlation coefficients between the Hour Angle support thermistor measurements are also high. This correlation is because the Hour Angle support thermistor temperatures shown in Figure 2.3 are not as strongly influenced by solar illumination since they are orientated East-West. A significant temperature differential between the Hour Angle supports does not develop. This is in contrast to the lower correlation coefficients between the North-South

oriented Declination thermistor temperatures (the temperatures are shown in Figure 2.2) which show a greater temperature differential developing due to the solar illumination effect.

It is interesting to note the high correlation coefficients between the South East Declination support thermistor readings (AT6 and AT8) and the Hour Angle support thermistor readings (AT9 and AT10). The correlation coefficients between the South West Declination supports thermistor readings are high ( $R_{ij} > 0.98$ ), however this relationship is not as strong as the previously mentioned one. I have not been able to explain these correlations satisfactorily.

The correlation matrix shows a high degree of redundancy of the data. Later in this chapter Principle Component Analysis is used to reduce the dimension of the data, and examine the effectiveness of the new data as a possible input.

The second correlation matrix (Table 6.2) shows strong correlation between the Declination support temperature differentials and the Declination pointing offset, as expected from the simple linear model outlined in Section 4.3 and the investigation in Section 6.2.2. The strongest relationship is between AT5-AT6 and the Declination pointing offset, with a correlation coefficient of  $R = 0.77350$ . This shows that the temperature differential between the East facing Declination supports (AT5 is the top NE thermistor and AT6 is the top SE thermistor) is the most strongly correlated to the Declination pointing offset. The Declination angle encoder is mounted on the East side of the Declination axis, which may play a role in causing this observation. There is not a strong relationship between the Hour Angle support temperature differentials and the Right Ascension pointing offsets.

Table 6.1: Correlation Matrix ( $R$ ) of the support temperatures, the ambient temperature and the pointing offsets. High correlations ( $R_{ij} > 0.99$ ) are highlighted in red and the diagonal is highlighted in blue

	AT1	AT2	AT3	AT4	AT5	AT6	AT7	AT8	AT9	AT10	Tambient	RA Offset	DEC Offset
AT1	1.00000	0.97634	0.99890	0.97029	0.99197	0.98417	0.99169	0.98133	0.98081	0.98674	0.95697	0.00427	0.14489
AT2	0.97634	1.00000	0.97429	0.99847	0.95889	0.99271	0.95691	0.99417	0.98814	0.98468	0.88770	-0.07281	-0.01128
AT3	0.99890	0.97429	1.00000	0.96841	0.99446	0.98449	0.99426	0.98146	0.98019	0.98632	0.96013	0.01582	0.15460
AT4	0.97029	0.99847	0.96841	1.00000	0.95260	0.99194	0.95057	0.99416	0.98882	0.98347	0.87863	-0.07854	-0.03424
AT5	0.99197	0.95889	0.99446	0.95260	1.00000	0.97823	0.99946	0.97372	0.97096	0.98018	0.96787	0.06757	0.19581
AT6	0.98417	0.99271	0.98449	0.99194	0.97823	1.00000	0.97630	0.99337	0.99128	0.99194	0.91575	-0.00382	0.03467
AT7	0.99169	0.95691	0.99426	0.95057	0.99946	0.97630	1.00000	0.97180	0.96989	0.97897	0.96952	0.06743	0.20174
AT8	0.98133	0.99417	0.98146	0.99416	0.97372	0.99337	0.97180	1.00000	0.99104	0.99069	0.90838	-0.01385	0.01888
AT9	0.98081	0.98814	0.98019	0.98882	0.97996	0.99128	0.96989	0.99104	1.00000	0.99379	0.91996	-0.02935	0.04011
AT10	0.98674	0.98468	0.98632	0.98347	0.98018	0.99194	0.97897	0.99069	0.99379	1.00000	0.93626	0.01334	0.06968
Tambient	0.95697	0.88770	0.96013	0.87863	0.96787	0.91575	0.96952	0.90838	0.91996	0.93626	1.00000	0.12997	0.30499
RA Offset	0.00427	-0.07281	0.01582	-0.07854	0.06757	-0.00382	0.06743	-0.01385	-0.02935	0.01334	0.12997	1.00000	0.25268
DEC Offset	0.14489	-0.01128	0.15460	-0.03424	0.19581	0.03467	0.20174	0.01888	0.04011	0.06968	0.30499	0.25268	1.00000

Table 6.2: Correlation Matrix ( $R$ ) of the temperature differential between axis supports and the pointing offsets. The highlights draw attention to the correlation between the Declination pointing offsets and the Declination support temperature differentials ( $AT_i - AT_j$ )

	AT1-AT2	AT3-AT4	AT5-AT6	AT7-AT8	AT9-AT10	RA Offset	DEC Offset
AT1-AT2	1.00000	0.95309	0.91820	0.92094	-0.38728	0.35479	0.71697
AT3-AT4	0.95309	1.00000	0.93573	0.94380	-0.40159	0.37480	0.72730
AT5-AT6	0.91820	0.93573	1.00000	0.98433	-0.36820	0.34244	0.77350
AT7-AT8	0.92094	0.94380	0.98433	1.00000	-0.35470	0.34125	0.76615
AT9-AT10	-0.38728	-0.40159	-0.36820	-0.35470	1.00000	-0.38297	-0.26539
RA Offset	0.35479	0.37480	0.34244	0.34125	-0.38297	1.00000	0.25268
DEC Offset	0.71697	0.72730	0.77350	0.76615	-0.26539	0.25268	1.00000

### 6.2.2 Binning of the Pointing Offsets

A binning procedure was used to confirm the dependence of pointing offset on temperature. The observations were grouped according to various classes of temperature (i.e ambient air temperature and both Declination and Hour Angle support temperature differentials), and the corresponding pointing offsets were plotted. The results are shown in Figure 6.2, 6.3 and 6.4. These show that the pointing offset does vary with the temperature difference between Declination supports (Figure 6.2) in agreement with Table 6.2, but does not vary significantly with the temperature differential between the Hour Angle supports (Figure 6.3) or the ambient air temperature (Figure 6.4).

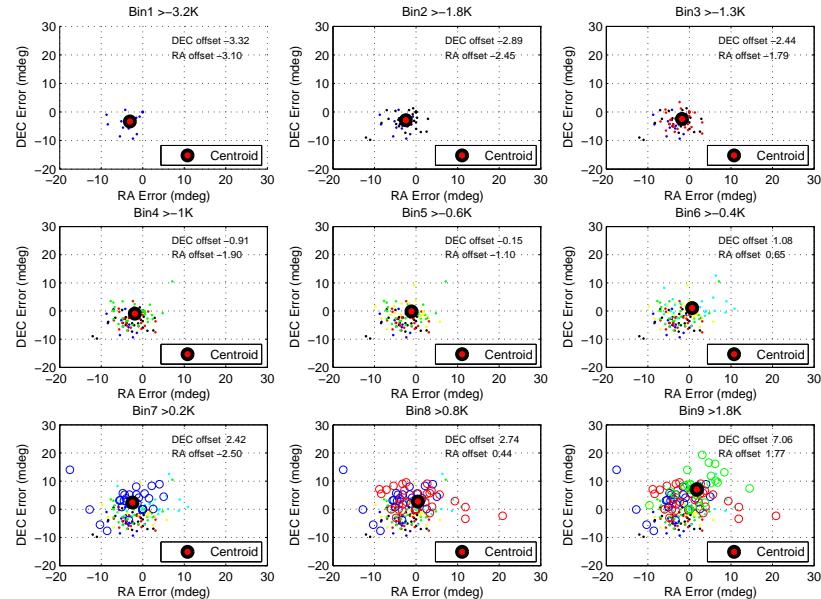


Figure 6.2: The pointing offsets are binned according to the temperature differential between AT5 and AT6 (see Table 2.1 for the locations of the thermistors). This is the temperature difference between Declination supports and can be written as AT5-AT6. We see the position of the pointing offset moving with bin number which implies a dependence of the pointing offset on AT5-AT6.

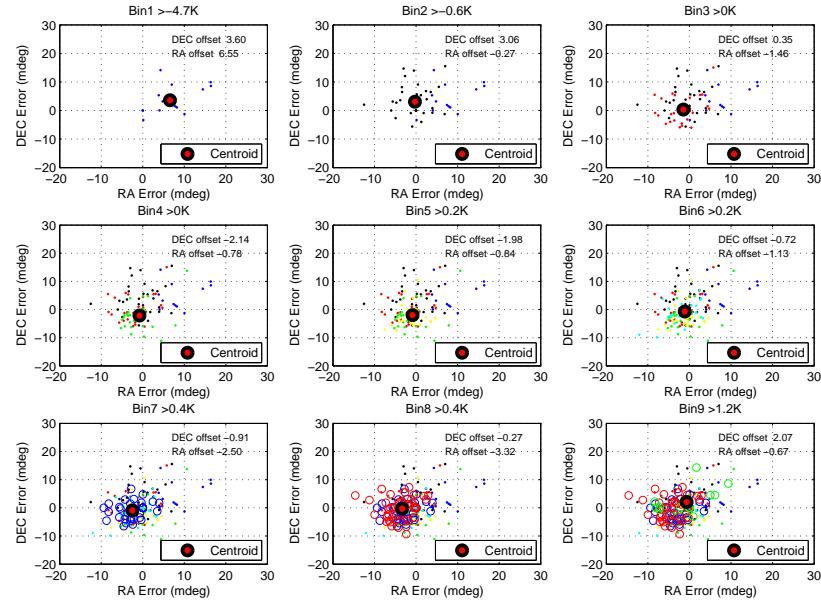


Figure 6.3: These graphs show the data binned using AT9-AT10 (see Table 2.1). This is the temperature difference between the Hour Angle supports. The graph does not show any marked relationship between the pointing offset and the temperature difference AT9-AT10. If there were we would see the pointing offset change with bin number.

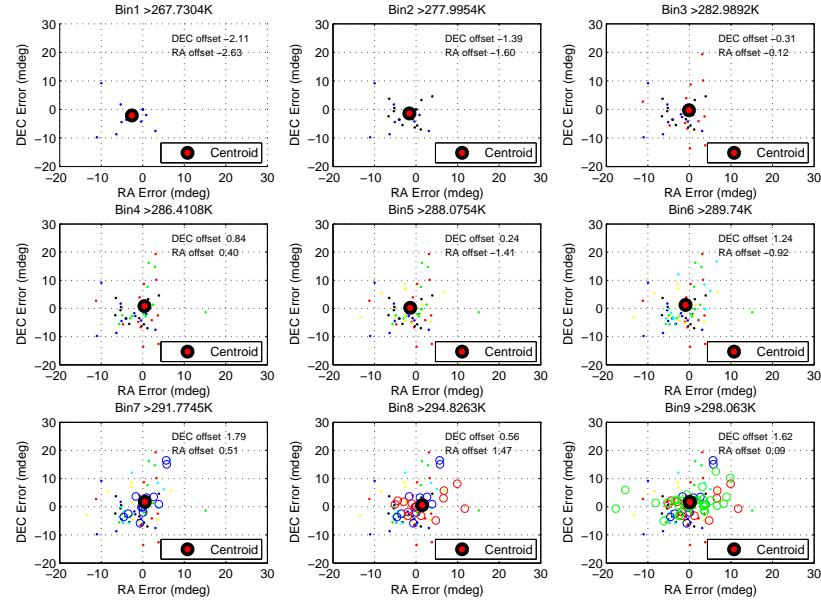


Figure 6.4: In this case the pointing offsets are binned according to the ambient temperature at the time of the observation. There is no clear dependence of the pointing offset on the ambient temperature. If this was the case we would be able to see the grouping of the pointing offset change with different bins.

### 6.2.3 Linear Fitting of the Pointing Offset to the Temperature Differential

The correlation matrix and the binning procedure outlined earlier both suggest a relationship between the Declination pointing offset and the temperature difference between the Declination supports. The data set that looked most promising was AT5-AT6 (see the correlation matrix in Table 6.2), and I pursued this as a starting point. The results are summarised in Figure 6.5a which show a strong relationship between the Declination support temperature differentials and the Declination pointing offsets.

This was not true for the Right Ascension pointing offsets which showed no correlation between pointing offset and Hour Angle (Right Ascension) support structure temperature differential (AT9 and AT10). These results are shown in Figure 6.5b.

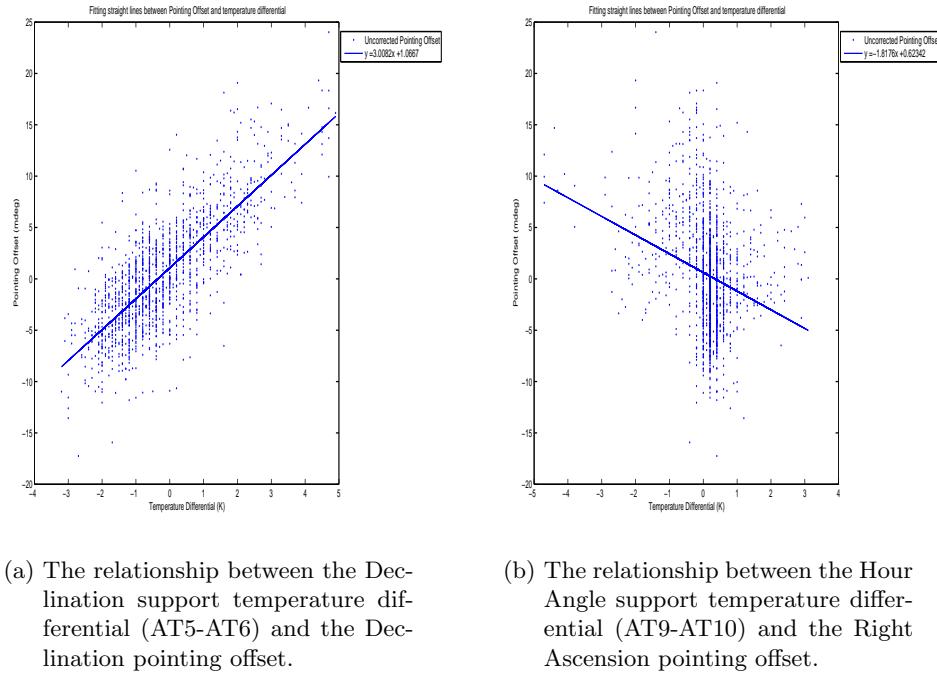


Figure 6.5: The relationships between the temperature differentials of various support struts and the pointing offset.

## 6.3 Simple Physical Model of the Declination Supports

A simple model of the Declination pointing offsets is described in Section 4.3. It describes a simple physical model of the Declination supports, and assumes that Declination pointing offsets are caused entirely by deformations of these Declination supports. While this is unlikely to provide a comprehensive result, it is interesting to see how successful the model is. To investigate this I plotted the actual Declination offsets against those predicted by the model with the results shown in Figure 6.6. The results were promising and gave credibility

to the theory that the Declination pointing offsets were caused by thermal expansion and deformation of the Declination supports. The data used has the VLBI pointing model parameters refitted as described in Section 5.8.

It would be possible to further refine this model and possibly obtain better results, however the model is limited by the rough approximations made in developing it. Thus the results are left at this stage as a further convincing piece of evidence that the Declination support temperatures play a part in causing the Declination pointing offset, and that this is a reasonable foundation from which to begin investigating.

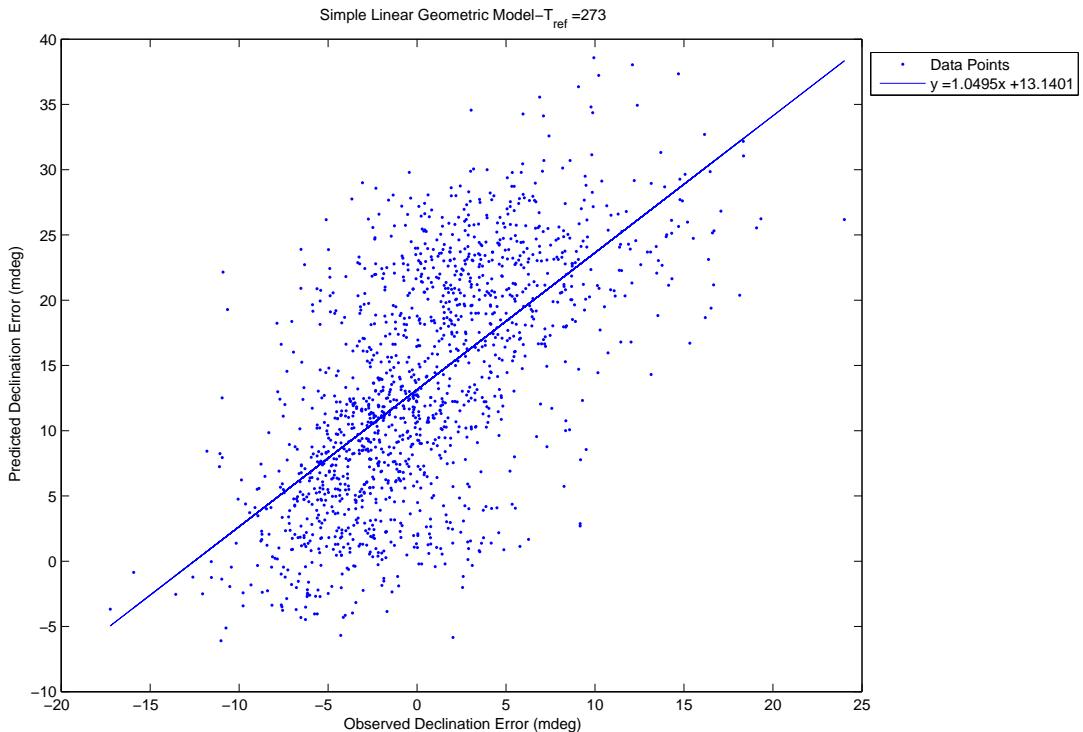


Figure 6.6: The correlation between the Declination pointing offset predicted by a simple geometric model (described in Section 4.3) of the telescope and the actual observed offsets.

## 6.4 Principle Component Analysis

As explained earlier Principle Component Analysis (PCA) seeks to redefine a data set in terms of a new basis, so as to maximise the Signal-to-Noise ratio and to minimise data redundancy. The effect is that the vectors formed from the new basis are both orthonormal to each other and ordered in terms of decreasing variance so that the first Principle Component has a greater variance than the second Principle Component and so forth.

Since each of the Principle Components is orthonormal and linearly independent (as they

form a basis of the vector space), it is interesting to see if there is any correlation between the Principle Components and the pointing offsets. Correlation matrices are included in Table 6.3 and Table 6.4, and the first three Principle Components are shown in Figure 6.7 and Figure 6.8. The Principle Components used in Table 6.3 and Figure 6.7 are derived from all the thermistor readings while the Principle Components used in Table 6.4 and Figure 6.8 use only the Hour Angle thermistors.

Table 6.3 shows that there is a good correlation between the second Principle Component and the Declination offset ( $R=0.7657$ ), but that there is no such correlation between any of the Principle Components and the Hour Angle offset. I have included Table 6.4 to demonstrate that the Hour Angle thermistor readings are definitely not linearly related to either of the pointing offsets. Table 6.4 does not show any correlation between Principle Components of the Hour Angle thermistors and the Hour Angle pointing offset.

In terms of the temperature data, the first Principle Component (i.e the vector with the greatest variance) will represent the large scale, diurnal temperature variation experienced by the thermistors, which we expect to have the largest variation. The second Principle Component will represent the next smallest variation of the data, which is likely to be the difference between the individual temperatures and the large scale diurnal temperature variations. If this is true then the second Principle Component is likely to be related to temperature difference between supports. This relationship is shown in Figure 6.9, which shows the clear linear relationship between the Declination support temperature differential, AT5-AT6, and the second Principle Component. Figure 6.10a shows that there is a strong linear relationship between the second Principle Component and the Declination offset as is expected from the large correlation between these two indicated by Table 6.3

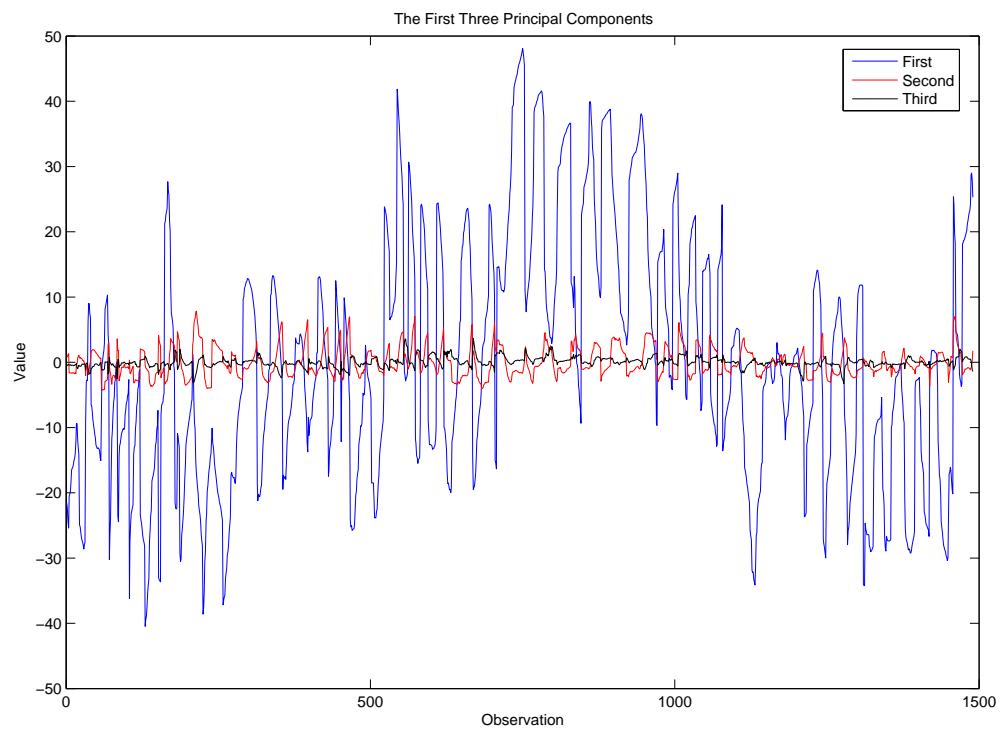


Figure 6.7: The Principle Components of the Declination Support temperatures

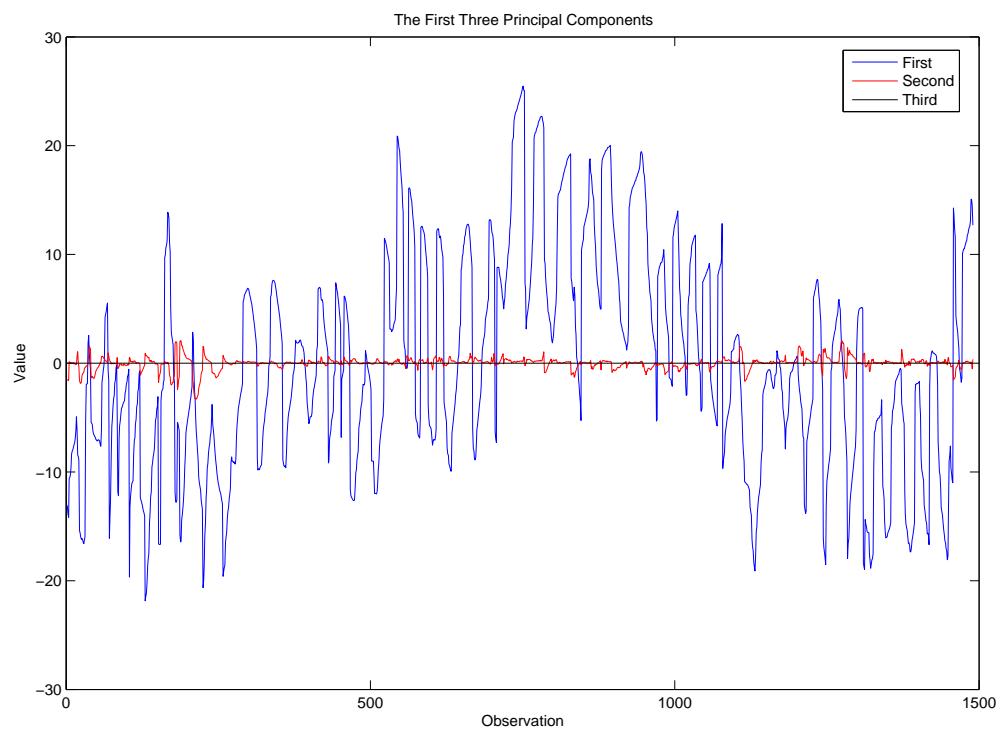


Figure 6.8: The Principle Components of the Hour Angle Support temperatures

Table 6.3: Correlation matrix between the Principle Components of all thermistor readings and the pointing offsets.

PC	1	2	3	4	5	6	7	8	9	10	RA Offset	DEC Offset
1	1.00000	0.00000	0.00000	0.00000	-0.00000	0.00000	-0.00000	0.00000	-0.00000	0.00000	0.01148	-0.07010
2	0.00000	1.00000	0.00000	-0.00000	-0.00000	-0.00000	0.00000	-0.00000	0.00000	0.00000	0.40332	0.76757
3	0.00000	0.00000	1.00000	0.00000	-0.00000	0.00000	-0.00000	0.00000	0.00000	-0.00000	-0.19638	-0.03062
4	0.00000	-0.00000	0.00000	1.00000	0.00000	-0.00000	0.00000	0.00000	-0.00000	-0.00000	0.41210	-0.06287
5	-0.00000	-0.00000	-0.00000	0.00000	1.00000	0.00000	-0.00000	0.00000	-0.00000	-0.00000	0.17813	-0.07955
6	0.00000	-0.00000	0.00000	-0.00000	0.00000	1.00000	0.00000	-0.00000	-0.00000	-0.00000	-0.09622	0.09810
7	-0.00000	-0.00000	-0.00000	0.00000	0.00000	-0.00000	1.00000	0.00000	0.00000	0.00000	-0.04416	-0.00890
8	-0.00000	0.00000	-0.00000	0.00000	-0.00000	-0.00000	0.00000	1.00000	-0.00000	-0.00000	-0.03963	-0.06084
9	0.00000	-0.00000	0.00000	-0.00000	-0.00000	-0.00000	-0.00000	0.00000	1.00000	0.00000	-0.04563	-0.01736
10	-0.00000	0.00000	-0.00000	-0.00000	-0.00000	-0.00000	-0.00000	-0.00000	0.00000	1.00000	0.02047	0.05204
RA Offset	0.01148	0.40332	-0.19638	0.41210	0.17813	-0.09622	-0.04416	-0.03963	-0.04563	0.02047	1.00000	0.25268
DEC Offset	-0.07010	0.76757	-0.03062	-0.06287	-0.07955	0.09810	-0.00890	-0.06084	-0.01736	0.05204	0.25268	1.00000

Table 6.4: Correlation matrix between the Principle Components of only the Hour Angle thermistor readings and the pointing offsets.

PC	1	2	RA Offset	DEC Offset
1	1.00000	-0.00000	0.01577	-0.04441
2	-0.00000	1.00000	-0.38411	0.27573
RA Offset	0.01577	-0.38411	1.00000	0.25268
DEC Offset	-0.04441	-0.27573	0.25268	1.00000

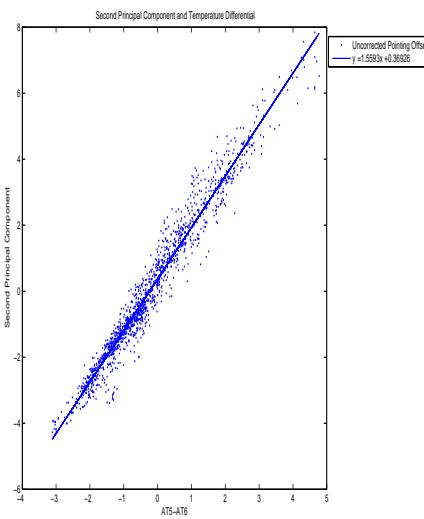
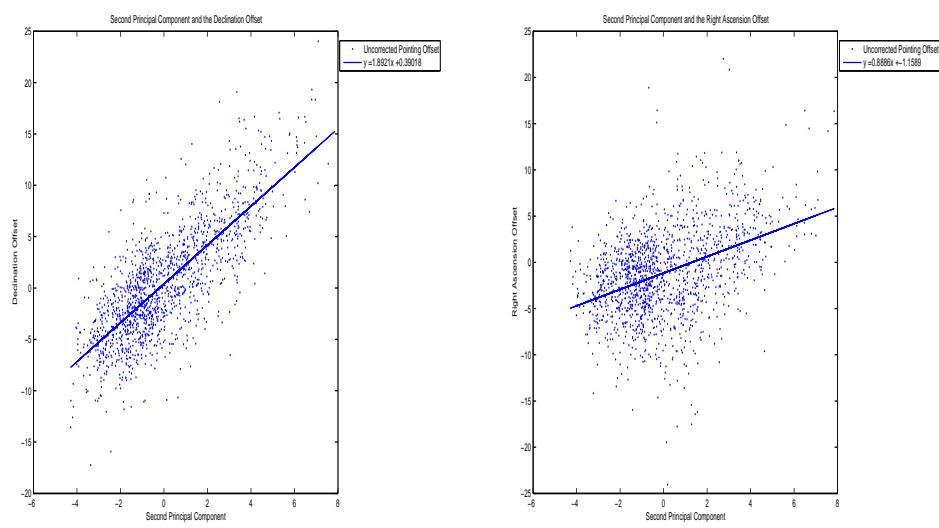


Figure 6.9: The second Principle Component plotted against the Declination temperature differential, AT5-AT6



- (a) The relationship between the second Principle Component and the Declination pointing offset
- (b) The relationship between the second Principle Component and the Right Ascension pointing offset

Figure 6.10: Pointing offsets and the second Principle Component

## 6.5 Linear Fitting of a Weighted Combination

The technique of using weighted inputs was introduced in Equation 4.11, Section 4.11. Equation 4.11 is repeated here for clarity

$$\text{Model Pointing Offset} = \sum_{i=1}^{i=n} a_i AT_i \quad (6.1)$$

The technique involves weighting the  $n$  inputs,  $AT_i$ , with the coefficients  $a_i$ , and varying the weights until the difference between the model pointing offset and the observed pointing offset is minimised. Larger weighting coefficients correspond to a better linear relationship between the input and the pointing offset.

### 6.5.1 Weighted Temperatures

Based on the relative success of the simple linear model and the clear relationship between the temperature differential and the Declination pointing offset, it seemed sensible to try and fit a model using the individual temperature readings as inputs,  $AT_i$ , to the model in Equation 6.1. It is important to note that this model is potentially compromised by the large redundancy between the variables (shown in Table 6.1). The fitting procedure may not work properly due to this redundancy, however the results are still interesting.

Since this model requires fitting to data, it is important to separate the data into a training set and a testing set. I removed 100 data points from the fitting and used these as test points. The model was fitted on the remaining data points and the result of this model when applied to the test data points is shown in Figure 6.11.

One of the things that becomes apparent from Figure 6.11d, is the marked effect that AT5 and AT6 have on fitting to the Declination pointing offset. These are the thermistors located at the top of the North-East and South-East Declination supports. The angle encoders are located on the East of the Declination supports and this may be why thermal deformations of these supports appear to be significant in causing Declination pointing offsets. This agrees with Section 6.2.1, which finds that the greatest correlation is between the temperature differential between AT5 and AT6 and the Declination pointing offset ( $R=0.7735$ ).

This model shows a marked improvement for the Declination pointing offset (45%), however the Right Ascension pointing offset only improves by 27%. This might be improved by increasing the number of thermistors along the Hour Angle supports.

### 6.5.2 Weighted Principle Components

As noted earlier, a large fraction of the temperature data is redundant. Redundant data is often problematic when fitting, due to the presence of a large number and extent of local minimums. Since the Principle Components minimise the redundancy, it seems sensible to attempt the same procedure outlined in the subsection 6.5.1, but using the Principle Components as the inputs  $AT_i$ . We expect the largest weighting (and hence best linear correlation) for the Declination pointing model to correspond to the second Principle Component since the highest correlation coefficient between the Principle Components and the Declination pointing offset occurs with the second Principle Component (see Table 6.3). The same correlation matrix suggests that the Right Ascension model will perform poorly (since there are no large correlation coefficients between the Right Ascension pointing offsets and the Principle Components). The results are shown in Figure 6.12. Figure 6.12d shows that the largest weighting for the Declination pointing offset model is given to the second Principle Component as expected.

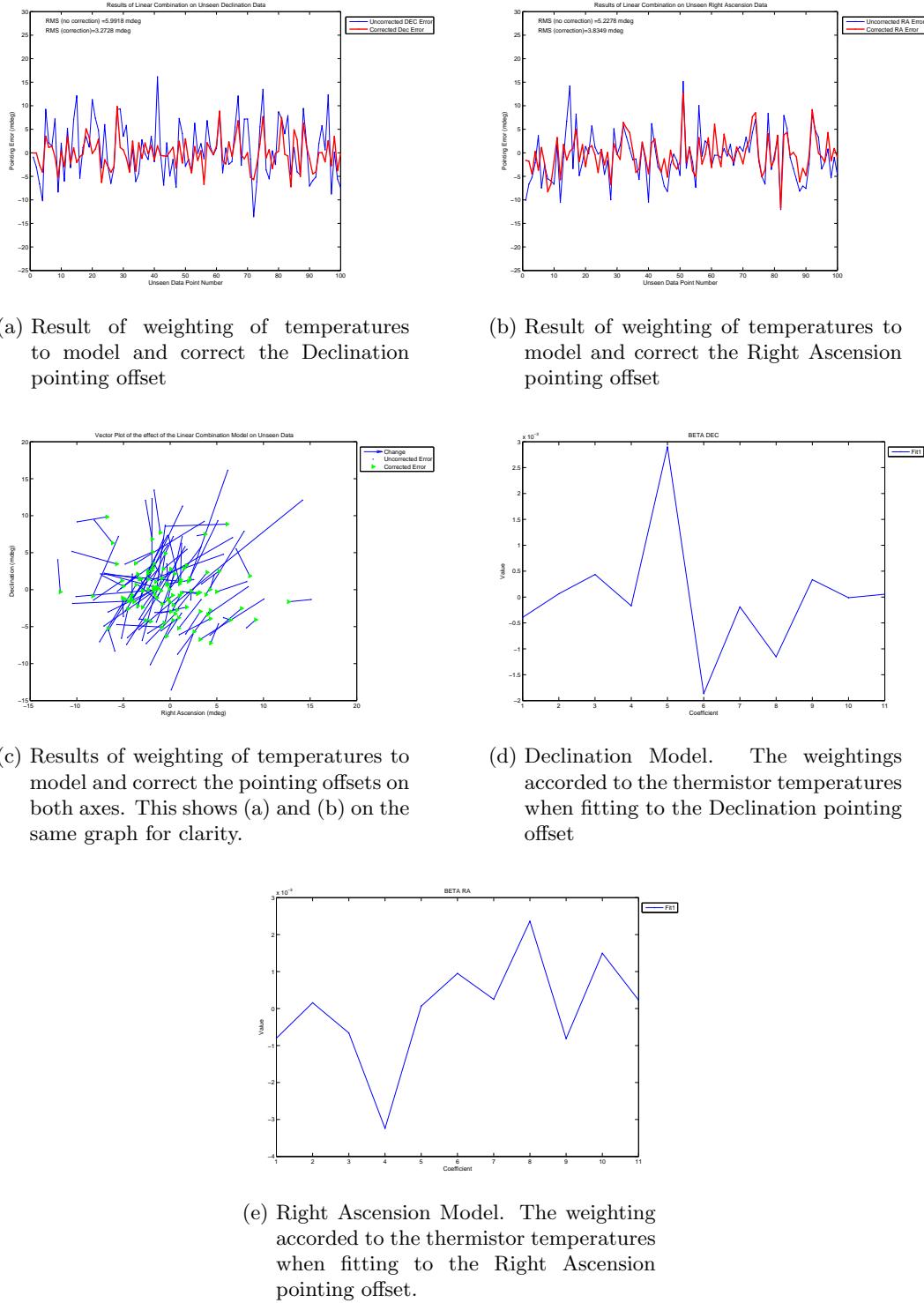


Figure 6.11: Weighted linear combination of the temperatures using AT1 → AT10 and  $T_{ambient}$  (represented by coefficient 11).

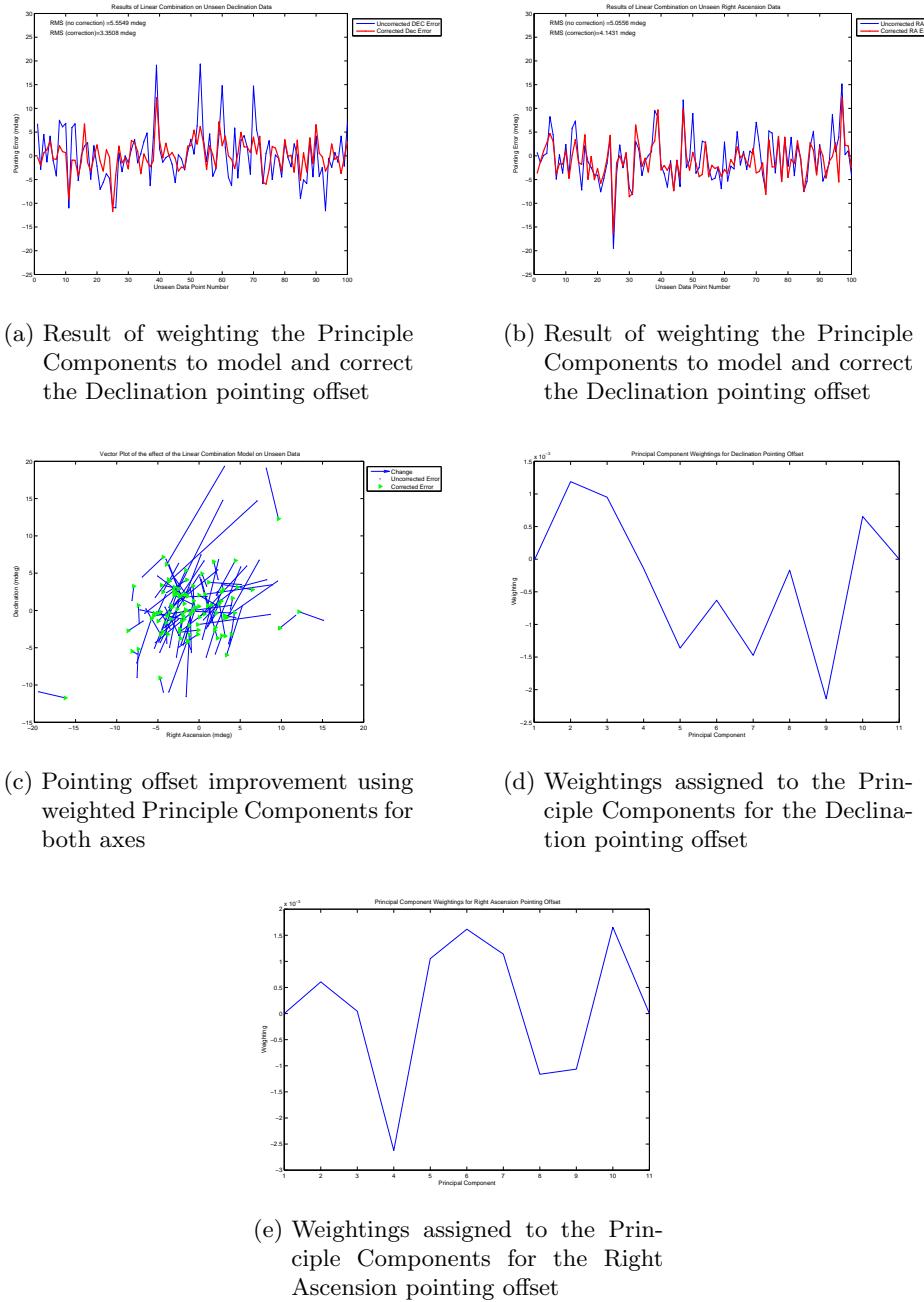


Figure 6.12: Weighting the Principle Components as a means of modelling the pointing offsets

## 6.6 Using the VLBI Pointing Model and Temperature Binning as a Solution

One way of trying to understand any temperature induced twisting effects would be to use the VLBI Pointing Model<sup>1</sup> as a physical model, and fitting this function to data binned according to a temperature criteria. It would be necessary to identify which of the parameters in the VLBI model might be susceptible to temperature variations, and to allow these to vary while keeping the others fixed. We would then be able to roughly determine how the fitted coefficients varied as a function of temperature.

This is potentially useful. Certain VLBI coefficients are related to twisting effects of the axes. This may occur due to temperature variations and, if this was the case, it would be revealed by any variations of the parameters with temperature. Descriptions of the coefficients which may be effected by temperature variations are shown in Table 6.5, with the full list included in Appendix B.

The effects on the coefficients once fitted to binned data are shown in Figure 6.13a. Little physical information is revealed by the process (indeed it seems to say that the Perpendicular Axis Skew and the “Tilt Over” effects mirror one another), however using the different coefficients as corrections for the various bins does appear to be effective when the corrected pointing offsets shown in Figure 6.14 are compared to the uncorrected pointing offsets shown in Figure 6.2. The pointing offsets remain roughly constant (compared to the variation seen in Figure 6.2) in each bin once the varying VLBI models are applied to each bin.

What is particularly interesting in 6.13a is that Coefficient 3 and Coefficient 6 mirror each others behaviour. It is difficult to gain any real physical information about what is happening from these graphs apart from the observation that the effect ( $P_5$ ) appears to vary in an almost sinusoidal manner as the Declination temperature differential increases.

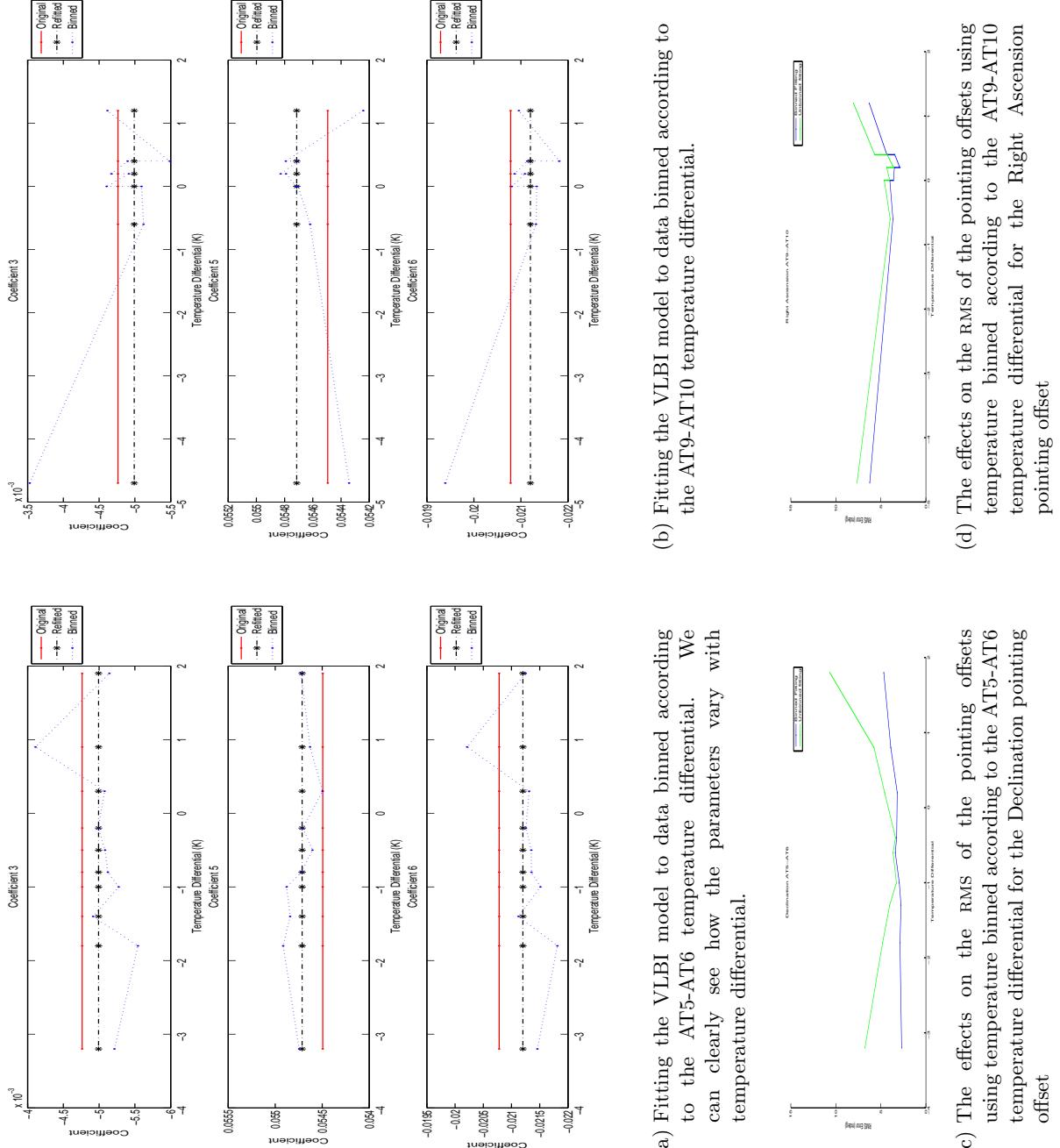
Table 6.5: Description of the fitted VLBI Pointing Model Coefficients (Himwich, 1993)

Coefficient	Description
$P_3$	Perpendicular Axis Skew
$P_5$	“Tilt Out”
$P_6$	“Tilt Over”

The results for the Declination pointing offsets is fairly good and can be seen in Figure 6.13c. These show the RMS of the pointing offset both before and after the correction has been applied. The Right Ascension pointing offset does not vary with the temperature differential between the Hour Angle supports suggesting that a different method will need to be employed to obtain a model for the Right Ascension pointing offsets.

---

<sup>1</sup>see Appendix B for further information on the VLBI pointing model and coefficients



(a) Fitting the VLBI model to data binned according to the AT5-AT6 temperature differential. We can clearly see how the parameters vary with temperature differential.  
 (b) Fitting the VLBI model to data binned according to the AT9-AT10 temperature differential.  
 (c) The effects on the RMS of the pointing offsets using temperature binned according to the AT5-AT6 temperature differential for the Declination pointing offset  
 (d) The effects on the RMS of the pointing offsets using temperature binned according to the AT9-AT10 temperature differential for the Right Ascension pointing offset

Figure 6.13: The effect of fitting the VLBI coefficients to data binned according to thermal properties.

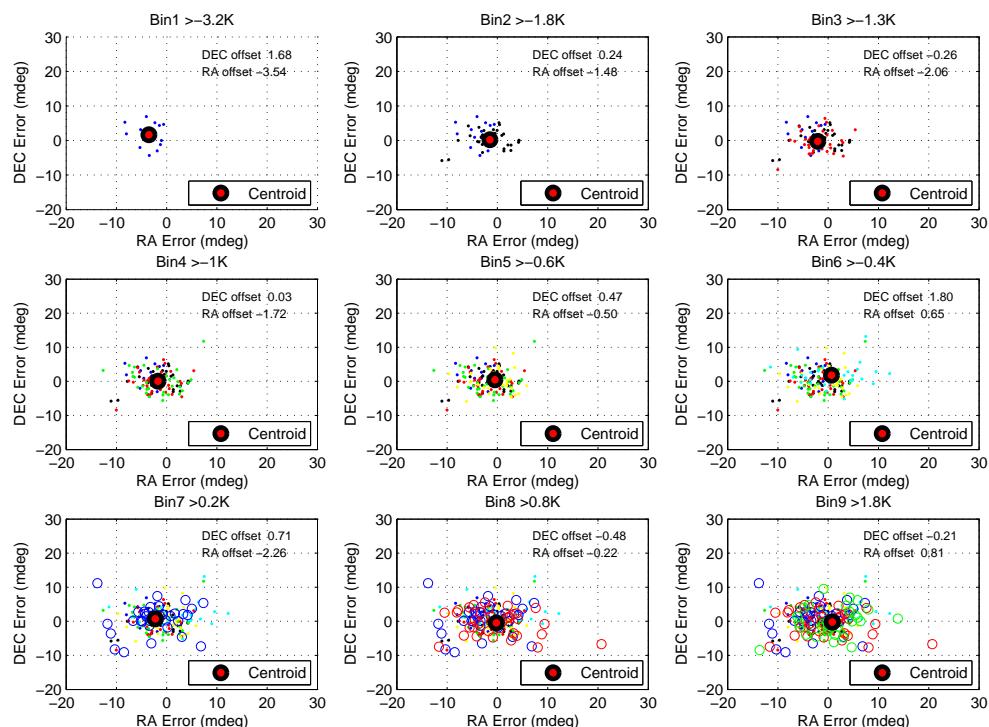


Figure 6.14: Graphs showing the corrected pointing offsets for each bin of the temperature differential AT5-AT6 as well as the mean Declination and Right Ascension pointing offset for each bin.

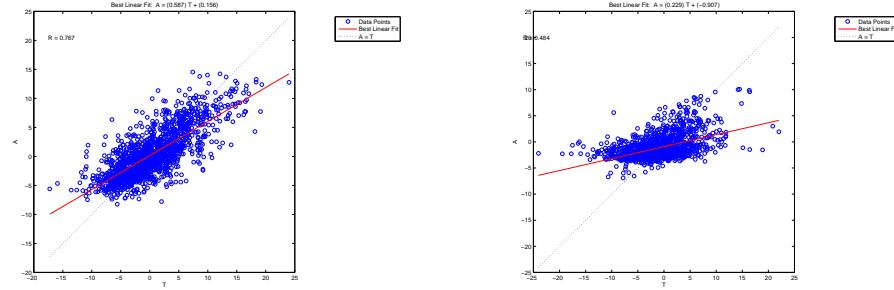
## 6.7 Neural Network

Neural networks present an interesting and promising solution to the problem. Neural networks are particularly useful in models where the underlying physical nature of the problem is not well understood. To demonstrate that a neural network is a feasible means of correcting the pointing offset, I set up a basic Neural Network and trained it using randomly selected data points.

### 6.7.1 Temperature Inputs

I chose to use the commonly utilised feed-forward back propagation network. I initially chose to have two layers of neurons. The first layer was comprised of 5 neurons and used a *tansig* transfer function (see Figure 4.2). The second layer was made of 2 neurons and used the *purelin* transfer function. The propagation algorithm used was the *Levenberg-Marquardt*. In order to avoid overfitting I used the *Early Stopping* technique and subdivided so that 25% was included in the validation data set, 25% was included in the test set, and the remaining 50% was used for training data.

I preprocessed the data by normalising the mean and standard deviation of the data using MATLAB's *prestd()* function. This is a standard step in training neural networks and allows for more efficient training. I used Principle Component analysis to discard any data that contributed less than 1% of the variance. This left me with only the first and second Principle Components, which contained over 99% of the data variation. The results of applying the model correction to unseen data are shown in Figure 6.15. The results were promising, particularly with regards to correcting the Declination pointing offset.



(a) A regression analysis of the predicted Declination offset against the observed offset.

(b) A regression analysis of the predicted Right Ascension offset against the observed offset.

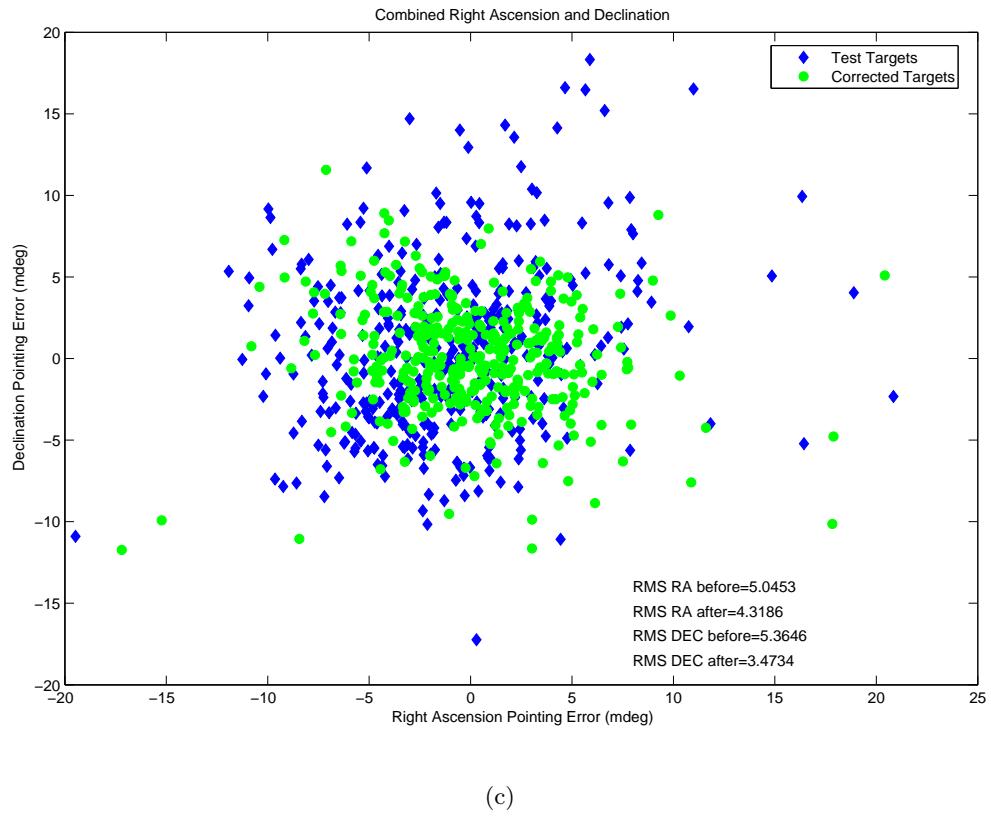
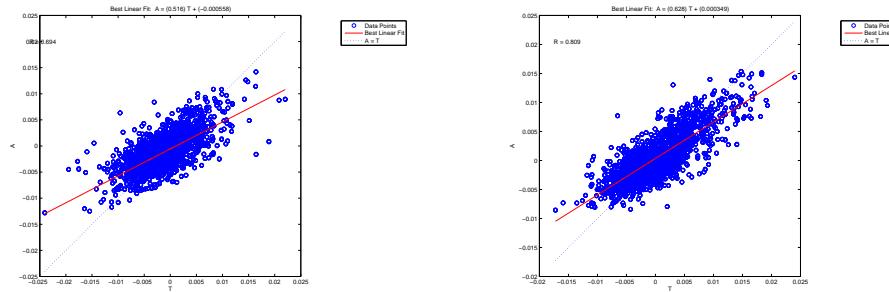


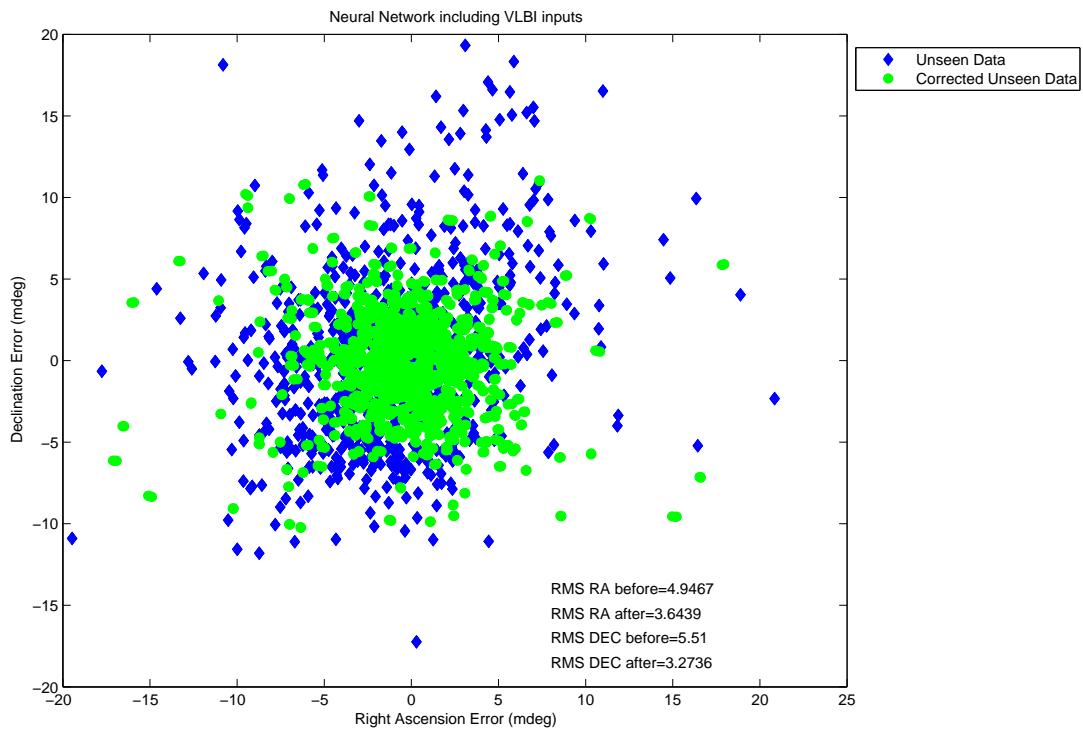
Figure 6.15: The results obtained using a neural network model trained on AT1→AT10 and  $T_{ambient}$ .

### 6.7.2 Neural Networks as a Substitute for the VLBI Pointing Model

Since the binning of the data and then refitting VLBI parameters had proved to be successful I decided to try to expand the Neural Network inputs to include the variables used in the VLBI fitting i.e Hour Angle and Declination as well as the support temperatures. I hoped that this would result in a trained neural network that could replace the VLBI pointing model and include thermal corrections as well. Typical results of this investigation are shown by Figure 6.16. I believe these to be a success. Figure 6.16 shows that a neural network could possibly be used as a pointing model to take all possible pointing offset contributions into account such as gravitational deformation, axis misalignments and more. The Right Ascension pointing offsets still appear to be problematic, however there is a clear improvement in general.



(a) Regression showing the predicted Right Ascension offset against the observed offset.  
(b) Regression showing the predicted Declination offset against the observed offset.



(c) The result of the neural network that incorporates VLBI inputs.

Figure 6.16: Neural network trained to replace the VLBI pointing model

# Chapter 7

## Conclusion

The results shown in the previous chapter are positive, particularly with respect to modelling the Declination pointing offsets. It appears that achieving an RMS pointing offset of less than 5 mdeg for both axes is realistically obtainable in the short term.

Further investigation should be carried out to determine the underlying physical cause of the Right Ascension pointing offsets, since none of the models implemented provides a completely satisfactory model for this axis. A suggestion from Prof. Heinz Ruther (U.C.T Geomatics Department) would be to try and obtain laser scan point clouds of the structure at different times during the day during the winter period. These scans could highlight directly where deformations are occurring. With this knowledge it might be possible to derive a simple model to accurately account for the Right Ascension pointing offsets. Alternatively a finite element model could be constructed from the laser scan dimensions and this could be used. Worth considering is that the backing structure of the telescope may be deforming as appears to happen at the Leighton Array at OVRO (Lamb and Woody, 2002) described in Section 3.3. Compensating for this would require installing a large number of temperature sensors across the backing structure and studying temperature differentials between these sensors.

Our understanding of the structural temperatures could be improved by using more temperature sensors across the structure. The results of this thesis indicate a relationship between pointing offset and Declination support temperatures. This suggests the need for more temperature sensors on the Declination supports for more comprehensive information. It would also be useful to place temperature sensors on any other Hour Angle supports to allow further insight to be gained on the possible causes of the Right Ascension pointing offset. Laser scans would be useful as a means of identifying critical points on the structure.

It would be useful to obtain more data, particularly over the 07h00→15h00 period when the pointing offset is most pronounced on the Declination axis. This would allow the model to be refined further and to produce better results in the future.

I think these are successful results in that they confirm that (at least for the Declination

pointing offset) simple and easy to implement models allow for the pointing offset to be brought to well within acceptable limits for the new receiver to operate. While the Right Ascension results are not yet completely satisfactory, I believe these findings lend weight to continuing with further investigations of this nature.

These findings give an idea of the type of thermal deformations that may effect *MeerKAT*. Since these dishes will have different backing and support structures a great deal of this investigation is not entirely applicable, however this does show that simple models might be suitable for use on *MeerKAT*.

## **Part II**

# **A Prototype Control System for the C-BASS Antenna**

# Chapter 8

## Introduction

### 8.1 The C-BASS Experiment

The Cosmic Microwave Background (CMB) captures a vast amount of information about the early universe and thus allows astronomers a glimpse at the Universe at the epoch where radiation and matter first decoupled (Keating et al., 1998). The CMB allows for a broad range of studies, among them determining the cosmological parameters. The CMB radiation is partially polarised because of rescattering of the primordial photons by the hot electron gas. Theory predicts that this polarisation is due to (1) variations in density in the early universe, and, (2) gravitational waves that originated shortly after the Big Bang. The state of polarisation can be completely described by the E mode and B mode components (Chiueh and Ma, 2002). Density variations in the early Universe would have generated variations in the E mode polarisation, while gravitational waves would have generated variations in the B mode polarisation (Chiueh and Ma, 2002; Zaldarriaga, 2004). While the polarised CMB temperature variations due to the effects of gravitational waves are predicted to be of the order of  $10^{-6} K$ , if they can be mapped they will provide information about the early history of the Universe, including information about the proposed Inflation phase of the Universe.

Many experiments are being designed and carried out to measure both the E and B mode fluctuations, however they all rely on being able to remove foreground contamination caused by polarised Galactic synchotron radiation to a similar degree of accuracy as that required by the experiment. Removing this foreground contamination is the primary scientific aim of the C-BASS experiment.

While a number of full sky surveys (King, 2006; Reich, 2006) of polarisation have been carried out at low frequencies ( $< 3GHz$ ), Faraday rotation, which is inversely proportional to the square of the frequency of radiation ( $F_{ROT} \propto \nu^{-2}$ ), contaminates the data at lower frequencies and does not allow for extrapolation into the CMB band at 60-150 GHz. At higher frequencies, the Galactic Emission becomes too faint to detect ( $T \propto \nu^{-2.7}$ ) and contamination

from the CMB itself causes problems as it increases in intensity relative to the Galactic foreground. 5 GHz is the optimum frequency to measure the intrinsic polarised Galactic Emission because both of these effects are negligible.

The C-BASS project has been designed to obtain precise measurements of the linear polarisation state of the Galactic synchotron radiation. To achieve this, a moderate resolution full sky survey at 5 GHz is proposed (King, 2006). Northern Hemisphere coverage will be provided by a 6 m dish located at the Owens Valley Radio Observatory, California (OVRO) and Southern Hemisphere coverage will be provided by a 7.6 m dish in the Karoo, South Africa. The project is a collaboration between University of Oxford, the University of Manchester, the California Institute of Technology and Rhodes University/HARTRAO.

The 7.6 m antenna (shown in Figure 8.1) to be used at the South African site was donated by Telkom, where it had been originally designed for Low-Earth Orbit (LEO) telecommunication applications. The telescope has been moved from the original Telkom site to HARTRAO, and will be relocated to the Karoo site after modification and testing. It requires a new control system to complete the change from telecommunications application to astronomy. Initial design and development of this control system provides the objective of this part of the thesis.



Figure 8.1: A photograph taken of the proposed C-BASS 7.6 m installed at HATRAO. Since this photo was taken an electronics shelter has been installed nearby.

## 8.2 Proposed C-BASS System

A functional diagram of the proposed C-BASS system is shown in Figure 8.2. The source signal will be received by the horn and sent to a cryogenic amplifier. The signal will then be passed through a digital back-end and then recorded by a observation PC. Low-level antenna control such as motor control and interlock monitoring, will be done by separate electronics (labelled *Antenna Control* in Figure 8.2), however the high level control (dominated by the angle transformations from celestial coordinates to horizontal coordinates) and the human interface, will be performed through the observation PC (labelled *Observing Computer* in Figure 8.2). Interfacing between the antenna control and the observation PC is through Ethernet. Time synchronisation between the digital back-end and the antenna control is achieved using a combination of a 1-PPS signal together with the SNTP Ethernet time synchronisation protocol. The SNTP protocol is detailed in Appendix E.3.

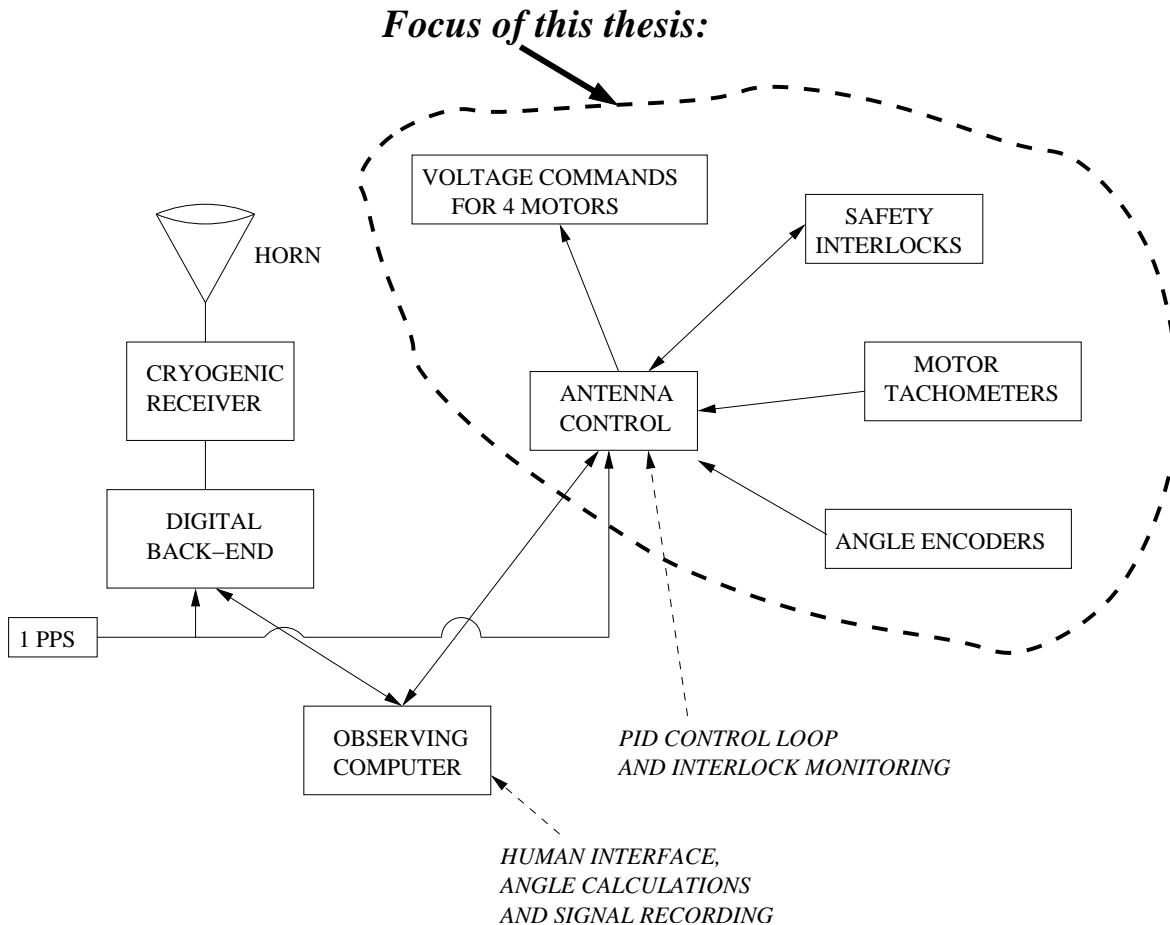


Figure 8.2: The layout of the proposed C-BASS system showing the receiver, controller and 1-PPS synchronisation signal

### 8.3 Existing NEC System

The antenna system acquired for the C-BASS project was originally designed by NEC for tracking LEO satellites. The original control system drove the antenna to an approximate satellite position and used a differential error beam to 'lock-on' to a monopulse signal originating from the satellite. Further details on the system (particularly low-level wiring) are included in Appendix F for use in ongoing development. This control system is not suitable for astronomical applications which rely on dead reckoning. However, if the control loop could be redesigned, almost all of the peripheral hardware (i.e drive control power amplifiers, angle encoders, limit switches etc.) could be successfully used for astronomical purposes.

#### 8.3.1 Existing Control System

A functional block diagram of the existing control system is detailed in Figure 8.3. It consists of two primary parts:

- Antenna Control Unit (ACU) This provides the high-level control of the original control

system using the differential error beam to calculate control signals. These signals are provided to the Drive Control Power Amplifier. The angle encoders interface to the ACU in the original control system.

- Drive Control Power Amplifier (DCPA) which consists of:

DCPA Control Board This provides the low-level control to control the DC motor current generated by the servopacks.

DCPA Relay Board This is used to control mechanical devices such as the motor clutches and brakes, and provides control of safety interlocks

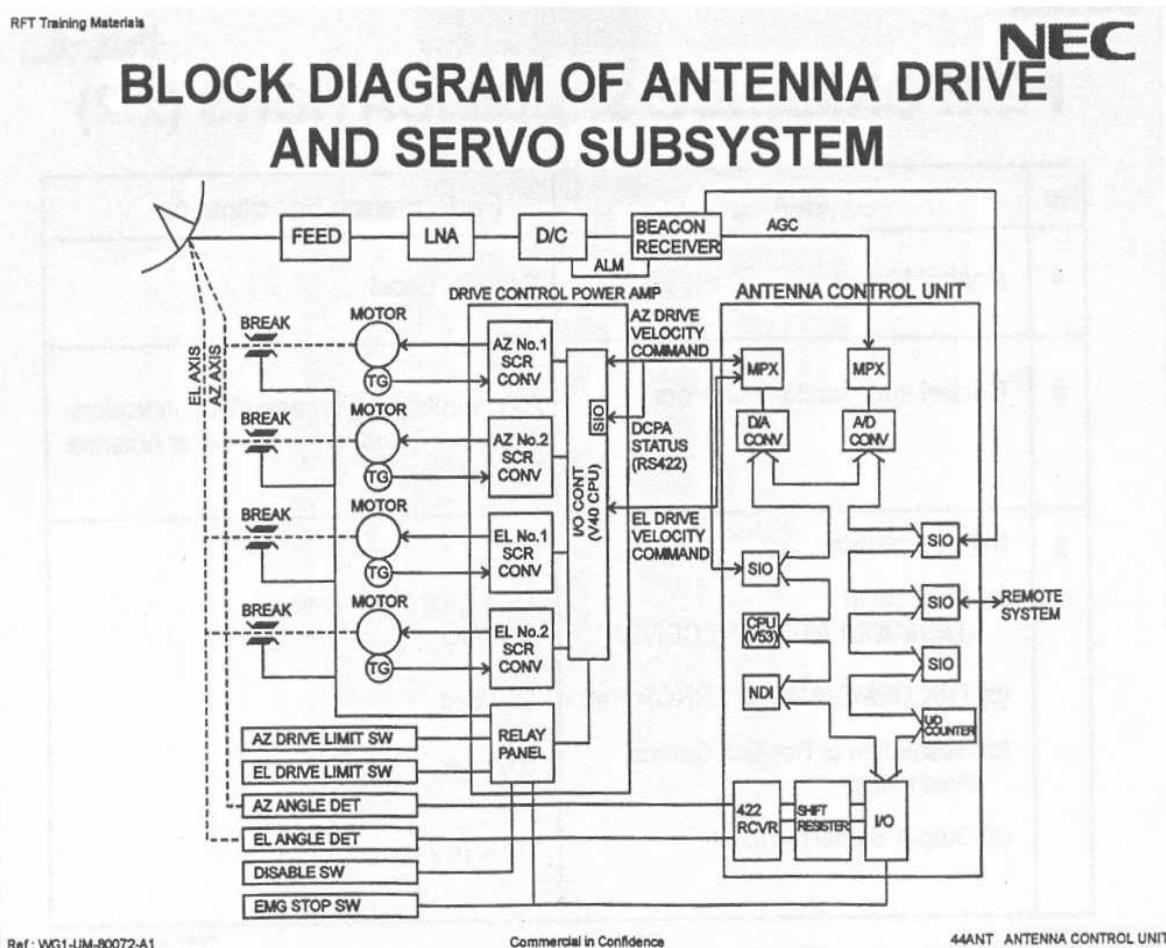


Figure 8.3: Overview of the existing control system (CBA, 1998)

A functional block diagram of the DCPA Control Board and Relay Board (for one axis) is shown in Figure 8.4. In the existing system the ACU supplies a  $\pm 10V$  DC velocity command voltage (labelled *AZ DRIVE VELOCITY COMMAND* and *EL DRIVE VELOCITY COMMAND* in Figure 8.3, and *VEL CMD* in Figure 8.4) for each axis to the DCPA Control Board. Limit switches and status registers interface to the DCPA Control Board, providing information on the antenna status. The DCPA Control Board provides four

command voltages for the servopacks (which drive the motors) and interfaces to the DCPA Relay Board, providing the control for clutches and brakes.

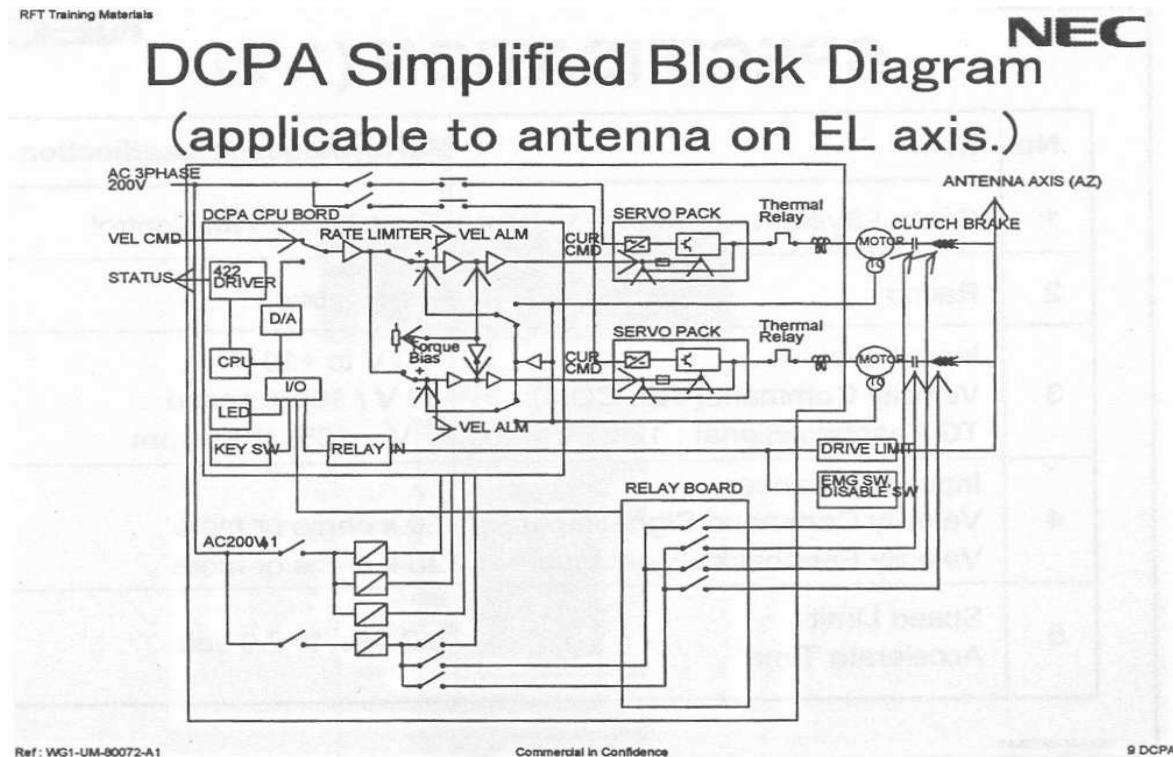


Figure 8.4: Functional block diagram of the existing DCPA Control Board and Relay Board for one axis (CBA, 1998)

### 8.3.2 Existing Hardware

This section provides background information on the existing hardware making up the components of the NEC antenna system.

#### Motors

There are 4 DC servomotors. They are positioned with two motors in opposition on each axis so as to minimise mechanical backlash effects.

#### Servopacks

The four servopacks (one used for each motor) require a DC  $\pm 10$ V velocity command voltage. They are used to provide a pulse-width modulated (PWM) drive for controlling the motors' speed.

#### Brakes

Each axis is equipped with a disc brake. They are fail-safe electric brakes which engage if current stops flowing through the internal solenoids. This means that a power failure

automatically engages the brakes. These are controlled by the DCPA Relay Board

### Clutch

A clutch is installed on each motor. It is engaged and disengaged by solenoid, in the same way as the brakes are. If the power (and current) is off the clutches will disengage. These are controlled by the DCPA Relay Board.

### Angle Encoders

The existing angle encoders are Heidenhain ROC416 rotary angle encoders (ROC). These are 16-bit absolute angle encoders that use an SSI data interface. This is a synchronous RS485 differential line interface where data is transmitted serially on the rising edge of a data clock. The interface allows for a data transmission clock frequency between 100 KHz and 1 MHz over a distance of up to 100 m. They currently interface with the ACU.

### Tachometers

Each motor is equipped with a tachometer that produces a DC voltage of  $\pm 7V/1000$  rpm. These velocity feedback signal are linked to the DCPA Control Board.

## 8.4 DC Motor Transfer Function

The DC motor transfer function is given by

$$\begin{aligned}\frac{\Theta_m(s)}{E_a(s)} &= \frac{K}{s(s + \alpha)} \\ \frac{\omega_m(s)}{E_a(s)} &= \frac{K}{(s + \alpha)}\end{aligned}\quad (8.1)$$

where  $E_a$  is the e.m.f applied to the motor,  $\Theta_m$  is the rotor position, and  $\omega_m$  is the rotor velocity. The constants  $K$  and  $\alpha$  are:

$$\begin{aligned}K &= \frac{K_t}{R_a J_m} \\ \alpha &= \frac{1}{J_m} \left( D_m \frac{K_t K_b}{R_a} \right)\end{aligned}$$

The motor constants (used to describe  $K$  and  $\alpha$ ) are:

- $K_t$ - the *motor torque constant*. This is the proportionality constant between the current through the armature and the torque produced at the rotor.
- $J_m$ - the *effective inertia* seen by the rotor.
- $R_a$ - the internal resistance of the motor.

- $D_m$ - the *drag constant* experienced by the rotor.
- $K_b$ - the *back e.m.f constant*. This is the proportionality constant between the rotor velocity  $\omega_m$  and the associated *back e.m.f*.

A thorough derivation of Equation 8.1 is included in Appendix C.

If we assume that the servo amplifier has a transfer function of the form

$$T_{servo} = \frac{K_{servo}}{s + \alpha_{servo}} \quad (8.2)$$

then the block diagram in Figure 8.5a shows a block diagram of the open loop system.

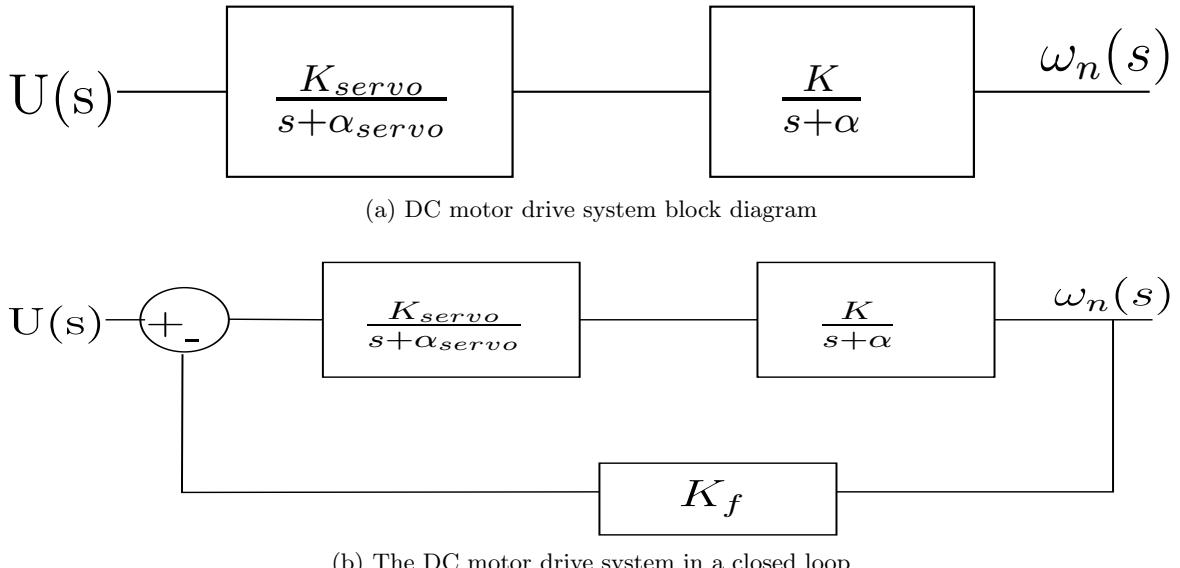


Figure 8.5: DC Motor servo systems

The open loop transfer function has the following form

$$\frac{\omega_m(s)_{open-loop}}{E_a(s)} = \frac{K_{servo}K}{(s + \alpha_{servo})(s + \alpha)} \quad (8.3)$$

$$= \frac{K_{servo}K}{s^2 + (\alpha_{servo} + \alpha)s + \alpha_{servo}\alpha} \quad (8.4)$$

and a closed-loop transfer function (Figure 8.5b) given by

$$G(s) = \frac{K_{servo}K}{s^2 + (\alpha_{servo} + \alpha)s + \alpha_{servo}\alpha + K_fK} \quad (8.5)$$

These are of the general form

$$G(s) = \frac{K}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (8.6)$$

Further information on analysing this system using frequency analysis and obtaining experimental values for the constants is included in Appendix D

## 8.5 Rate feedback

Gawronski (2002) examined a classic antenna control system (Figure 8.6) and performed simulations to determine whether it would be possible to get similar performance without using rate feedback. This would mean that the control system could be simplified, and as a result, would be less expensive.

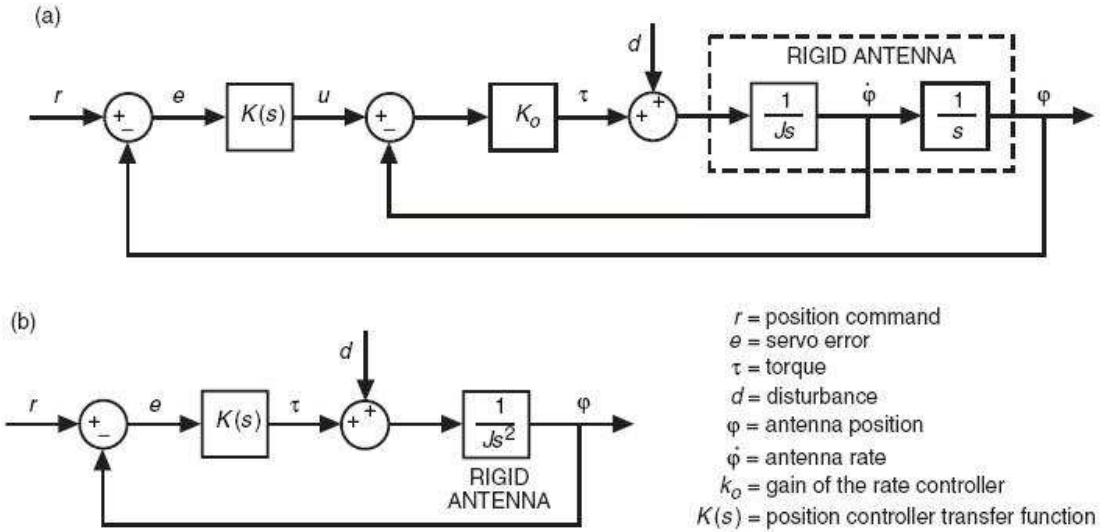


Figure 8.6: A classic antenna control system. (a) with rate and position feedback and (b) with position feedback only (Gawronski, 2002)

The results showed that both control systems could be used with success, however the rate feedback system showed better disturbance rejection. It is interesting to note that the model with position feedback only performed better in position tracking. The proposed system to be used at C-BASS will include rate feedback, although it is interesting to note that this may not be necessary.

## 8.6 Fundamental Frequency of an Antenna

The fundamental frequency of an antenna is defined as the first resonant frequency of the antenna. This is given roughly by its natural frequency  $\omega_n$  provided a second-order approximation of the transfer function is valid.

Gawronski and Souccar (2003) provides a plot, contained in Figure 8.7, which can be used to estimate the fundamental frequency. From the figure we see that for the 7.6 m diameter C-BASS antenna we expect  $\omega_n$  to be approximately 5 Hz ( $\omega_n \approx 31$  rad/s). Harp (2006) suggests that a good estimate of a small antenna's fundamental frequency, can be obtained by driving the antennas by hand! Vigorous shaking can be used to feed back the strongest resonance frequency and the frequency itself can be estimated by counting the vibrations over a period

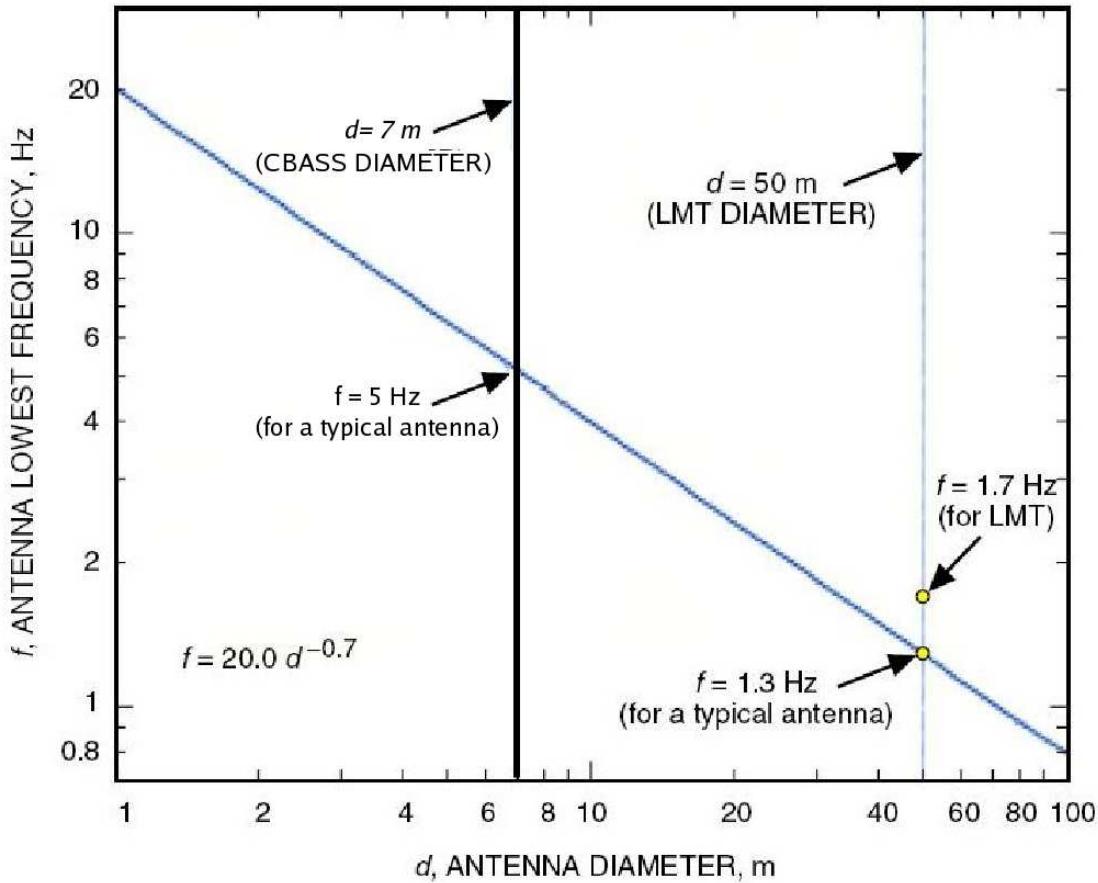


Figure 8.7: Antenna fundamental frequency versus antenna diameter. The plot is based on data from Lowest Servo Resonant Frequency Chart by the Aerospace Corporation (Gawronski and Souccar, 2003). It was used to show that the LMT is a rigid antenna since it has a higher resonant frequency than average. We can use this plot to estimate the 7.6 m C-BASS antenna resonant frequency as approximately 5 Hz.

of time. I have not been able to do this without releasing the motor brakes which needs the drive electronics to be installed.

## 8.7 Requirements of the New Antenna Control System

The requirement was to build a system that was inexpensive, simple to operate, reliable, and flexible. Available tools, such as the 8-bit ATMEL microcontroller, should be used and the resulting digital servoloop needed to refresh at 100 Hz to provide fast, accurate response. A pointing accuracy of 60 mdeg (1/10 of the 600 mdeg beamwidth) was also required<sup>1</sup>. There is a similar need for controllers with these characteristics on the *MeerKAT* array.

A further requirement was that the control system could utilise as much of the existing hardware as possible. This placed constraints on the design, which are summarised below:

<sup>1</sup>See Section 2.1.1 for information on required pointing accuracy

- Use of the SSI Angle Encoder interface to allow the use of the existing ROC416 Angle Encoders
- Use the existing safety interlock system
- Use the existing servopacks. This requires a velocity command voltage between  $\pm 10V$
- Analogue to digital converters able to record the  $\pm 7V/1000$  rpm tachometer feedback from the motors

## 8.8 Proposed Control System

The functional block diagram of the proposed replacement control system is shown in Figure 8.8. The proposed system comprises two main parts:

1. A control board will provide low-level control to the servopacks using feedback from the angle encoders and the motor tachometers to close the control loop. High-level control will be provided by from a PC using an Ethernet interface. A functional block diagram is shown in Figure 8.9.
2. A second control board which will control the motor clutches and brakes as well as monitor the interlock safety switches. This will interface via Ethernet to the same PC providing drive status information and control of the brakes/clutch. A functional block diagram is shown in Figure 8.10

I have developed a generic control board that can be easily adapted for use in either (1) or (2). Code was written in C rather than Assembly to ensure ease of upgrading and to make the code easily understandable. The system uses a simple Proportional-Integral-Derivative (PID) algorithm to provide a velocity command voltage, and external interrupts can be used to monitor interlock safety switches.

A board similar to the existing DCPA control system still needs to be developed (block labelled *MOTOR CONTROLLER* in Figure 8.8 and *VOLTAGE BIASING* in Figure 8.9) which will use the motor command voltage to provide suitable voltages for the two motors on each axis. A first suggestion would be to design a board which biased the command signal for one of the servopacks so that the two motors on each axis would run correctly relative to each other. This is the solution used in the original NEC system shown in Figure 8.4.

The existing code can be adapted to control the clutches/brakes using the same generic control board layout as the one used for providing motor control. An interface to similar relay boards (the block labelled *Brake Relay Board/Clutch Relay Board* in Figure 8.8 and

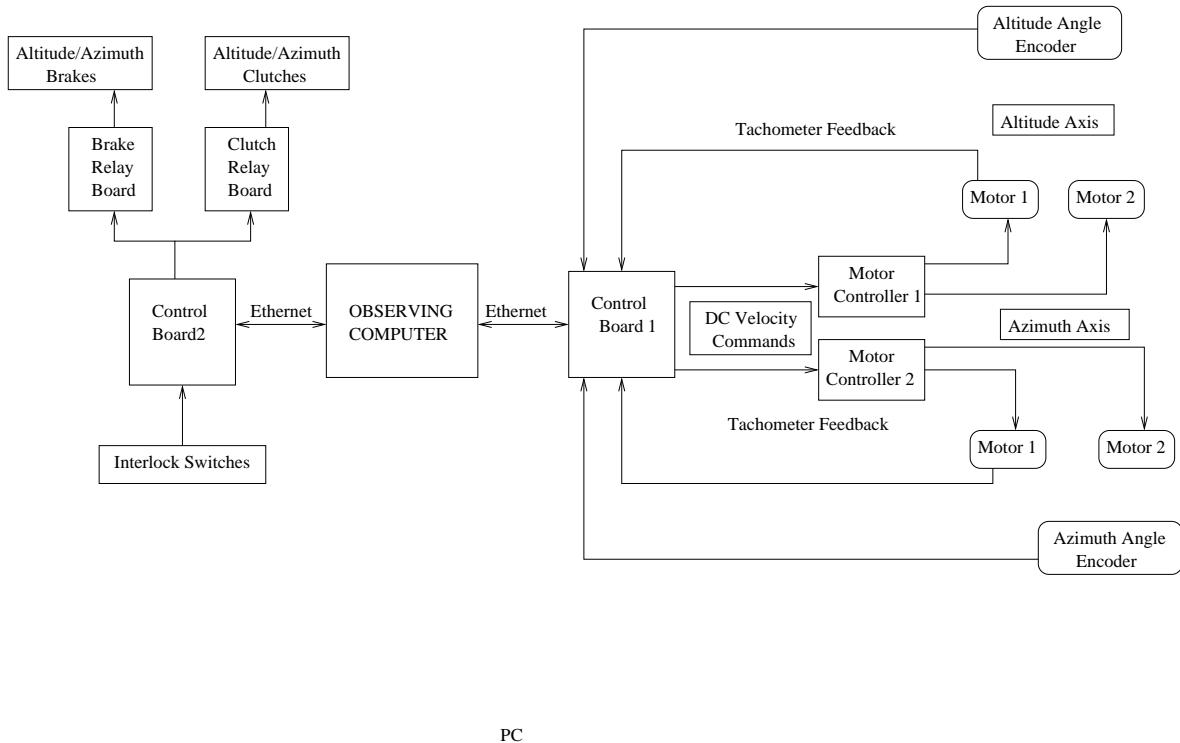


Figure 8.8: A functional block diagram of the proposed C-BASS control system.

*Brake/Clutch Relay Control* in Figure 8.10) as those used in the NEC system, needs to be built to control the clutches/brakes. Interlock switches can be monitored by external interrupts on the microcontroller.

### 8.8.1 Testing and Implementation Difficulties

I have not yet fully implemented the controller on the C-BASS antenna. The arrival of a new cherry picker has been unforeseeably delayed, and as a result the drive electronics for C-BASS could not be safely moved from the TELKOM site and have not yet been installed. However, initial testing of the controller on other test equipment is promising.

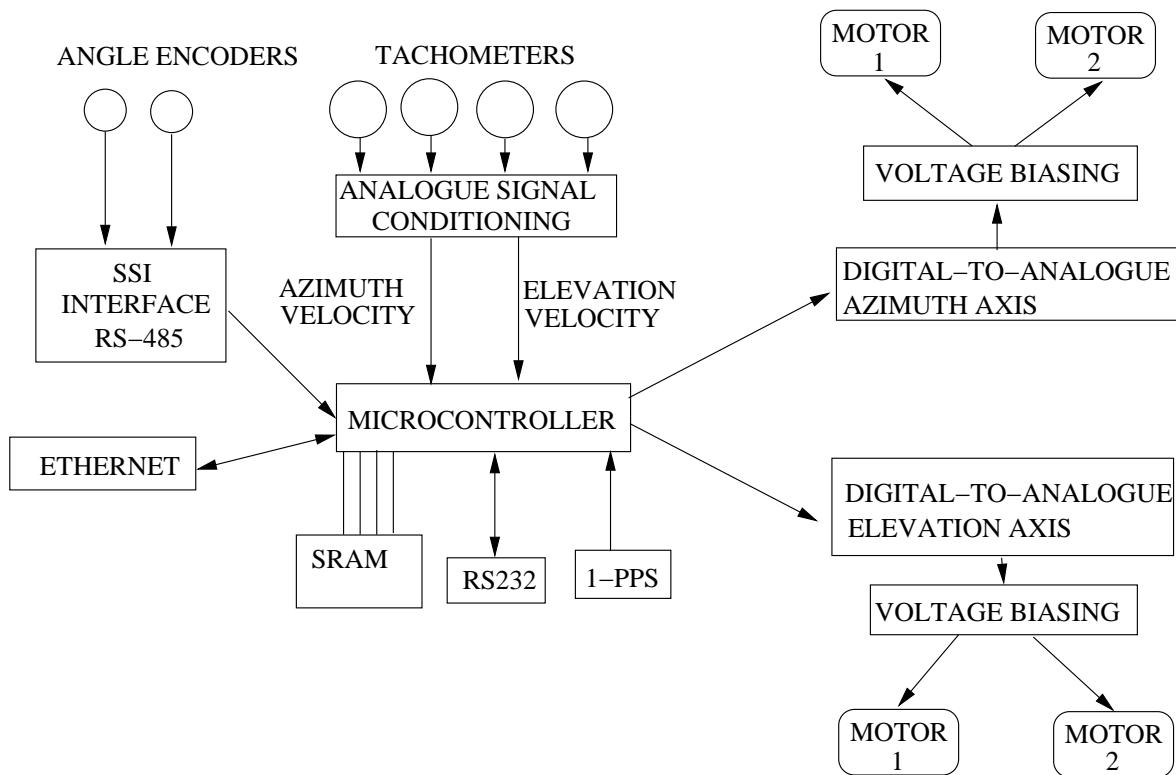


Figure 8.9: A functional block diagram of the Antenna Control board or Control Board 1 in Figure 8.8

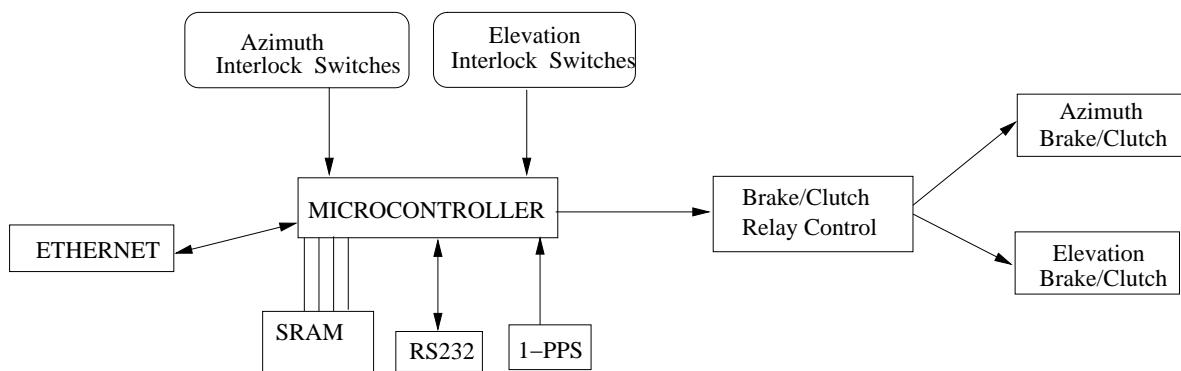


Figure 8.10: Functional block diagram of the control board to be used for monitoring the limit switches and providing control to the clutches/brakes. This is Control Board 2 in Figure 8.8

## 8.9 Overview of the Following Chapters

Chapter 9 gives overview of the board development together with descriptions of hardware and peripherals is then included and suggestions for future hardware upgrades are given.

A description of the layout of the embedded code is then provided in Chapter 10. This is provided primarily because this is a work in progress, and an understanding of the code will be useful to future developments of the control system. Suggestions for future improvements are made.

I then describe a PID tuning algorithm and present results in Chapter 11 obtained using the prototype control board on a small DC motor driven by a PWM drive. These results included control results and reliability results.

Finally, conclusions drawn from this part of the thesis are presented in Chapter 12.

## Chapter 9

# Architecture and Hardware

This chapter provides reasons for the choice of microcontroller for the C-BASS project as well as giving reasons for choosing Ethernet as the interfacing medium. Potential problems with Ethernet are outlined.

The chapter introduces the peripheral hardware used, and provides reasons for choosing these. It concludes with grounding information and a summary of the hardware development process, as well as providing ideas for future improvements of the system hardware.

### 9.1 The ATMega128 Microcontroller

I chose an 8-bit microcontroller as a solution since I had access to a readily available development environment. The ATMega128 is an 8-bit microcontroller produced by ATTEL. It has a maximum clock rate of 16 MHz, has six 8-bit I/O ports and 4 KB of internal SRAM. The 128 KB of internal flash memory makes it suitable for use with the relatively bulky programs written in C (as opposed to Assembly Language).

I chose the ATMega128 as it appeared to be the 8 bit microcontroller with the highest specifications across the board (number of I/O Ports, clock speed, flash memory, SRAM etc.).

The ATMega128 features a transparent interface to external SRAM, which was required because of the the relatively high memory demands of Ethernet packet buffers.

A drawback of the ATMega128 is that it is only available in a surface mount version. This made initial testing difficult, however once we had committed to a provisional board design this was no longer an issue.

### 9.2 Ethernet

#### 9.2.1 Why Ethernet?

Ethernet has formed a major backbone of local area networks since its introduction in 1973-1975. Ethernet is a widely accepted standard and, as such, is supported by a broad range

of hardware and software. Ethernet is robust, easy to install, well understood, inexpensive and readily available. Using Ethernet as an interfacing medium is a good choice for modern systems. Further information on the Ethernet standard is available in Appendix E.

### 9.2.2 Potential Problems Using Ethernet as the Interface Medium

A quick investigation on the Internet yielded a number of commercially available control systems using Ethernet, demonstrating that Ethernet has previously been used successfully for this type of application. I was not able to find any details of implementation of Ethernet as an interface medium in the scientific literature available to me, however there were obvious challenges that needed to be overcome.

The first of these is whether the microcontroller had enough processing power to maintain the sending and receiving of packets. Whilst not computationally intensive the packet construction and deconstruction requires a potentially large number of microcontroller clock cycles. To overcome these problems, the processing time was minimised as explained in Section 10.2.3.

The second was that of accurate time keeping needed to synchronise the data from the digital back-end and the controller. Since it is likely that a large portion of the processing power of the microcontroller will be occupied by packet sending and transmission, it is difficult to rely on internal clocks on the microcontroller over a long period of time. Besides this, time synchronisation between the server and the microcontroller is complicated by the packet nature of the Ethernet communication with the potential for variable propagation times (depending on the network load). In the case of a dedicated network this variability might be considered negligible, however an algorithm taking variability into account should be used to ensure that time synchronisation is maintained, together with a synchronising signal to make this synchronisation as accurate as possible. The solution used in this implementation is a combination of a 1-PPS signal and the SNTP protocol and is explained in Section 10.4.3.

### 9.2.3 Ethernet Controller

As mentioned previously, Ethernet is a widely available technology. The Ethernet Controller used in this project is the commercially available ENC28J60 (Rev. 4) chip manufactured by Microchip. A small board is commercially available that includes all the necessary external magnetics, crystals, resistors etc. to implement a complete, functioning Ethernet port. The schematic for this is shown in Appendix K, Figure K.1.

The interfacing to the microcontroller is done via an SPI interface. The commands available on the ENC28J60 are summarised in Figure I.1. The ENC28J60 has an 8 KB

internal SRAM buffer for storing Ethernet frames.

### 9.3 External SRAM

The ATMega128 features a transparent External Memory Interface which allows up to 64 Kb external memory to be addressed in exactly the same way as the internal memory. External memory is required due to the relatively high memory use of Ethernet communication. The GCC compiler allows different memory configurations to be used. I elected to place the heap (memory declared by the malloc() function) on the external memory. The internal memory is dedicated entirely to the stack and internally declared variables. This is done to minimise the risk of stack overflow errors.

The external ram used is the 128 Kb CS18LV10245 from Chiplus (Chiplus, 2004). Only the first 64 Kb is used because of the 16-bit limitation of the External Memory Interface addressing scheme. A zero wait state is required to successfully write and read from the SRAM, and the ATMega128 Buskeeper is enabled. This ensures that a defined logic level is always present on the External Memory Interface address lines (AD7:0). Another note is that, contrary to the specifications detailed in the ATMega128 datasheet, I did not have any problems with the 74HC series latch. The 74HC latch is used successfully at 16 MHz without any problems.

### 9.4 Digital to Analogue Converters (DAC)

The existing hardware requires an analogue  $\pm 10V$  DC control voltage. This requires a bipolar DAC that is capable of generating these voltages. I have chosen to use the 12-bit Analog Devices AD667 (Analog Devices, b) which has the following desirable characteristics:

- 12-bit resolution
- Easy configuration to the bipolar ( $\pm 10V$ ) output
- Nibble-by-nibble data uploads (only 4 data lines and 2 address lines required)
- Parallel Interface for high speed data transfer.

While it may appear wasteful to use a parallel interface for the DAC given the availability of other interfaces which would use less I/O space, at the design stage I was not sure how much time would be available between samples to update the DAC and to read the A/D. At the same time I needed to minimise the use of I/O lines. This DAC (with parallel nibble-by-nibble data uploads) seemed a reasonable compromise. Grounding the DACs properly is important if they

are to meet their quoted specifications. Information on correct grounding was obtained from (Analog Devices, b). Initially I had not followed these suggestions carefully and noise levels of  $\approx 200$  mV peak-to-peak were present in the DAC output. I have subsequently connected the analogue 'high quality' ground to the digital ground on a separate grounding plane (i.e a thin piece of aluminium). I have referenced the ATMega128 ground to this plane. This change has resulted in a noise level drop to approximately 20 mV peak-to-peak on the DAC outputs. Further improvements could be made, however the servopack responses need to be tested to see whether any further improvements would be noticeable.

## 9.5 Angle Encoders

The beamwidth of the 7.2 m C-BASS antenna at 5 GHz is approximately 600 mdeg ( $\Theta_{HPBW} \approx \frac{1.2\lambda}{d}$ ). It is possible that the final receiver design will require that the antenna be under-illuminated, reducing the effective diameter and increasing the beamwidth. Part 1 of this thesis mentions that pointing accuracy needs to be less than 1/10th of the beamwidth on a typical radio telescope suggesting a pointing accuracy of approximately 60 mdeg.

The bit resolution of angle encoders is not necessarily the same as the effective accuracy, and a number of effects, such as quantisation performance and axis misalignment, reduce the accuracy of measurements. An angle encoder with a 16-bit resolution will typically have a guaranteed accuracy of 14-bits (approximately 20 mdeg), so a typical 16-bit angle encoder should provide sufficient effective accuracy to meet requirements with some margin allowed for other pointing effects such as those outlined in Part 1.

The existing angle encoders are ROC416 HEIDENHAIN 16-bit encoders which use an SSI interface. Since at least one of the original ROC416 angle encoders was not functioning, I investigated options for replacement. Replacement encoders should have the same dimensions as the ROC416s and should use the same interface if possible. This will allow other existing hardware (e.g physical mounts and shaft couplers) to be used as far as possible. I chose the FRABA OCD-SL00B-0016-S10S-CRW 16-bit angle encoders. These have the same physical dimensions as the ROC416 encoders and use the same SSI interface. Datasheets are included in Appendix K.

## 9.6 RS-485 Drivers

The angle encoders have a SSI (RS-485) interface. This implies that the control board would require RS-485 transceivers to be built in. Additionally, if possible, transceivers that could be enabled and disabled would allow for the data lines to be multiplexed to allow communication

with both angle encoders using minimum I/O space on the microcontroller. I have chosen to use the ADM489A transceivers to provide the RS-485 interface. These feature enable/disable functionality for both transmitting and receiving which allows for multiplexing.

## 9.7 Analogue to Digital Sampling

The antenna control system requires velocity feedback as shown in Figure 8.8 and Figure 8.9. In the existing equipment a tachometer signal from the motors is present in the form of an analogue voltage. This needs to be digitised using an analogue to digital (A/D) converter to provide velocity feedback to the PID control loop running on the microcontroller. Two options exist to do this:

- Use the internal 10-bit ATMega128 A/D converter
- Add separate A/D converters to the board

While the second option does have a number of advantages (primarily noise reduction), the first has been provisionally chosen. Using the internal 10-bit A/D converters reduces the number of required external peripherals and 10-bit is considered to be a high enough resolution. Signal conditioning analogue circuitry (shown in Figure 9.1) is used to bring the tachometer voltage to the required range using the ATMega128 A/D onboard 2.56 V reference. This circuitry provides a voltage offset as well as amplification to bring the voltage into the  $0 \rightarrow 2.56V$  range. It requires low-noise, high stability operational amplifiers. After consultation with the electronics staff at HARTRAO I have decided to use OP177 high precision operational amplifiers. These are suitable due to the following features

- $25 \mu V$  offset voltage at  $25^\circ C$
- $0.1 \mu V/^\circ C$  offset voltage drift

This means that the analogue circuit should remain suitably stable over both time and temperature.

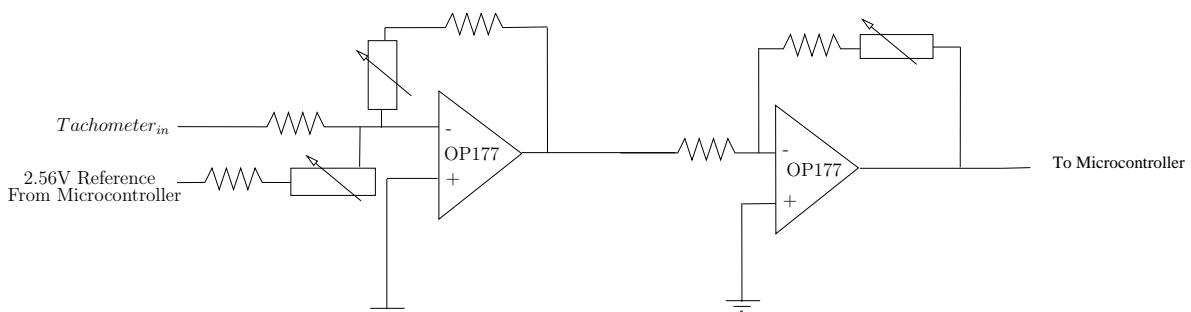


Figure 9.1: The signal conditioning circuitry used in the tachometer sampling

I have included sufficient analogue circuitry to sample two external voltages. The input range can be adjusted by resistor selection, and fine tuned using variable trim resistors. This design uses resistors optimised to the expected tachometer voltage ranges, however other voltage ranges can be used with suitable adjustment of the resistor values. Further details of the A/D analogue circuitry, tuning the variable resistors and selecting resistor values for other input voltage ranges are included in Appendix I.4.

## 9.8 RS232 and UART Settings

An RS232 port allows for a secondary channel of communication to the board and was used extensively in debugging. The UART settings implemented are specified in Table 9.1.

Table 9.1: RS232 format used for communications in the final version of the code.

Baud Rate	38400
Parity	none
Data bits	8
Stop bits	1

The UART-RS232 voltage level conversion is done using a *MAX232*.

## 9.9 Prototyping and Board Design

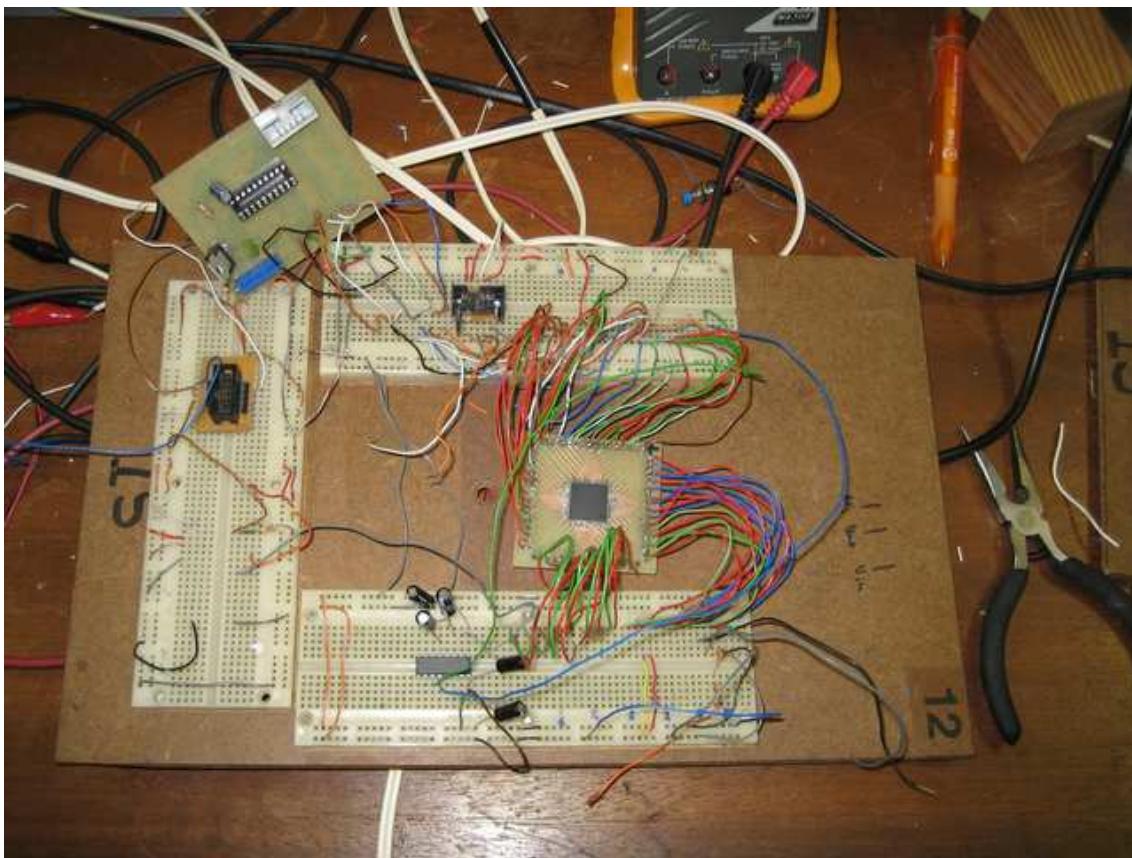
An initial hardware prototyping environment was required to investigate the interfacing signals of the microcontroller and peripheral devices. The ATMEGA128 is a surface mount device which made testing difficult since it could not be plugged directly into a breadboard. I decided that the best way to overcome this was to build a small PCB onto which just the device could be mounted. The PCB provided substantial pads for the connection of external hook-up wiring that provided the necessary connections to a breadboard. Experiments began on the equipment shown in Figure 9.2a. This was far from ideal as it proved extremely unreliable, particularly at high clock speeds, however it did provide insight of how the I/O ports would need to be distributed in the first prototype board.

I began design once I had confidence that I understood what was required and how the prototype board should be laid out, . The majority of the ATMega128 I/O lines were connected to pin headers with the idea that connections could be soldered in later. Other board hardware was laid out in the same way.

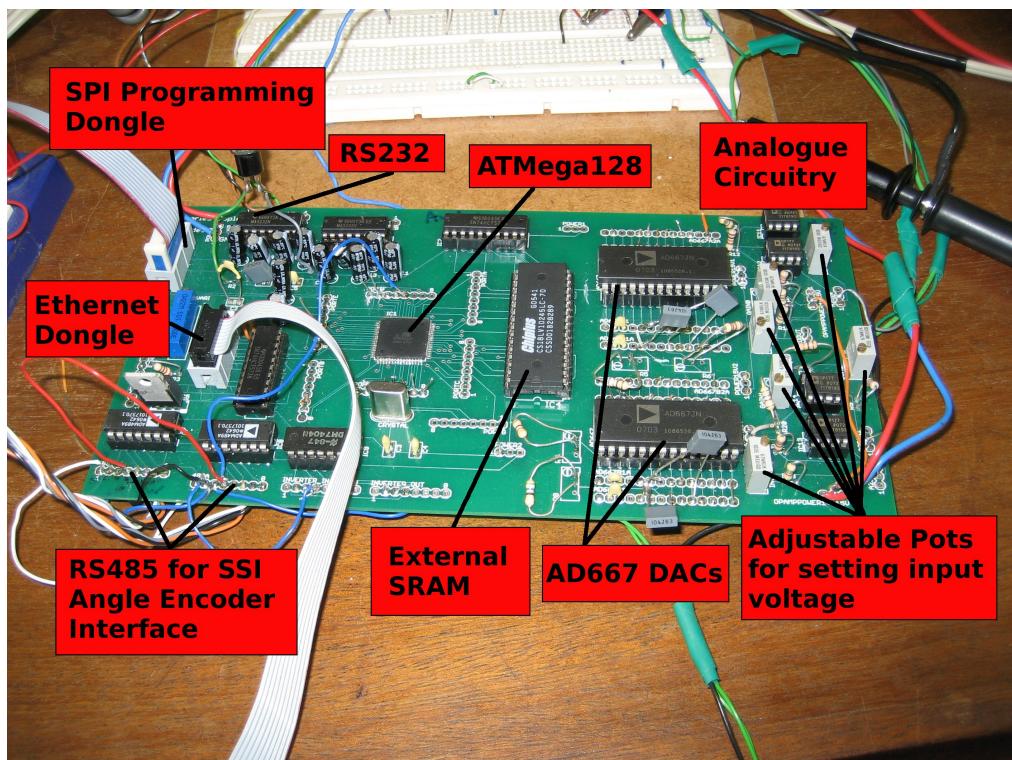
Five initial prototyping boards were produced by Mastercircuits in Johannesburg. A photograph of the board is shown in Figure 9.2b and a block diagram is given in Figure 9.3. Printed circuit board diagrams are included in Appendix G and the Gerber files and board schematics (in Eagle format) for the prototype board (together with an upgraded prototype

board design) are included on the accompanying CD. A description of the external jumpers required to complete the circuit is included in Appendix H. Further detailed information on the various board components is included in Appendix I.

The prototype design has proved to be flexible and adaptable. Since the complete C-BASS drive electronics has not yet been moved to the HARTRAO site, testing was not possible on the antenna, however testing was performed on a small DC motor system with a similar PWM drive (see Chapter 11). Although a revised board will eventually be required for robustness, the current design has proved to be adequate as a testing platform. The board is likely to be useful in other, perhaps unrelated developments for future Rhodes students.



(a) The original test equipment. Notice the large number of wires which resulted in testing difficulties and unreliability.



(b) The prototype controller board with labels of the external devices installed.

Figure 9.2: The evolution of the prototype board

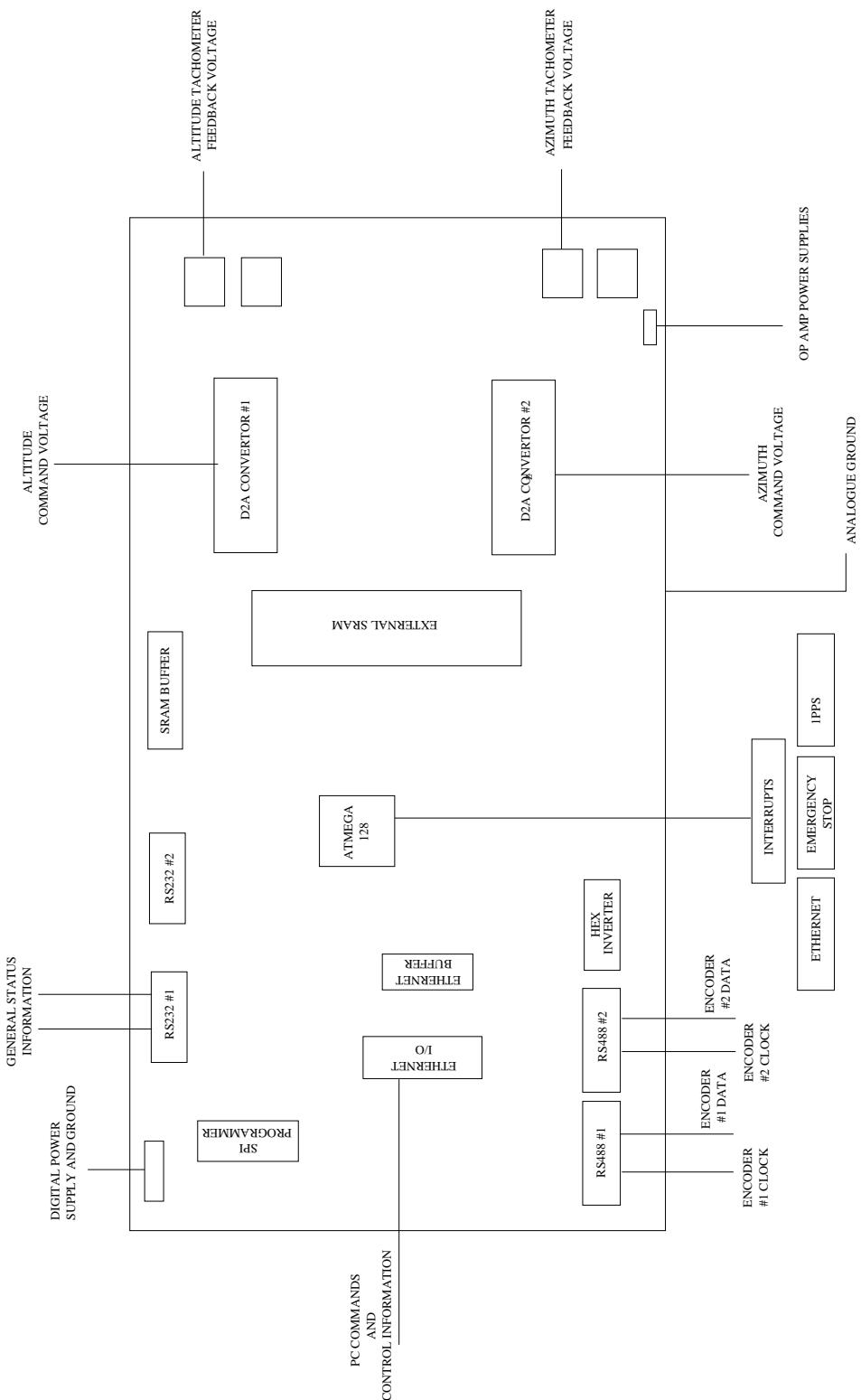


Figure 9.3: Functional block diagram of the prototype control board including component placement. A photograph of the board is shown in Figure 9.2b for comparison.

## 9.10 Future Hardware Improvements

### 9.10.1 DAC

After examining the board used by the SMA antennas (Hunter et al., 2001), I realised that it might prove useful to use 4 DACs rather than the 2 that I initially designed for. This would allow for better individual control of the motors. A possible advantage would be an easily implemented slew mode, where both motors drive in the same direction. This would allow for rapid movement during observations, with the possibility remaining of switching back to the anti-backlash mode for finer control when required. While this could be done on this prototype board, it would be difficult.

The AD5625R/AD5645R/AD5665R series DACs from Analog Devices, features an I<sup>2</sup>C (TWI) serial interface which is capable of interfacing at 330 kHz. The series features four digital outputs on each IC. Different versions allow digital to analogue conversion at up to 16-bit resolution, and a suitable bipolar output is easy to configure using external circuitry. This would be a suitable future upgrade of the board.

### 9.10.2 A/D

Because it was not possible to test the board on the antenna, it was not possible to confirm whether the 10-bit ADC on the ATMega128 was sufficient for tachometer feedback. It might prove necessary to add additional external ADC's to the board. This would allow for individual motor tachometer feedback, rather than one single motor tachometer feedback for each axis. An example of a suitable upgrade is the AD7982. This features an SPI interface (which can be used for both the ENC28J60 and the A/D), 18-bit resolution and the option of daisy-chaining multiple A/D's and reading them sequentially. This would solve all problems of limited I/O lines and is a suitable candidate to be considered for a future upgrade.

### 9.10.3 Trim Resistors

Depending on future tests it may be desirable to replace the current manual potentiometers used as trim resistors by digital potentiometers. This would allow the offsets to be dynamically controlled over the network, reducing manual intervention and upkeep of the controllers. Currently there is likely to be a small drift of these offsets over time. The AD5206 IC features 6-channel outputs and is available in 10 K potentiometer packages with 256 positions on each channel. This gives a resolution of 39 ohms and could be used as trim resistors for configuring the external A/D circuitry. The interface is SPI. An option for the 1 K potentiometers used as trim resistors on the DACs is the AD8403. A 1 K version is available with 4 channels and 256 positions per channel giving a resolution of 3.9 ohms. This features an SPI interface.

What is particularly interesting with both of these options is the price. From a preliminary investigation, both IC's are less expensive than the manual potentiometers counterparts.

#### 9.10.4 Grounding

The analogue and digital parts of the board are separated from each other. The grounds of both are connected to a separate grounding plane in an effort to minimise digital return currents in the analogue ground reference. There is a small potential design flaw in that the A/D grounding is connected to the ATMega128 ground. The ground signal is first passed through a low-pass RC filter to remove digital noise, however I think it would be better to provide the A/D ground directly from the analogue ground in future upgrades.

From brief preliminary investigations, the noise on the DAC voltage is approximately 20 mV peak-to-peak. Installation on the C-BASS antenna will give us a better idea as to whether this is an acceptable level for providing control to the system. I have decided not to try to improve this level until this has been tested.

I was careful regarding grounding in the design of the prototype board, however I believe grounding could be improved on considerably in the future. More care could be taken to ensure that there is minimal contamination of the analogue ground signal by the high frequency noise present in any digital ground caused by the return currents of digital signals. I did not consider grounding issues to be of primary importance in the development of this board, as it is a proof-of-concept design, however future designs need to consider this.

Ott (2001) summarises the primary grounding concerns as minimising the size of current loops (especially the path that a return current must take through the grounding plane) and ensuring that the system has only one reference plane. These lessen the loop and dipole antenna effects respectively, thus reducing interference in the analogue ground.

The article suggests separating the digital and analogue components, but suggests using one common ground plane, and not separate digital and analogue ground planes as is often suggested elsewhere. This seems a poor way of solving the problem of contamination between digital and analogue grounds. However the article shows that the return path of the contaminating digital return current will be on the ground plane directly beneath the signal trace, because this is the path of lowest impedance (lowest inductance) for a high frequency signal. For this reason there will be negligible contamination of the analogue ground, provided the digital components are separate and that no digital signal paths cross into the analogue section.

Components requiring both analogue and digital grounds (DACs, A/Ds) should straddle the digital/analogue partition line and care should be taken to avoid routing digital signals

through the analogue section as this would cause digital contamination of the analogue ground plane below the signal path.

#### **9.10.5 16 and 32 bit microcontrollers**

This investigation deals with using 8 bit microcontrollers and investigating their suitability to provide low level control. It is likely that 16 and 32 bit microcontrollers would be even more successful. The option of installing an actual operating system on 32 bit microcontrollers ( $\mu$ CLinux) is attractive. The use of these microcontrollers would, however, increase the cost and complexity of the system.

# Chapter 10

# Software

Explaining embedded software can be difficult, and is often not satisfactory because of the low-level detail in the code. The system has not yet been implemented on the C-BASS antenna and as such someone needs to take this project forward. This chapter attempts to justify and explain the code processes being implemented as clearly as possible. An emphasis is placed on flow diagrams and code extracts.

The choice of C as the programming language is explained, coding aspects such as failsafes, timing, the servo-loop, and the control algorithm are dealt with and the Ethernet stack is outlined. The chapter concludes by summarising possible future code improvements.

## 10.1 C Programming Language

While assembly code may be the optimal language for microcontroller programming in terms of speed and code space optimisation, there are many disadvantages associated with this very low-level language. In terms of development time, assembly coding is slow, and maintenance of the code is difficult. Upgrading cryptic assembly code presents a particularly difficult problem for someone not involved in the development of the original code, and refactoring experimental code is equally challenging.

Since the controller was unlikely to perform optimally in the initial stages, ease of upgrade and ease of understanding were of great importance. It was unlikely that the initial code would be sufficient for final specifications and I therefore decided to proceed using the C language.

C is a flexible and feature-rich language that allows low-level access to the processor. It is therefore particularly suited to microcontroller programming. An open source C compiler (AVRGCC) for ATMEL AVR microcontroller family is readily available and has proved to be reliable. C is the preferred programming language at HARTRAO, thus ensuring compatibility with coding standards at the observatory. Another advantage is the existence of C drivers for the ENC28J60 Ethernet Controller (which gave a foundation to build on, although they did require slight alterations).

Disadvantages of C as opposed to assembly code are few, but potentially highly problematic. Depending on the level of optimisation used during compilation, polling loops may be automatically discarded. This means that all timers should be interrupt driven . This is a problem when dealing with short timer requirements due to the overhead required in initialising the interrupt.

A second disadvantage is the bulky nature of C code compared to Assembly. On microcontrollers with little internal flash this may necessitate the use of Assembly, however the ATMega128's 128 Kb of internal flash memory proved to be sufficient.

## 10.2 Control Process and Data Flow

High level angle transformation computations are handled by the Observation PC with the low-level servoloop handled by the microcontroller. The servoloop uses position commands (from the PC) and system feedback (in the form of angle encoder positions and tachometers) to drive the antenna to a desired position. A high-level diagram describing the servoloop is shown in Figure 10.1.

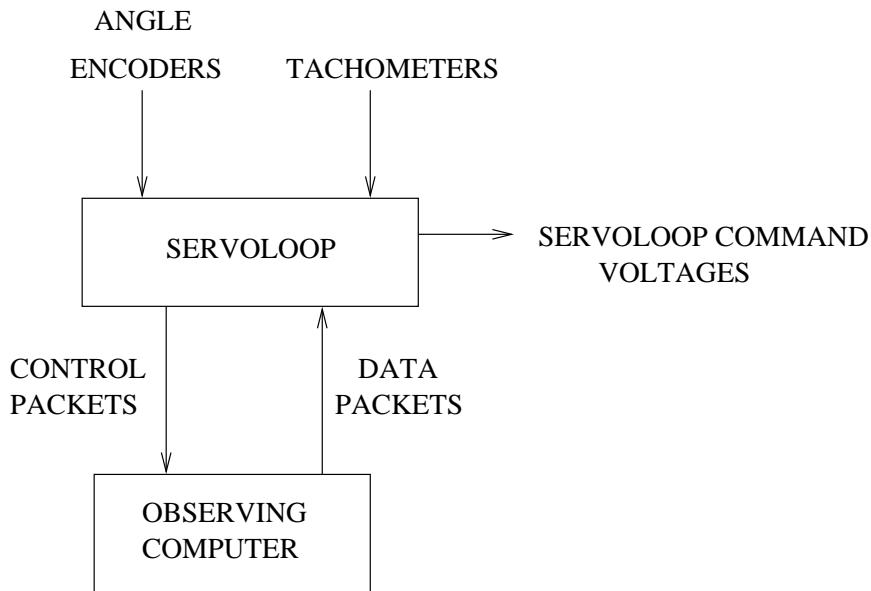


Figure 10.1: The layout of the servoloop system

A low-level diagram of the servoloop process is shown in Figure 10.2. The servoloop samples at 1/100 s intervals using the internal Timer Counter 3C on the ATMEGA 128. This sample interval was chosen following the rule of thumb given in Westcott (2000) which suggests a sampling interval of between 1/10→1/100 of the desired settling time. This sample interval of 1/100 s will give a settling time of between 0.1→1 s. As a first estimate this is sufficient. It may, however, be necessary to increase the sampling rate in the future if a shorter

settling time is required.

Once a second, a control packet containing control angle information is uploaded to a temporary buffer. This data is transferred to the Command Angle Buffer at the start of the next second (determined by the 1-PPS signal), and provides the necessary  $2 \times 100$  incremental angles needed to maintain a position on the two axes for the following second. The servoloop extracts the correct two command angles from the buffer (one for each axis) every  $1/100$  s, and records angle encoder readouts and tachometer voltages as system feedbacks. A PID control algorithm uses the command angles and the system feedbacks to send appropriate voltages to the servopacks via the DACs.

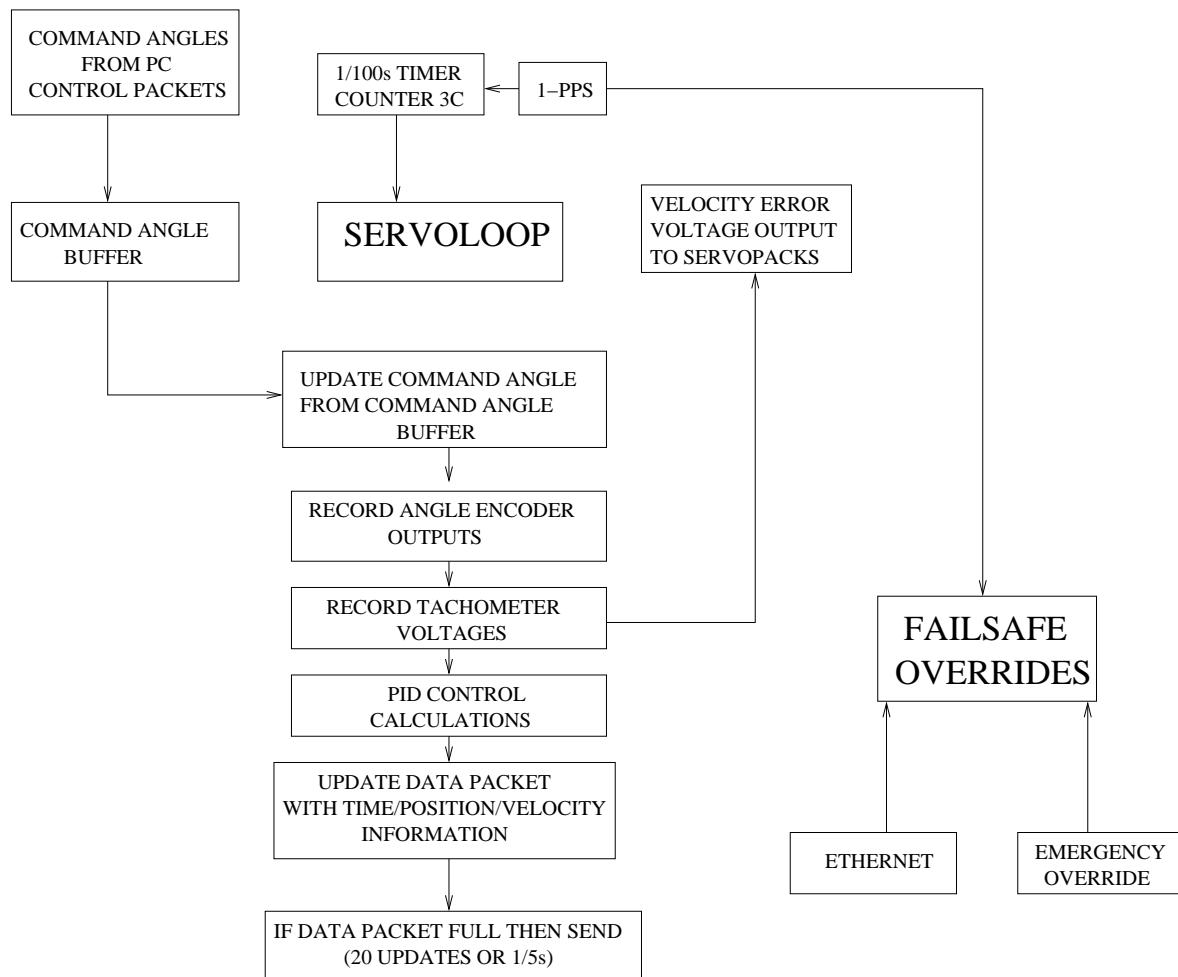


Figure 10.2: Low-level diagram of the servoloop

### 10.2.1 Control Board Data Transfer

As shown in Figure 10.1, there are two types of UDP data packets that are expected to go between the PC and the microcontroller. The first are *data packets* which have an extended UDP header consisting of a two byte sequence number (used as a primitive packet loss error check) and a two byte data packet tag as an identifier. An example from the code is shown below.

```
from udp.c - udp_send()
.....
// tag with a sequence number
*(udp_packet_ptr + 21 )=sequencenumber;
// tag udp_packets with UDP_PACKET flag 0x00e1
*(udp_packet_ptr + 22 )=UDP_PACKET; /
// insert data into the packet - 2 byte variables
for (cntr=0;cntr<len;cntr+=1)
{
    *(udp_packet_ptr+23+cntr)=(*(data_ptr+2*cntr))+(*(data_ptr+2*cntr\
    +1)<<8);
}
.....
```

The addition of the UDP\_PACKET tag allows for the PC to easily distinguish these as data packets.

The second class is a *control packet*. These are constructed in the same way but without the UDP\_PACKET tag.

```
from udp.c - udp_send_command()
.....
//tag with a sequence number
udp_send_command_packet[21] = sequencenumber;
//insert data into the packet- 2 byte variables
for (cntr=0;cntr<len;cntr+=1)
{
    udp_send_command_packet[22+cntr]=(data_ptr[2*cntr])+(data_ptr[2*\
    cntr+1]<<8);
}
.....
```

### 10.2.2 Control Board Commands

An important aspect of the controller is the command set used in the *control packets*. A considerable effort was made to make this as flexible as possible to allow future upgrades. All commands/requests sent (from either the PC or the microcontroller) have a unique identifier byte. All replies sent from either side have a similar unique identifier byte. The code is set up in such a way that commands can be added to the microcontroller repertoire with a minimum fuss. No existing code alterations are necessary. An addition of the *command identifier* and the resulting execution code is all that is necessary. The *command identifier* tag should be

added to the *process\_packet.h* header file as a definition. These unique byte identifiers from *process\_packet.h* are summarised in I.2.

The actual handling of received packets is taken care of using the `Process_packet()` function found in *process\_packet.c*. Received packets can be either COMMAND or COMMAND\_REPLY type packets as shown in the code extract below.

```
Extract from process_packet.c -Process_packet()
void Process_packet(int len, unsigned char *packet){
extern char netstatus;
unsigned char MYPORtchar[2];
////////// INCOMING UDP DEFINITION /////////////
//get the PORT number into hexadecimal for comparison if UDP data \
arrives
MYPORtchar[0]=(MYPORt>>8);
MYPORtchar[1]=(MYPORt <<8)>>8;

if((packet[12] ==0x08))
{
// This is an Ethernet packet and we should look at it

    if((packet[30] ==MIP0) && (packet[31] == MIP1) && (packet[32] \
    ==MIP2) && (packet[33] == MIP3))
    {
        //this will be an IP packet i.e UDP , ICMP , NTP
        if((packet[36] == MYPORtchar[0]) && (packet[37] == \
        MYPORtchar[1])){
            //this is a command control or command reply packet
            switch (packet[42])
        }

        case COMMAND:
            //a command packet
            command(packet[43],packet);
            break;

        case COMMAND_REPLY:
            //a command reply packet
            command_reply(packet[43],packet);
            break;
    }
    .....
}
```

If any action is required after a particular PC COMMAND Packet is received, then the action should be described in the `command(char COMMAND, char *packet_ptr)` function found in *process\_packet.c*. The `command(char COMMAND, char *packet_ptr)` function takes action depending on the COMMAND input. Similarly if a PC command reply requires action by the microcontroller, this can be specified in the `command_reply(char COMMAND, char *packet_ptr)` function. The layout of the code is fairly self explanatory and an extract from `command(char COMMAND, char *packet_ptr)` is included below. This shows the actions taken when a RESET command is received by the microcontroller.

1. Send “RESET” to UART and RS-232
2. Send acknowledgement via Ethernet to Observation PC
3. Set the DACs to zero
4. Reset the microcontroller using the Watchdog Timer

```

extract from process_packet.c- command(char COMMAND, char *packet\
_ptr)
...
    switch(COMMAND)
{       case RESET:
            printf_P(PSTR("RESET\n"));
            // send an acknowledgement out of courtesy
            udp_data[0] = COMMAND_REPLY;
            udp_data[1] = RESET;
            udp_send_command(2,udp_data);
            // set the DACs to zero!
            d2a_altitude(0);
            d2a_azimuth(0);
            // reset the microcontroller using the watchdog timer
            WDTCSR |=(1<<WDE);
            break;
            .....
}

```

### 10.2.3 Command Angle Buffer

As mentioned in Section 10.2, it is necessary to maintain a command angle on the microcontroller for each servoloop sample without sending these continuously from the PC. Doing this would result in large overheads on the microcontroller as it would have to unpack these packets and would mean that a large amount of potentially unnecessary data was being transferred along the Ethernet. I decided on two different ways of achieving this.

1. The PC would calculate two desired command angles for two subsequent seconds. From this a desired command angle change could be calculated and the related command angle increment per sample interval (i.e 10 ms) could be calculated. The PC would upload the initial desired command angle together with the required command angle increment, and the microcontroller would use these to update its desired command angle at each sampling instant.
2. The PC would calculate two desired command angles for two subsequent seconds. The PC would then calculate the angle increment necessary per sample interval (10 ms) and would calculate the command angle for each axis during each servoloop sample. These  $2 \times 100$  command angles would be sent via a Ethernet packet and, would all be loaded onto a temporary buffer on microcontroller sometime during a second. At

the beginning of the subsequent second they would be transferred from the temporary buffer to the command angle buffer. During each servoloop sample a pointer is simply be incremented to update the control angle to the correct one. This was implemented primarily to allow for the control system to be tested with step responses which would not have been possible using the other method.

In the first method the choice of numerical calculation method was complicated by the relatively high frequency at which samples were taken (i.e 10 ms). While the microcontroller is capable of floating point calculations (and these are simple to implement in C), these floating point calculations are implemented in the software and not hardware. As such a typical calculation required approximately 1 ms in this case. This not only meant that the available time between samples was being reduced, it meant that the servo-response was being delayed. This had potential implications to our approximation to a continuous system and was to be avoided if possible.

To combat this integer arithmetic was implemented and reduced the apparent calculation time to approximately 0.1 ms. This method of calculation was employed in every similar situation (including the command angle updates) in an effort to reduce numerical calculation overheads.

The disadvantage of the second method was the relatively large memory requirements. Once external SRAM was added this was no longer a problem.

Both methods were kept as options in the microcontroller code, however the second method is preferred and is shown in Figure 10.2 and is described in Section 10.2. The actual choice is recorded to EEPROM to maintain the correct option even after rebooting.

#### 10.2.4 PID Control Algorithm

An easily implemented PID control algorithm is used in this software. It was adapted from the algorithm used by Westcott (2000) to use integer arithmetic in order to lower the microcontroller overhead.

```
long UpdatePID(SPid *pid, short error, unsigned short position, short velocity){
    long pTerm, dTerm, iTerm, ptest, error_long;

    pTerm = 0;
    dTerm = 0;
    iTerm = 0;
    long output;
    // multiply P by a constant using integer arithmetic
    pTerm = ((long)pid->pGain * (long)error) >> (pid->p_integer);
    // calculate the I with limiting factors to prevent potential \
    integral wind up!
    pid->iState += error;
```

```

        if(pid->iState > pid->iMax){
            pid->iState=pid->iMax;
}
else if (pid->iState < pid->iMin)
{
    pid->iState = pid->iMin;
}

// multiply I by a constant using integer arithmetic
iTerm = ((long)pid->iGain * (long)pid->iState)>>(pid->\i_integer);

// multiply D by a constant using integer arithmetic
dTerm = ((long)pid->dGain *((long)position - (long)pid->\dState))>>(pid->d_integer);

// save the current position (or velocity depending on the \
control system) to obtain an estimate of the derivative for \
the next loop.
pid->dState = position;

output = pTerm + iTerm -dTerm;
//make sure there is no overflow since the output should be \
in the interval -32768 ->32767!
if(output >32767){
    output = 32767;
}
if(output <-32768){
    output = -32768;
}

return output;
}

```

The PID algorithm calculates its output by using the error between control signal and feedback signal. The output is a weighting of the error's actual value (P), an integral of the error (I) and a derivative of the error (D). A potential problem is integral windup where the integral value accumulates to such a large value that response becomes slow. This is avoided in this code by limiting the integral value to always be between certain limits (PID->iMin and PID->iMax).

### 10.2.5 Failsafe Overrides

An Emergency Override Interrupt will reset the microcontroller and stop the servoloop if triggered. The microcontroller has built in protection should the Ethernet interface fail. If an Ethernet packet is not processed by the microcontroller at least every second, the motor control lines are zeroed and the ENC28J60 is reset. If this does not succeed in reestablishing communication within 5 seconds, then the microcontroller is reset.

A further failsafe is carried out on the 1-PPS signal. If this is not present, the microcontroller will zero the motor control lines and reset itself.

### 10.3 Power Up Process

On power up the following conditions must be checked.

- 1-PPS Signal is present
- Emergency stop interrupt (Interrupt 4) is at a high level.
- Ethernet ribbon cable is plugged into board and the ENC28J60 is plugged in

If they are not then the Ethernet connectivity will fail and diagnostic messaging will only be available through the RS-232 port.

A flow diagram of the power on boot process is shown in Figure 10.3. This diagram shows the early initialisation of the external SRAM, the RS-232 and the Ethernet Controller. This is followed by initialising the structures used for the PID control in the servoloop, loading non-volatile data from EEPROM into SRAM and sending an ARP request from the ENC28J60 to allow it onto the Ethernet network. This is followed by the idle-state termed 'Wait for Interrupts' on the diagram.

The processes available in the idle-state of the microcontroller are interrupt-driven and are also shown in Figure 10.3. The individual interrupt processes are described later in this chapter.

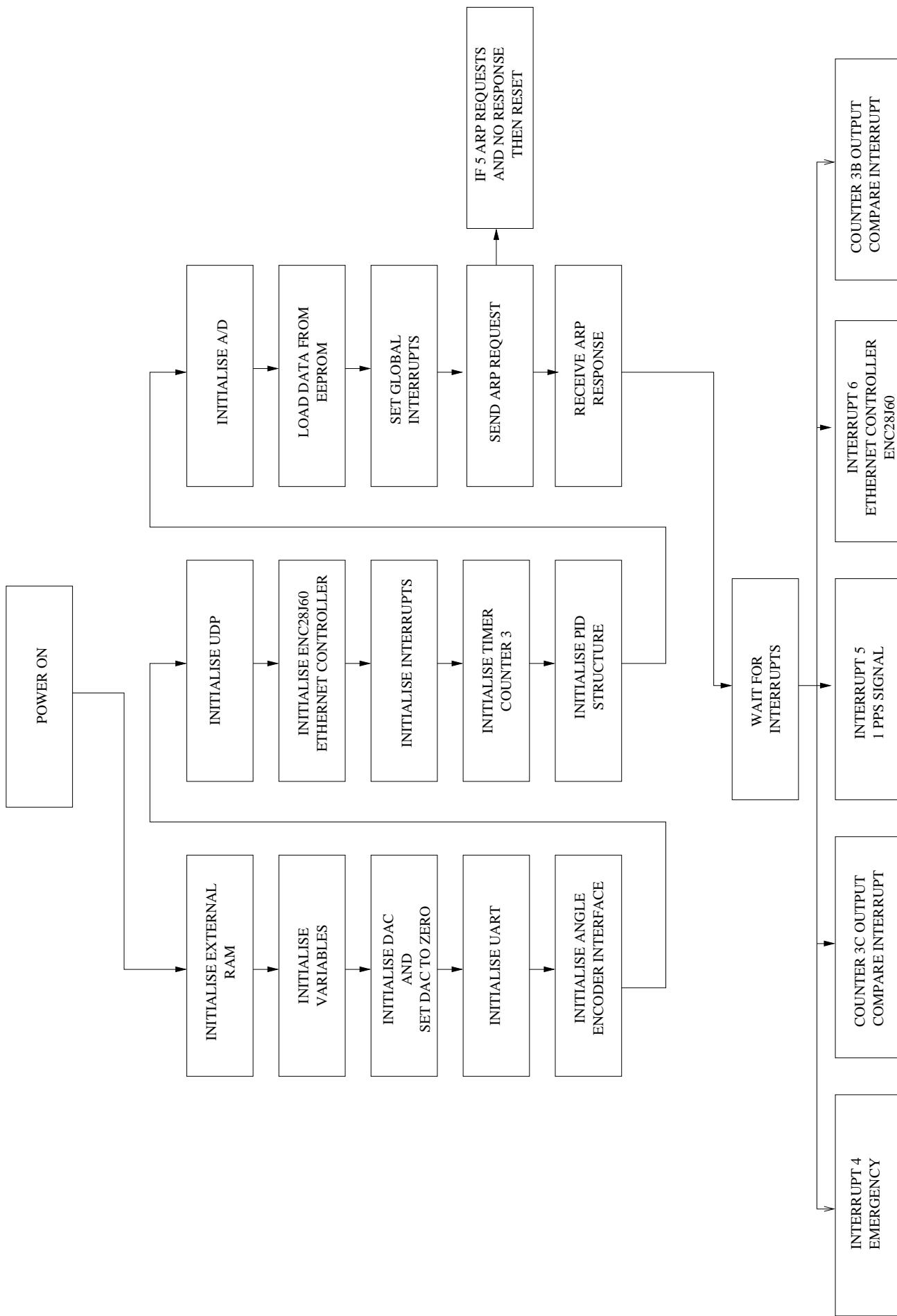


Figure 10.3: Flow diagram of the boot and idle status of the control board together with the interrupt driven processes.

## 10.4 Interrupt Driven Processes

After the power on processes are complete the microcontroller enters an idle state. All subsequent processes are interrupt driven. These processes include emergency overrides, time keeping, handling the servoloop and dealing with Ethernet communications.

### 10.4.1 External Interrupt 4 - Emergency Override

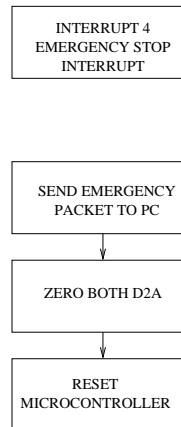


Figure 10.4: Flow Diagram of processes triggered by Emergency Stop Interrupt (INT4)

The Emergency Override interrupt was given the highest priority. The reason for including this is to provide an external override to the control board which will zero the command output voltages and reset the microcontroller. The interrupt acts as a failsafe and could be interfaced to one of the limit switches or to an external panic switch as a means of stopping all drive systems.

### 10.4.2 Timer Counter 3C Compare Interrupt - Servoloop Sampling

Timer Counter 3C provides the timing for the 1/100s servoloop updates and the Interrupt is enabled by a *control packet* from the Observation PC. Counter 3 is /256 of the 16 MHz crystal clock (meaning that 1 second is represented by 62500 counts on Counter 3). The Compare interrupt generates an interrupt when Timer Counter 3 matches a Compare Value. To achieve the 1/100s timing, the Compare Value is incremented by 624 each time the interrupt is serviced (since  $624 \approx 62500/100$ ). The following interrupt will then occur in 624 Counter 3 counts (or 1/100s). I have found by trial and error that an increment of 624 produces timing closer to 1/100s than the expected increment of 625. This may be due to the time required to service the interrupt. The Compare Value is reset to zero on each 1-PPS signal. A flow diagram is shown in Figure 10.5.

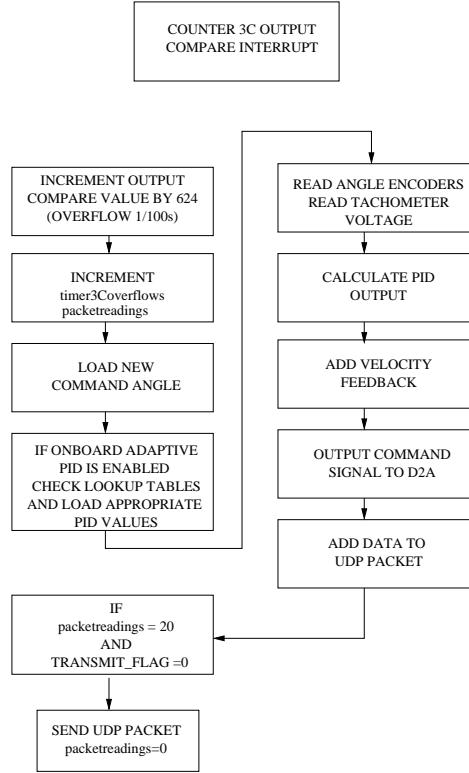


Figure 10.5: Flow diagram of the processes triggered by a Counter 3C Interrupt on Compare Match

#### 10.4.3 External Interrupt 5 - 1-PPS

Time synchronisation between the digital back-end and the controller is necessary and will be done using a combination of the SNTP time synchronisation protocol (detailed in Appendix E.3) and a 1-PPS signal. The time to the nearest second will be handled by SNTP, and the time resolution will be improved by using a 1-PPS signal to synchronise times. This process is outlined below.

##### 1-PPS Signal

This 1-PPS signal will trigger Interrupt 5 on the microcontroller, resetting the Timer Counter 3 Compare Value and incrementing a counter to keep track of the time. The Command Angle buffer is updated for use in the servoloop while servicing Interrupt 5. This interrupt is used as a way of sending SNTP requests every 150 seconds which provides a backup to the 1-PPS time synchronisation.

This interrupt allows a counter (*ping\_counter*) to be incremented each second. This counter is reset on receiving a packet through Ethernet, allowing the microcontroller to monitor the time since the last received packet. If a packet is not received for 2 seconds then the microcontroller enters an emergency mode assuming that there is an error with the Ethernet communications. The DACs are zeroed to prevent unwanted motor movement and

the ENC28J60 is reset to attempt to reopen Ethernet communication. If this state continues for more than 5 seconds the microcontroller is reset and will wait in idle mode. This provides a failsafe against Ethernet communication problems.

The interrupt also resets the *ppsflag* counter. This is used as a failsafe against problems with the 1-PPS signal. The process is handled by the Timer Counter 3B Compare Interrupt explained in Section 10.4.5. A flow diagram of the processes handled by the 1-PPS interrupt is shown in Figure 10.6.

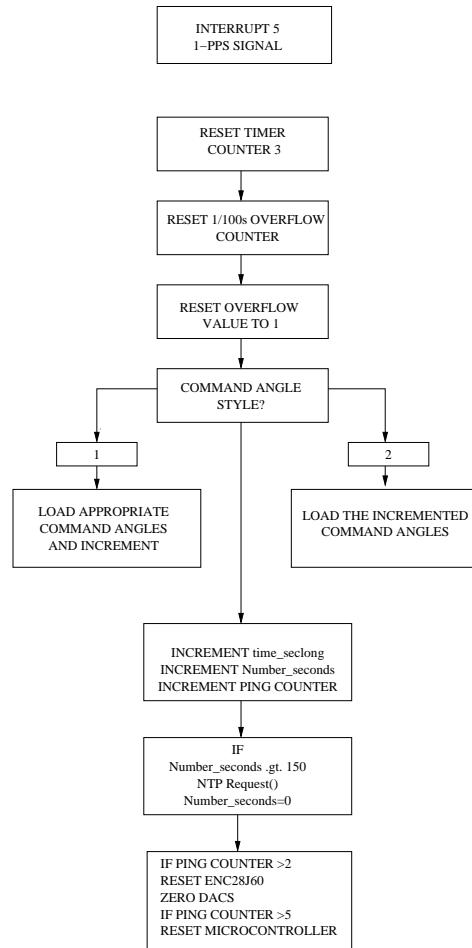


Figure 10.6: Flow diagram of the processes triggered by a 1-PPS Signal (INT5)

## SNTP

On starting the servoloop sampling (i.e enabling Timer Counter 3C Compare Interrupt) and every 150 s thereafter a SNTP request will be sent to the Observation PC. This will ensure that the time is maintained to the correct second while observations are being undertaken.

### 10.4.4 External Interrupt 6 - Ethernet Interrupt

The ENC28J60 features a single interrupt pin  $\overline{INT}$  which changes level if an enabled interrupt condition occurs. This interrupt is connected INT6 on the ATMega128 which will trigger on

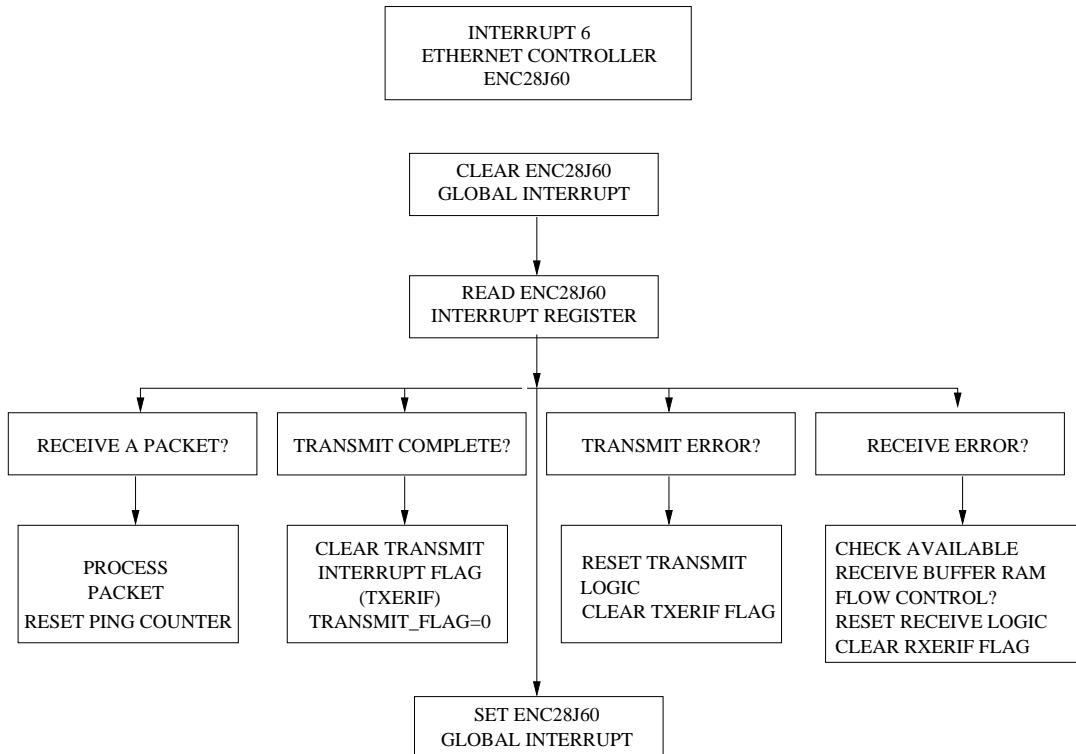


Figure 10.7: Flow diagram of the processes triggered by a ENC28J60 Interrupt Signal (INT6)

a level change. The interrupt conditions enabled on the ENC28J60 are as follows:

- Receive Packet Interrupt (EIE\_PKTIE) This interrupt will set the EIR\_PKTIF interrupt flag when a packet is received by the ENC28J60.
- Transmit Packet Interrupt (EIE\_TXIE) The EIR\_TXIF interrupt flag will be set when the transmission of a packet is completed by the ENC28J60.
- Transmit Error Interrupt (EIE\_TXERIE) The EIR\_TXERIF interrupt flag will be set should an error occur during a packet transmission.
- Receive Error Interrupt (EIE\_RXERIE) The EIR\_RXERIF interrupt flag will be set should an error occur during the reception of a packet.

All of these will trigger the ENC28J60 *INT* pin to be driven low.

#### 10.4.5 Timer Counter 3B Compare Interrupt

This is the second Compare Value Interrupt available on Timer Counter 3. This interrupt is serviced once a second and increments a counter called *ppsflag*. This counter is zeroed while servicing Interrupt 5 (the 1-PPS signal). This allows the microcontroller to 'know' the elapsed time since the last 1-PPS signal was received. If *ppsflag* exceeds 3 counts then 3 seconds have

elapsed and there is a problem with the 1-PPS signal. An error message is sent through the UART and the microcontroller resets. This provides a failsafe against 1-PPS problems.

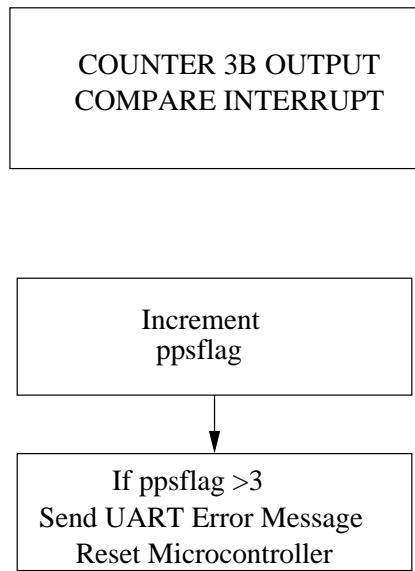


Figure 10.8: Flow diagram of the processes triggered by a Counter 3B Interrupt on Compare Match

## 10.5 Ethernet Configuration

Background information on the Ethernet communication channel is available in Appendix E. This section gives a brief overview of the code used for the ENC28J60 controller, in particular its initialisation and various options that may need to be changed in the future for optimum performance.

### 10.5.1 Initialising

The SPI interface on the ATMega128 is initialised to begin operation. Before starting with the ENC28J60, it is essential to reset the ENC28J60 and to make sure that it is ready to begin operation by polling the ESTAT-CLKRDY bit. This bit is cleared on power on and sets after allowing time for the oscillator and integrated PHY to stabilise. Once this has been set the ENC28J60 is ready to begin operation.

The ENC28J60 has 8 Kb of SRAM which is used for sending and receiving packets. In order to begin receiving packets the ERXST and ERXND pointers need to be set to appropriate memory locations. Memory between these pointers is used for buffering received packets, with the rest being used as a transmit buffer. During this initialisation the ENC28J60 should not receive any packets and should have the receive logic disabled.

Following this the MAC registers need to be set to specify the Ethernet configuration. The ENC28J60 MAC address needs to be set, packet characteristics need to be defined and

either full or half duplex should be selected. This is followed by enabling interrupts that will be used. Hereafter the ENC28J60 should have the receive logic enabled once more.

This process is described more thoroughly in the datasheet (ENC, 2004). The code used to initialise the ENC28J60 is shown in Appendix I.2.

### 10.5.2 Packet Filters

A final decision on what packet filters to use with the ENC28J60 has yet to be made. It is possible to filter out any broadcast packets which would prevent any packets from external sources tying up processor time. The MAC and IP address of the ENC28J60 could be manually set on the PC so that ARP broadcasts were entirely unnecessary. This has the advantage of limiting the chances of overloading the microcontroller. On the other hand this would severely reduce the versatility of the board for purposes where communication to other PC's may be necessary. The code to define this setting is included in `enc28j60.c` in the `enc28j60Init()` function. The relevant code is appropriately commented and can be changed to suit the final decision. Note that if broadcast packets are not accepted, the GIP should be set to SIP in the `enc28j60conf.h` so that the initial ARP requests are sent to the server PC as explained in Section 10.5.3

### 10.5.3 MAC Address, IP Address, Port Settings and NTP and ARP

The microcontroller will request NTP and ARP updates from the IP address specified as GIP in the `enc28j60conf.h` file. The MAC and IP Addresses of the ENC28J60 are specified in this file as ENC28J60\_MAC and MIP. Any changes should be made solely to this file in the appropriate places. The gateway IP address is defined as GIP and the PC (server) IP address is defined as SIP. Should it be necessary to limit communication to be solely between ENC28J60 and the server PC then the GIP address should be changed to the PC IP address i.e GIP=SIP.

### 10.5.4 Available Microcontroller Ethernet Software

Most TCP/IP applications do not require high frequency data transmission. Upon investigating previously written software that might be suitably adapted to our purposes, all roads seemed to lead to a TCP/IP stack known as  $\mu$ IP (Dunkels). The software was written specifically for microcontroller based applications and featured wonderful functionality in implementing both HTTP servers, NTP servers (and clients) as well as a TCP, IP and UDP stack implementation. Unfortunately, after investigation, I was led to believe that the software was optimised to send and receive packets once every second. This would have been entirely unsuitable for driving a telescope at a maximum speed of  $10^{\circ}/s$ . Having looked at the software

with the benefit of hindsight, I think I may have been hasty in coming to this conclusion. Using this software as an option should be investigated further in the future.

## 10.6 PC Code

The PC code is in a very primitive state and has been used exclusively for testing the functionality of the microcontroller. I have tried to keep the layout between the PC code and the microcontroller code as consistent as possible. There are three primary requirements of the PC code:

1. Sending control packets to the microcontroller
2. Updating the command angle buffer on the microcontroller
3. Processing data packets from the microcontroller

These have all been implemented in the code. This is contained in the accompanying CD. Further detail is not relevant to this thesis as I am primarily interested in investigating the board functionality and not in developing a robust PC control system. The PC code was used exclusively for testing purposes and will require a complete rewrite when finally implemented.

## 10.7 Existing bugs

I have experienced difficulties when trying to run the controller over an open network. I think this is due to the large number of other packets which the microcontroller needs to deal with. Filtering out any packets not specifically destined for the microcontroller (e.g broadcast packets) overcomes this problem, however it is not an ideal solution. I have however temporarily used this option together with manually setting the ARP settings on the PC so that periodic ARP requests are not required (`arp -s ip_address mac_address` on linux machines). As a result of this, I stopped experiencing any problems. In future this must be resolved.

### 10.7.1 Sending Packets Errata

Sending packets is handled by interrupts however it should be noted that the ENC28J60 contains an errata (ENC). The problem is that the internal transmit logic may stall causing packet sending to stop. The recommended workaround is to reset the transmit logic before each packet is sent. This is what is done in this case. An additional safety feature is included by enabling the Transmit Error Interrupt (EIE\_TXERIE). Should a transmit error still occur it will trigger a level change on INT6. The transmit error will be identified as the cause and

the transmit logic will be reset again. This has proven to be a successful combination of transmit error prevention/handling procedures.

## 10.8 Future Improvements

For the control board to be truly suitable for robust control over an open network, a greater buffer of command angles should be uploaded. A 5 second buffer should be sufficient. I have found that the system has occasional glitches when exposed to an open network and have narrowed the cause of this down to issues with command angle packets not being timeously processed. If the setup was changed so that there was always a 5 second buffer (instead of the current buffer which is always less than 1 second), I believe this small problem would be solved. I did not have time to do this myself.

While the system as it stands should work satisfactorily on a single antenna, application to antenna arrays could easily be achieved with one central control PC if this minor issue is solved. This would make the controller suitable for use with *MeerKAT* development.

## Chapter 11

# Performance and Reliability Tests

As explained in Section 8.8, the control board is designed to provide velocity command voltages for controlling the DC motors. Moving the drive electronics from the Telkom site to HARTRAO has taken a great deal longer than originally envisioned and as a result I have not been able to implement the system on the C-BASS antenna.

Since acceptance tests of an electronic device are important, I have used the control board to control a slightly different piece of equipment with similar characteristics. This is not an ideal situation, however it does demonstrate the versatility of the system. It also provides a test of the entire system's operation. The tests allow the Ethernet, A/D's, DACs and angle encoders operation to be tested in a working system.

This chapter describes the test equipment, explains the PID tuning process, and shows the results obtained on the test equipment. The chapter concludes by noting the reliability of the control board.

### 11.1 Equipment

The test platform consists of a DC electric motor driven by a pulse width modulated (PWM) signal, similar to the C-BASS motor system described in Subsection 8.3.2. A photograph of the test environment is shown in Figure 11.1, and a diagram of the setup is shown in Figure 11.2. The motor's velocity is measured by an infrared cycle counter which produces an output voltage proportional to velocity. The pulse width modulation is controlled by the input voltage to PWM circuitry also shown in Figure 11.1.

This provides a reasonable test platform for the PID control loop as it is a trivial task to modify the code to use the velocity voltage as a feedback signal through the A/D. It is a velocity control system and not a position control system, however it still provides a test for the overall system operation including data flow.

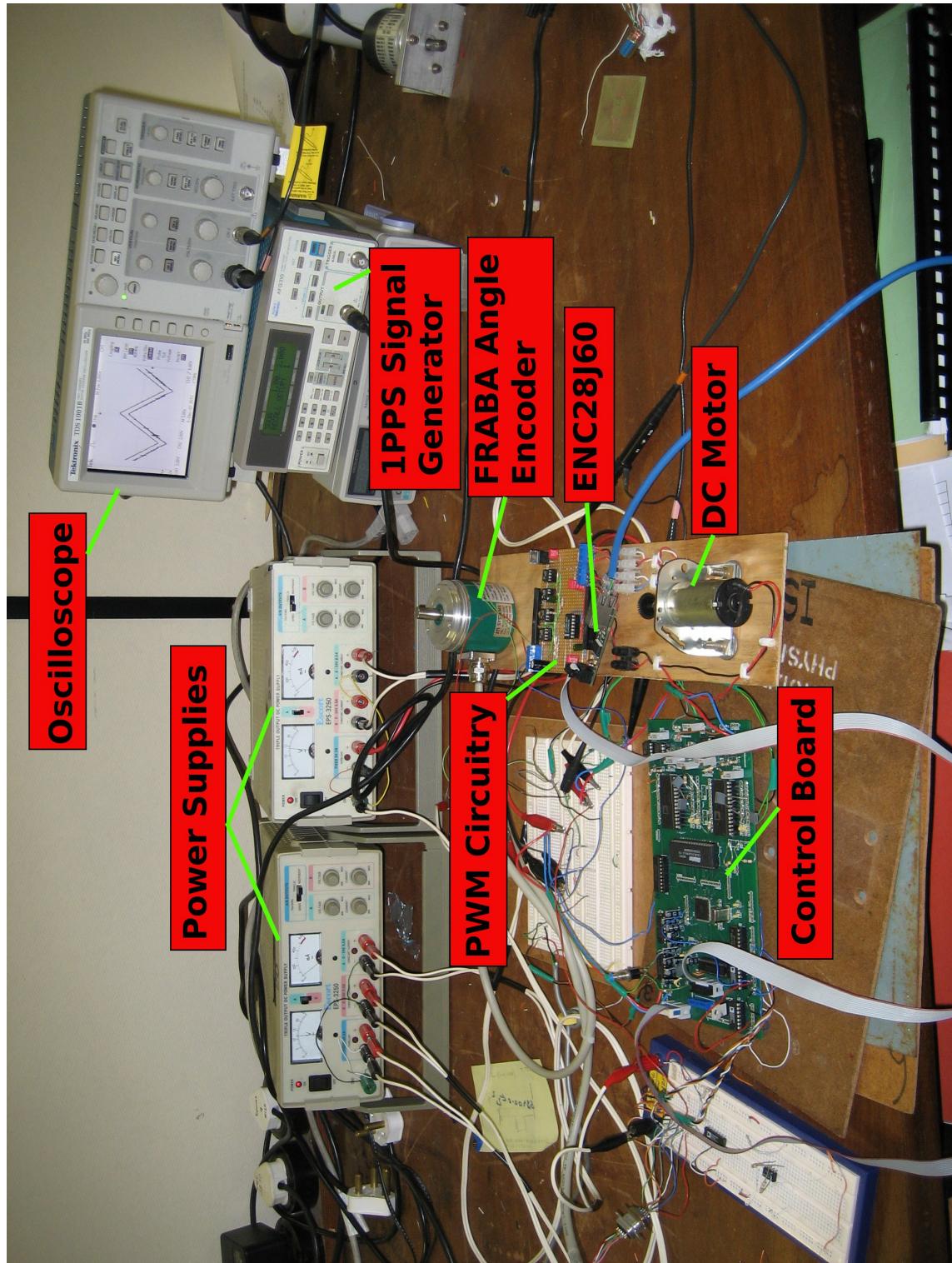


Figure 11.1: Photograph of the equipment used to test the controller. I did not include the server PC since it made it difficult to distinguish between the control hardware.

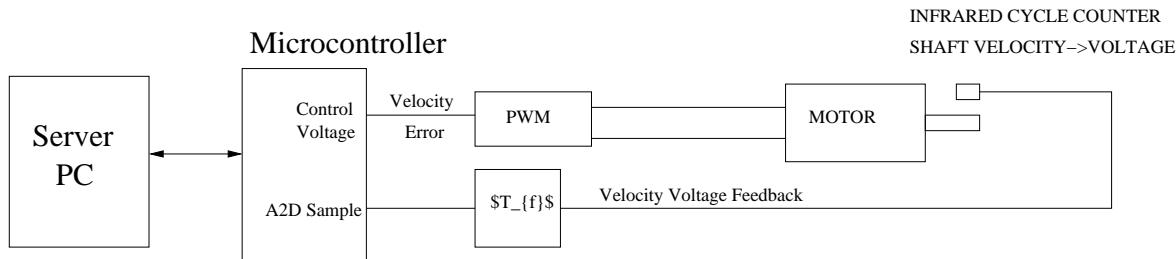


Figure 11.2: Diagram of the equipment used in the control testing

## 11.2 PID tuning

Westcott (2000) explains a simple PID tuning methodology. While there are other, more sophisticated methods, I used this as I only wanted to show DC motor results as a demonstration, and didn't need any more sophisticated options.

The methodology is as follows:

1. Set Proportional gain to a small value (i.e 1) and the Derivative gain to a large value such as 50.
2. Increase the derivative gain until oscillations or excessive noise become obviously apparent.
3. Drop the derivative gain by a factor of 2.
4. Now increase the proportional gain until obvious oscillations are detected. Now decrease the proportional gain by a factor of 2.
5. Set the integral gain to a value between 0.01 and 0.001 and increase it until the onset of oscillations. Then decrease it a little until these oscillations disappear.

Examples of tuning the PID coefficients using this method are given in Figure 11.3 with more detail included in Appendix J.

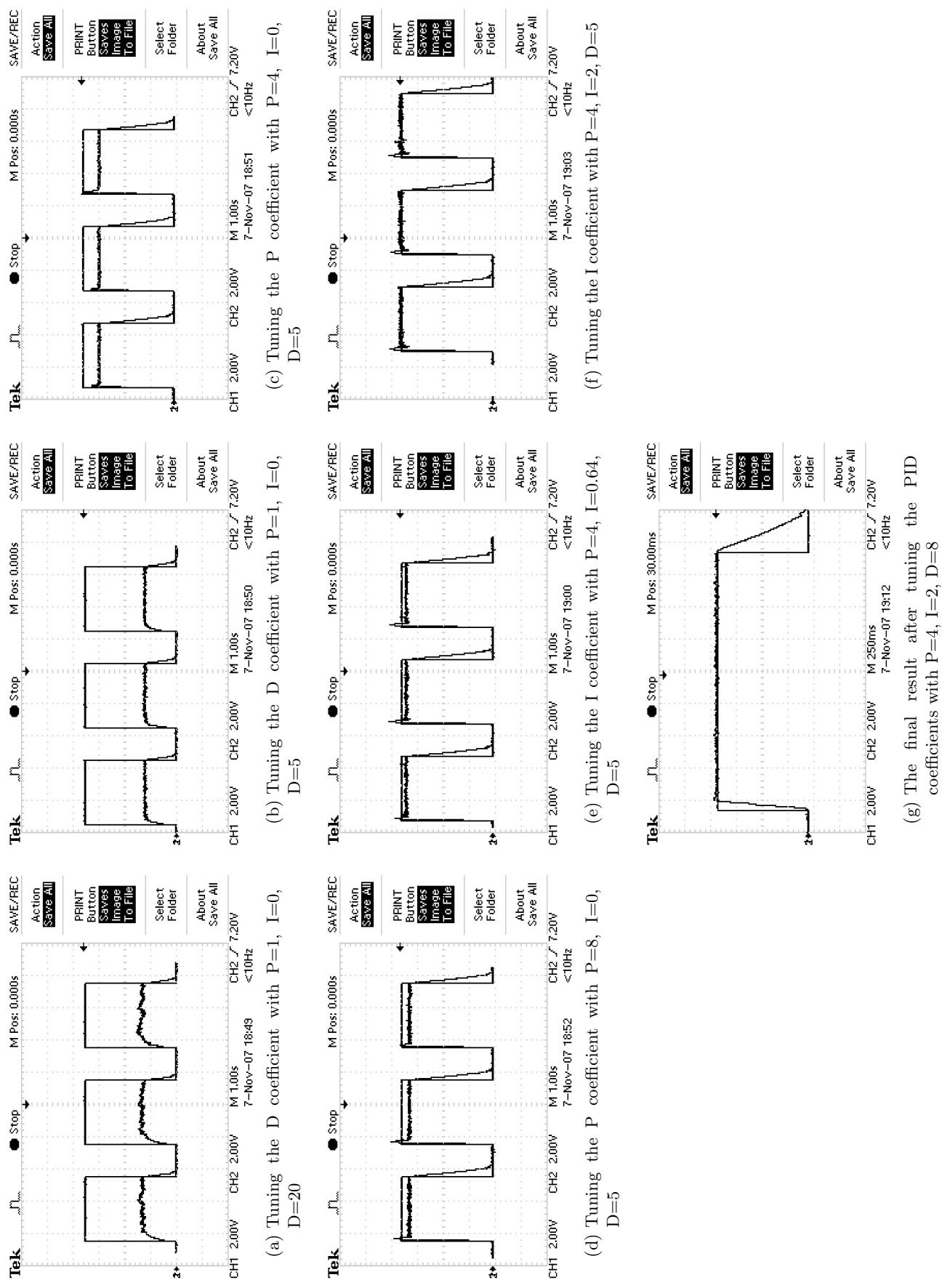


Figure 11.3: These figures show the results of changing and tuning the PID coefficients using the step response of the test equipment. The final tuned result is included in Figure 11.3g. Further detail of the process is included in Appendix J

### 11.3 Adaptive PID Control

One of the great advantages of using a digital controller is that the control system can be changed according to circumstances. Adaptive PID control can be used when the transfer function of a system changes under different load conditions. Different PID constants are needed to achieve optimum performance at these different conditions.

The programming of this system makes this particularly easy to achieve- when a condition is met, a packet containing the new PID constants is simply sent to the microcontroller. It then updates, and the PID control changes according to the new conditions. The downside is that this will increase the number of packets that are handled by the microcontroller and may cause it to fail sporadically. A balance needs to be struck between the 'resolution' of the adaptive PID and the need for a consistent and reliable control system.

PID constants used to achieve good response at high speeds were not effective at low speeds, creating the need for adaptive PID control. This requires the PID "constants" to be dynamic. In this case the PID constants vary with commanded motor speed. Further examples are included in Appendix J.3.

By applying the PID tuning principles elaborated in Section 11.2 and Appendix J together with adaptive PID control we arrive at the two results shown in Figure 11.4. The feedback signal from the test equipment is noisy which is why the response is not perfect, however these figures are promising considering the primitive test equipment used. They show that the PID controller needs fine tuning, but that it is working.

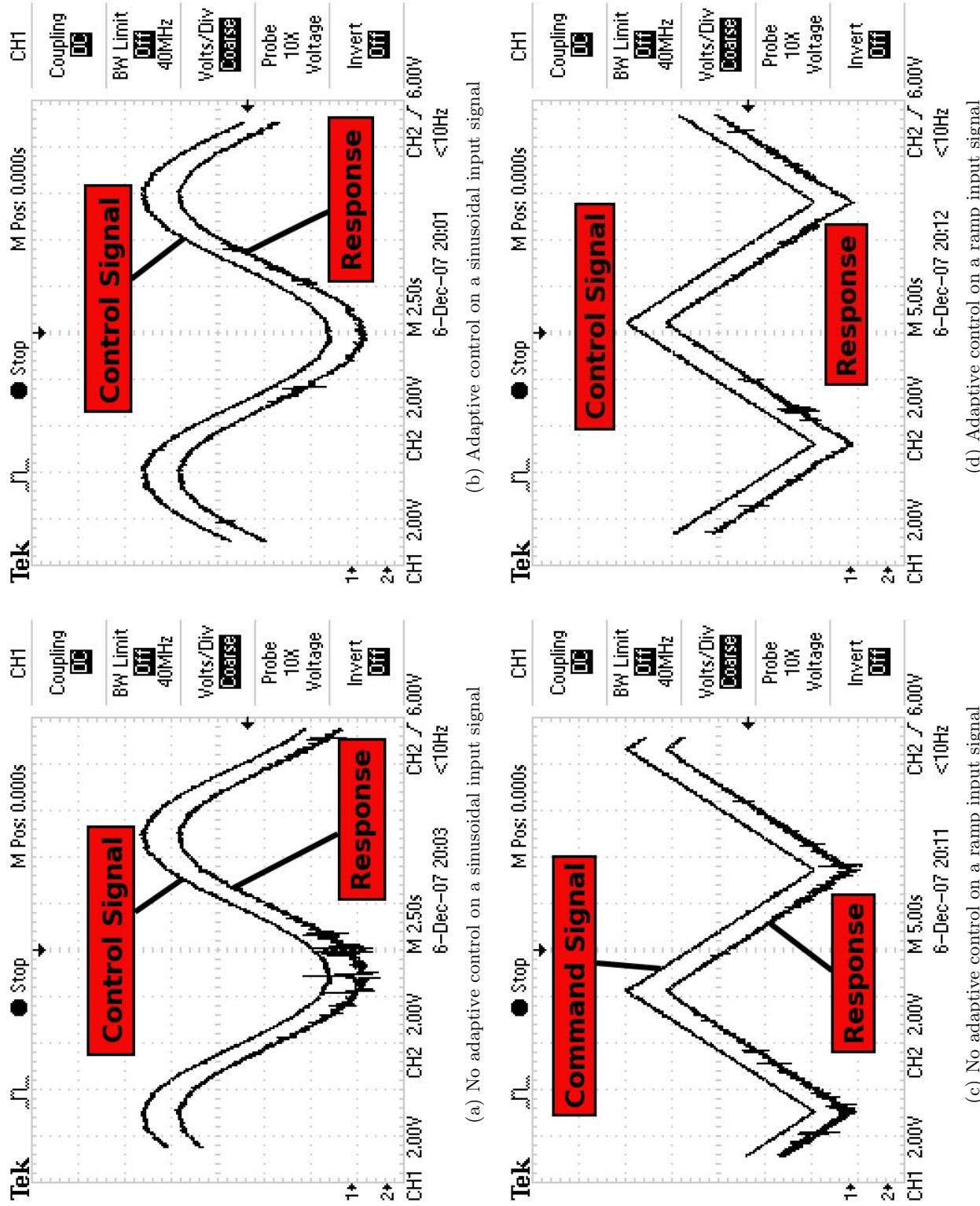


Figure 11.4: Different command signals and the controlled motor responses with and without adaptive control applied. Notice the comparatively poor response (what appears to be noise) of the signal without adaptive control, particularly evident at low command velocities. There is still some noise in the adaptive control system however the tracking is a great deal better particularly at the same low velocities.

## 11.4 Reliability Tests

Reliability of the controller is important. To test the microcontroller and the ENC28J60 I used the ICMP 'ping' facility. This involves sending a UDP datagram from a source to a destination. The destination copies the data and sends it back to the original source. Different packet sizes and sending frequencies can be set.

I elected to test the device using a packet size and frequency similar to that which it would experience when running a motor controller. The ICMP packets sent were 400 bytes in length and I sent them 5 times per second.

The test was carried out over the course of a week. The server PC had the IP and MAC address of the control board manually added (using `arp -s 00:00:de:ad:be:ef 146.231.84.81`) which minimised the number of additional processes carried out on the microcontroller, and the Ethernet cable went directly from the board to the server PC. During the course of the week there were 10 automatic reboots of the device however the device was able to successfully recover from all of these problems.

# Chapter 12

## Conclusion

The aim of this investigation was to build a new control system to allow a communications antenna to be used for astronomical purposes.

I have built a versatile, reliable piece of hardware that can be used in a large number of applications requiring interfacing to a PC through Ethernet. The code that I have written is designed to be flexible and easy to understand. This should allow for relatively painless transfers of the system to other purposes.

The controller has proved promising when used on a small DC motor with tachometer feedback. This has demonstrated the ease of implementation to a system with a simple PID tuning process giving good results. The system has been left running for a week without requiring any external assistance. Further testing is required on the C-BASS system in the future.

It will probably be necessary to use a second board as a means of controlling the clutches, brakes and limit switches of the C-BASS antenna. I believe that one of these boards can be easily adapted to this purpose, which would allow real-time monitoring of these devices over Ethernet.

I believe this system can be used in any number of purposes, and the prototype board is flexible enough for future preliminary investigation to begin on the C-BASS antenna. Possible hardware and code upgrades have been outlined, and with the successful implementation of these options I think that a series of these boards could be used to successfully control and monitor a telescope array from a central control PC over an isolated Ethernet network.

# Bibliography

- CBASS: C-Band All-Sky Survey, <http://www.astro.caltech.edu/cbass/>
- Microchip, ENC28J60 Rev.B4 Silicon Errata-Datasheet
- Mounting Instructions: ROC 415,416,417-Datasheet, Tranreut, Germany
- NEC, RFT Training Manual, 1998
- Microchip, ENC28J60-Datasheet, 2004
- Analog Devices, Low Cost Precision IC Temperature Transducer-AD592\*-Datasheet, Norwood, MA
- Analog Devices, Microprocessor-Compatible 12-Bit D/A Converter-AD667\*-Datasheet, Norwood, MA
- Anderson, E., Bai, Z., Bischof, C., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Ostrouchov, S., Sorensen, D., 1999, Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition
- ANSYS, 1994, ANSYS 5.7, Software
- ATMEL, ATMega128-Datasheet, rev. 2467avr10/06 edition
- Bayley, A. M., Davis, R. J., Haggis, J. S., Karcher, H., 1994, Astronomy and Astrophysics, 283, 1051
- Chiplus, Chiplus, CS18LV10245-Datasheet, 2004
- Chiueh, T., Ma, C.-J., 2002, ApJ, 578, 12
- Condon, J., 2003a, Technical report, Green Bank Telescope, PTCS/PN/10
- Condon, J., 2003b, Technical report, Green Bank Telescope, PTCS/SN/3
- Condon, J., Broderick, J., Seielstad, G., 1989, AJ, 97, 1064
- Condon, J., Broderick, J., Seielstad, G., 1993, AJ, 106, 1095
- Condon, J., Constantines, K., 2003, Technical report, Green Bank Telescope, PTCS/SN/8,
- Condon, J. J., 1997, Publications of the ASP, 109, 166
- Deegan, P., AVR Project Makefile Template, <http://electrons.psychogenic.com>
- Doan, C., Liang, S., Department of Civil Engineering, National University of Singapore, Singapore
- Dunkels, A., The uIP Embedded TCP/IP Stack, <http://www.sics.se/~adam>

- Emerson, D., 1990, Technical report, NRAO
- Gawronski, W., 2002, Technical report, Jet Propulsion Laboratory, CALTECH
- Gawronski, W., 2004
- Gawronski, W., Souccar, K., 2003
- Gaylard, M., 2005a, Hartebeeshoek Radio Astronomy
- Gaylard, M., 2005b, Uncertainty Estimates in Fitting a Gaussian Profile
- Harp, G., The ATA Servo Loop, 2006
- HARTRAO, HARTRAO Radio Telescope Website, <http://www.hartrao.ac.za/>
- HARTRAO, HARTRAO Schedule file,  
<http://www.hartrao.ac.za/nccsdoc/inputfiles/inputs-3.html>
- Himwich, W., 1993, Technical report, NASA/Goddard Space Flight Center Space Geodesy Project
- Holdaway, M., 1990, Technical report, NRAO
- Holdaway, M., 1991, Technical report, NRAO
- Holdaway, M., 1997, Technical report, NRAO, 949 N.Cherry Ave, Tucson, AZ, 85721-0655, Memo178
- Holmeide, O., Skeie, T., 2001, IEEE, Emerging Technologies and Factory Automation. Proc.
- Hunter, T. R., Wilson, R. W., Kimberk, R., Leiker, P. S., 2001, in *Bulletin of the American Astronomical Society*, Vol. 33 of *Bulletin of the American Astronomical Society*, p. 791
- Joliffe, I., 2002, Springer
- Kaper, H. G., Smits, D. W., Schwarz, U., Takakubo, K., van Woerden, H., 1966, Bulletin Astronomical Institute of the Netherlands, 18, 465
- Keating, B., Timbie, P., Polnarev, A., Steinberger, J., 1998, ApJ, 495, 580
- King, O., 2006, *Master's thesis*, Rhodes University
- Kloth, R., SNTP date and time (UTC) from an NTP time server,  
<http://www.kloth.net/software/sntp.php>
- Kwon, Y., Bang, H., 2000, CRC Mechanical Engineering Series, CRC
- Lamb, J., Forster, J., 1993, Technical report, Owens Valley Radio Observatory, California Institute of Technology, MMA Memo 100
- Lamb, J., Woody, D., 1998, Technical report, Owens Valley Radio Observatory, California Institute of Technology, MMA Memo 234
- Lamb, J., Woody, D., 2002, Proc. URSI General Assembly
- MATLAB R14, Chapt. Neural Networks Toolbox, Mathworks
- Mills, D., University of Delaware, RFC 2030 - Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI, 1996

- Nise, N., 2000, John Wiley & Sons
- Nothnagel, A., Wirtz, C., Sauerbier, M., Sobanski, M., 1997, Proc. 13th Working Meeting on European VLBI for Geodesy and Astrometry
- Ogata, K., 1970, Electrical Engineering Series, Prentice-Hall
- Ott, H., 2001, Printed Circuit Design Magazine, p. 8
- POSITAL FRABA, Absolute Rotary Encoder(SSI)-Datasheet, info@posital.de
- Press, W., Teukolsky, S., Vetterling, W., Flannery, B., 2001, Cambridge University Press
- Reich, W., 2006, ArXiv Astrophysics e-prints
- Rhodes University, 2006, Radio Telescopes and Instrumentation
- Rodriguez, E., TCP vs. UDP, <http://www.skullbox.net/tcpudp.php>
- Schlens, J., 2005, A Tutorial on Principal Component Analysis
- Schwab, C., 2004, The Industrial Ethernet Book
- Sinha, N., 1986, Holt,Rinehart and Winston
- Sullivan, A., 2001, *Master's thesis*, Rhodes University
- Tanenbaum, A., 1996, Prentice-Hall International, Inc
- The Mathworks, 2005, MATLAB 7.0, Software
- Ukita, N., 1999, Publ. Natl. Astron. Obs. Japan, 5, 139
- Ukita, N., Ezawa, H., Mimura, H., Sugamuma, A., Kitazawa, K., Masuda, T., Kawaguchi, N., Sugiyama, R., Miyawaki, K., 2001, Publ. of the Natl Astron. Obs. Japan
- Von Hoemer, S., 1975, A+A
- Wallace, P., 2004, SLALIB Positional Astronomy Software, <http://www.starlink.ac.uk/WCS>, Introduction to celestial coordinate systems,  
<http://www.atnf.csiro.au/people/mcalabre/WCS/Intro/WCS11.html>
- Westcott, T., 2000, PID without a PHD,  
<http://www.embedded.com/2000/0010/0010feat3.htm>
- Wikipedia, 2007a, Ethernet — Wikipedia, The Free Encyclopedia, [Online; accessed 27-June-2007]
- Wikipedia, 2007b, TCP/IP model — Wikipedia, The Free Encyclopedia, [Online; accessed 27-June-2007]
- Williams, A., 1981, McGraw-hill book Company
- Yaskawa, SERVOPACK-DC Servomotor Controller for Speed Control Type CPCR-MR01C to -MR99C-Datasheet, Document:TSE-C717-12G
- Zaldarriaga, M., 2004, in W. L. Freedman (ed.), *Measuring and Modeling the Universe*, p. 309

# **Part III**

# **Appendix**

## Appendix A

# Elliptical Gaussian Uncertainties

The derivation below is taken from Condon (1997) with less specific reference to Kaper et al. (1966).

The equation for an elliptical Gaussian can be represented as

$$G(x, y) = A \exp \left[ -\frac{(x - x_0)^2}{2\sigma_x^2} - \frac{\beta(x - x_0)(y - y_0)}{\sigma_x \sigma_y} - \frac{(y - y_0)^2}{2\sigma_y^2} \right] \quad (\text{A.1})$$

$A$  gives the peak amplitude,  $(x_0, y_0)$  the central position of the Gaussian,  $\sigma_x$  and  $\sigma_y$  are the RMS lengths of the major and minor axes and  $\beta$  is a measure of the position-angle difference between the principal axes of the ellipse and the coordinate axes (Condon, 1997).

A description of estimating uncertainties for fitting the elliptical Gaussian to images is described by Condon (1997). This method can be used to estimate uncertainties for fitting to this data.

Suppose an image is constructed of  $m$  independent amplitude measurements  $a_k (k = 1, \dots, m)$  each having the same Gaussian error distribution with the same RMS  $\mu$ . If this is the case then the best-fit is one which minimises the sum of the residual variances

$$\chi^2 = \sum_{k=1}^m \frac{[a_k - G(x_k, y_k)]^2}{\mu^2} \quad (\text{A.2})$$

The function  $G(x, y)$  has 6 (potentially) free parameters ( $p_i$ ) where  $p_1 = A$ ,  $p_2 = x_0$ ,  $p_3 = y_0$ ,  $p_4 = \sigma_x$ ,  $p_5 = \sigma_y$ ,  $p_6 = \beta$ . Each of these has one normal equation constraining the fitting

$$\frac{\partial \chi^2}{\partial p_i} = \frac{2}{\mu^2} \sum_{k=1}^m [a_k - G(x_k, y_k)] \frac{\partial G(x_k, y_k)}{\partial p_i} = 0 \quad (\text{A.3})$$

Furthermore, if we assume that the residuals are small then we can linearise the normal equations using a Taylor-series approximation

$$[a_k - G(x_k, y_k)] \approx \sum_{j=1}^n \frac{\partial G(x_k, y_k)}{\partial p_j} dp_j \quad (\text{A.4})$$

Using Equation A.4, Equation A.3 becomes

$$\frac{2}{\mu^2} \sum_{k=1}^m \sum_{j=1}^n \frac{\partial G(x_k, y_k)}{\partial p_j} \frac{\partial G(x_k, y_k)}{\partial p_i} dp_j = 0; \quad i = 1, \dots, n \quad (\text{A.5})$$

To simplify the above we define the above is the  $n \times n$  symmetric matrix  $\mathbf{D}$  with elements

$$\mathbf{D}_{ij} = \sum_{k=1}^m \frac{\partial G(x_k, y_k)}{\partial p_j} \frac{\partial G(x_k, y_k)}{\partial p_i} \quad (\text{A.6})$$

This represents the coefficients of the linearised normal equations. If the data samples cover the main peak of the Gaussian and are uniformly separated by separation  $h$  in each coordinate, the sum over  $k$  can be approximated by integration.

$$\sum_{k=1}^m \frac{\partial G(x_k, y_k)}{\partial p_i} \frac{\partial G(x_k, y_k)}{\partial p_j} \approx \frac{1}{h^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{\partial G(x, y)}{\partial p_i} \frac{\partial G(x, y)}{\partial p_j} dx dy \quad (\text{A.7})$$

Evaluating these double integrals is most easily accomplished in the limit as  $\beta \rightarrow 0$  which requires that arbitrary coordinate axes be chosen so that they are nearly parallel to the principal axes of the ellipse (since  $\beta$  is a measure of the angle between the principal axes and the coordinate axes). If we let  $p_1 = A$ ,  $p_2 = x_0$ ,  $p_3 = y_0$ ,  $p_4 = \sigma_x$ ,  $p_5 = \sigma_y$ ,  $p_6 = \beta$  then evaluating the double integral results in the coefficient matrix  $\mathbf{D}$  given below

$$\mathbf{D} \approx \frac{\pi A}{2h^2} \begin{pmatrix} \frac{2\sigma_x\sigma_y}{A} & 0 & 0 & \sigma_y & \sigma_x & 0 \\ 0 & \frac{A\sigma_y}{\sigma_x} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{A\sigma_x}{\sigma_y} & 0 & 0 & 0 \\ \sigma_y & 0 & 0 & \frac{3A\sigma_y}{2\sigma_x} & \frac{A}{2} & 0 \\ \sigma_x & 0 & 0 & \frac{A}{2} & \frac{3A\sigma_x}{2\sigma_y} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{A\sigma_x\sigma_y}{2} \end{pmatrix}.$$

## A.1 Error Matrix and Error Propagation

The inverse matrix  $\mathbf{D}^{-1}$  is called the error matrix because the variance  $\mu^2(F)$  of a differentiable function  $F(p_1, \dots, p_n)$  of the fitted parameters is

$$\mu^2(F) = \mu^2 \sum_{i=1}^n \sum_{j=1}^n D_{ij}^{-1} \frac{\partial F}{\partial p_i} \frac{\partial F}{\partial p_j} \quad (\text{A.8})$$

This error matrix for the full 6 parameters is given by

$$D^{-1} \approx \frac{2h^2}{\pi A} \begin{pmatrix} \frac{A}{2\sigma_x\sigma_y} & 0 & 0 & \frac{-1}{2\sigma_y} & \frac{-1}{2\sigma_x} & 0 \\ 0 & \frac{\sigma_x}{A\sigma_y} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\sigma_y}{A\sigma_x} & 0 & 0 & 0 \\ \frac{-1}{2\sigma_y} & 0 & 0 & \frac{\sigma_x}{A\sigma_y} & 0 & 0 \\ \frac{-1}{2\sigma_x} & 0 & 0 & 0 & \frac{\sigma_y}{A\sigma_x} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{2}{A\sigma_x\sigma_y} \end{pmatrix}.$$

The diagonal elements of  $D^{-1}$  represent the variance such that

$$\mu^2(A) = \mu^2 D_{11}^{-1} \approx \frac{2A^2}{\pi\sigma_x\sigma_y} \left( \frac{h^2\mu^2}{A^2} \right) \quad (\text{A.9a})$$

$$\mu^2(x_0) = \mu^2 D_{22}^{-1} \approx \frac{2\sigma_x}{\pi\sigma_y} \left( \frac{h^2\mu^2}{A^2} \right) \quad (\text{A.9b})$$

$$\mu^2(y_0) = \mu^2 D_{33}^{-1} \approx \frac{2\sigma_y}{\pi\sigma_x} \left( \frac{h^2\mu^2}{A^2} \right) \quad (\text{A.9c})$$

$$\mu^2(\sigma_x) = \mu^2 D_{44}^{-1} \approx \frac{2\sigma_x}{\pi\sigma_y} \left( \frac{h^2\mu^2}{A^2} \right) \quad (\text{A.9d})$$

$$\mu^2(\sigma_y) = \mu^2 D_{55}^{-1} \approx \frac{2\sigma_y}{\pi\sigma_x} \left( \frac{h^2\mu^2}{A^2} \right) \quad (\text{A.9e})$$

$$\mu^2(\beta) = \mu^2 D_{66}^{-1} \approx \frac{4h^2}{\pi\sigma_x\sigma_y} \left( \frac{h^2\mu^2}{A^2} \right) \quad (\text{A.9f})$$

The equations can be simplified if we introduce a parameter describing the signal to noise ratio of the fit. Each independent data point used in constraining the Gaussian fit can be given a statistical weight **proportional to the square of its signal-to-noise ratio**,  $p_k = G(x_k, y_k)/\mu$ . Quadratically summing over all the data points in the fit produces the number  $\rho^2$ , which is a measure of the overall signal-to-noise ratio of the Gaussian fit.

$$\begin{aligned} \rho^2 &= \sum_{k=1}^m \frac{G(x_k, y_k)^2}{\mu} \\ &\approx \frac{1}{h^2 \mu^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} G^2(x, y) dx dy \\ &= \frac{\pi \sigma_x \sigma_y}{h^2} \left( \frac{A}{\mu} \right)^2 \end{aligned} \quad (\text{A.10})$$

We see the  $\rho^2$  is the product of two numbers. The first is an effective number  $N = \pi \sigma_x \sigma_y / h^2$  of independent samples in the fitted Gaussian and the second is the square of the signal-to-noise ratio of the central sample  $\left( \frac{A}{\mu} \right)^2$ . Substituting this into equations A.9 leads to

$$\frac{2}{\rho^2} \approx \frac{\mu^2(A)}{A^2} \quad (\text{A.11a})$$

$$= \frac{\mu^2(x_0)}{\sigma_x^2} \quad (\text{A.11b})$$

$$= \frac{\mu^2(y_0)}{\sigma_y^2} \quad (\text{A.11c})$$

$$= \frac{\mu^2(\sigma_x)}{\sigma_x^2} \quad (\text{A.11d})$$

$$= \frac{\mu^2(\sigma_y)}{\sigma_y^2} \quad (\text{A.11e})$$

$$= \frac{\mu^2(\beta)}{2} \quad (\text{A.11f})$$

For astronomers these quantities are usually expressed in terms of the full-width between half-maximum points (FWHM) where the major diameter is  $\Theta_M \sqrt{8 \ln 2} \sigma_x$  and the minor diameter is  $\Theta_m \sqrt{8 \ln 2} \sigma_y$ . The major position angle  $\phi$  is used instead of  $\beta$  which is defined as

$$\frac{(x - x_0)^2}{2\sigma_x^2} + \frac{\beta(x - x_0)(y - y_0)}{\sigma_x \sigma_y} + \frac{(y - y_0)^2}{2\sigma_y^2} = 1 \quad (\text{A.12})$$

For small values of  $\beta$

$$\lim_{\beta \rightarrow 0} \phi = \beta \left( \frac{\sigma_x \sigma_y}{\sigma_y^2 - \sigma_x^2} \right)$$

Using Equation A.8 we can get the error in  $\phi$  as

$$\begin{aligned} \mu^2(\phi) &= \mu^2 \left[ D_{44}^{-1} \left( \frac{\partial \phi}{\partial \sigma_x} \right)^2 + D_{55}^{-1} \left( \frac{\partial \phi}{\partial \sigma_y} \right)^2 + D_{66}^{-1} \left( \frac{\partial \phi}{\partial \sigma_\beta} \right)^2 \right] \\ &\approx \frac{4 \sigma_x \sigma_y}{\pi (\sigma_x^2 - \sigma_y^2)^2} \left( \frac{h^2 \mu^2}{A^2} \right) \end{aligned} \quad (\text{A.13})$$

Equation A.13 is only valid for  $\mu^2(\phi) \ll 1$

Using these quantities we arrive at

$$\rho^2 = \frac{\pi}{8\ln 2} \frac{\Theta_M \Theta_m A^2}{h^2 \mu^2} \quad (\text{A.14})$$

The uncertainties are given by

$$\frac{2}{\rho^2} \approx \frac{\mu^2(A)}{A^2} \quad (\text{A.15a})$$

$$= 8\ln 2 \frac{\mu^2(x_0)}{\Theta_M^2} \quad (\text{A.15b})$$

$$= 8\ln 2 \frac{\mu^2(y_0)}{\Theta_m^2} \quad (\text{A.15c})$$

$$= \frac{\mu^2(\Theta_M)}{\Theta_m^2} \quad (\text{A.15d})$$

$$= \frac{\mu^2(\Theta_m)}{\Theta_m^2} \quad (\text{A.15e})$$

$$= \frac{\mu^2(\phi)}{2} \left( \frac{\Theta_M^2 - \Theta_m^2}{\Theta_M \Theta_m} \right)^2 \quad (\text{A.15f})$$

In using these equations for my estimates of uncertainty, I calculated the noise ( $\mu$ ) by using the RMS value of the first 40 data points, which I assumed were sampled when the beam was not on source. I used a sample spacing interval ( $h$ ) which I obtained from the scans.

## A.2 Estimates of Uncertainties in this data

Pointing offsets were obtained from observations by fitting elliptical Gaussians to point source observations. The relationships described above were used to estimate the uncertainties of the pointing offsets derived using this method. These are shown in Figure A.1. From the figure it is clear that the magnitude of the error is significantly less than the target pointing offset of 4 mdeg and, for this reason, I have treated these uncertainties as negligible.

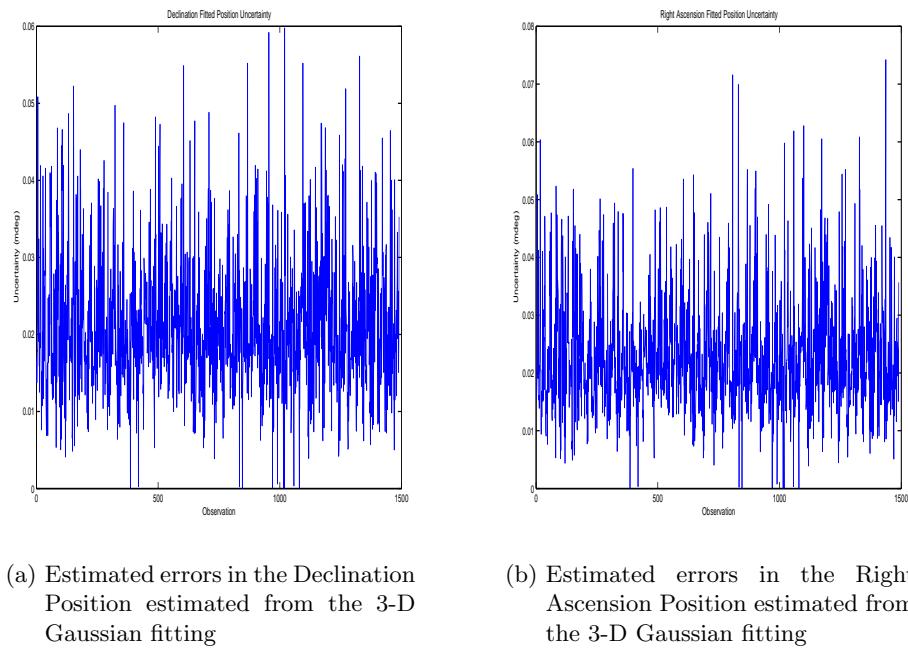


Figure A.1: Typical uncertainties in position estimated using a 3-D Gaussian. The values are all much less than 1 mdeg and thus are not significant in the context of this project.

## Appendix B

# VLBI Parameter listing

### B.1 Description of the model and parameters

The VLBI pointing model is parameter based and has the form shown below. It is described in detail by Himwich (1993).

$$\begin{aligned}\Delta X = & P_1 - P_2 \cos \phi \sin X \sec Y + P_3 \tan Y - P_4 \sec Y + P_5 \sin X \tan Y \\ & - P_6 \cos X \tan Y + P_{12} X + P_{13} \cos X + P_{14} \sin X + P_{15} \cos 2X \\ & + P_{16} \sin 2X\end{aligned}\quad (\text{B.1})$$

$$\begin{aligned}\Delta Y = & P_5 \cos X + P_6 \sin X + P_7 - P_8(\cos \phi \cos X \sin Y - \sin \phi \cos Y) \\ & + P_9 Y + P_{10} \cos Y + P_{11} \sin Y\end{aligned}\quad (\text{B.2})$$

For the HARTRAO Antenna X is the Hour Angle, Y is the Declination and  $\phi$  is the geodesic latitude ( $25.8873^\circ$  S) at HARTRAO, however the model is suitable for use with other local coordinate systems.

$P_1$  is the X-angle offset, the difference of the X-angle encoder bias (positive if encoder reading is too high) minus 'tilt around', which is the tilt of the antenna around the  $Y=90^\circ$ . (positive if apparent  $X,Y=(0^\circ,0^\circ)$  is closer to true  $X,Y=(+90^\circ,0^\circ)$ ).

$P_2$  is the X angle sag, the effect of gravity on the RF axis of the dish projected on the X direction (positive if the RF axis is lower)

$P_3$  is the perpendicular axis skew, the apparent  $Y=+90^\circ$  to true  $Y=0^\circ$  plane lack of orthogonality in the plane perpendicular to the current X angle meridian (positive if apparent  $Y=+90^\circ$  is closer to true  $X,Y=((\text{current } X)-90^\circ,0^\circ)$ )

$P_4$  is the box offset, RF-axis to radial direction misalignment along the X direction (positive if RF-axis is toward the increasing X angle direction)

$P_5$  is the "tilt out", tilt of the apparent  $Y=+90^\circ$  toward the true  $X,Y=(0,0)$  position (positive if apparent  $Y=+90^\circ$  is closer to true  $X,Y=(0,0)$ )

$P_6$  is the "tilt over", tilt of the  $Y=+90^\circ$  toward the  $X,Y=(+90^\circ,0^\circ)$  position (positive if apparent  $Y=+90^\circ$  is closer to true  $X,Y=(+90^\circ,0^\circ)$ )

$P_7$  is the Y angle offset, difference of the Y angle encoder bias (positive if encoder reads too high) minus the sum of the skew of  $Y=+90^\circ$  along the current X meridian angle (positive if apparent  $Y=+90^\circ$  is farther from true  $X,Y=(\text{current } X, 0^\circ)$ ) plus the RF axis to radial direction misalignment along the Y direction (positive if the RF axis is toward the increasing Y angle direction)

$P_8$  is the Y angle sag, the effect of gravity on the RF axis of the dish projected on the Y direction (positive if the RF axis is lower)

$P_9$  is an *ad hoc* Y-angle excess scale factor (greater than 0 if the encoder read-out changes faster than the actual antenna position)

$P_{10}$  is an *ad hoc*  $Y \cos Y$  coefficient

$P_{11}$  is an *ad hoc*  $Y \sin Y$  coefficient

$P_{12}$  is an *ad hoc* X-angle excess scale factor (greater than 0 if the encoder read-out changes faster than the actual antenna position)

$P_{13}$  is an *ad hoc*  $X \cos X$  coefficient

$P_{14}$  is an *ad hoc*  $X \sin X$  coefficient

$P_{15}$  is an *ad hoc*  $X \cos 2X$  coefficient

$P_{16}$  is an *ad hoc*  $X \cos 2X$  coefficient

## B.2 Discussion of Model

The model is an attempt to explain and quantify two primary effects found in telescopes.

- Misalignments in the coordinate system
- Gravitational deformation and other *ad hoc* terms

The parameters  $P_1, P_3, P_4, P_5, P_6$  and  $P_7$  are representative of misalignments of the RF feed with all the other parameters representing gravitational deformations or *ad hoc* terms.

Table B.1: VLBI Pointing Model Parameters

Description	Original	Refitted	% Change
$P_1$	-0.0074075498	-0.01049454780367	29.4153
$P_2$	-0.0063556978	-0.00686492941186	7.4179
$P_3$	-0.0047638724	-0.00517999928007	8.0333
$P_4$	-0.0024362272	-0.00279710700103	12.9019
$P_5$	0.0544945337	0.05452390127463	0.0539
$P_6$	-0.0207822602	-0.02080699817014	0.1189
$P_7$	-0.0595969707	-0.05996367310289	0.6115
$P_8$	-0.0234767608	-0.02450930739531	4.2129
$P_9$	0	-0.00110319602355	100.0000
$P_{10}$	0	-0.00023846410765	100.0000
$P_{11}$	0.0173795968	0.01756346900581	1.0469
$P_{12}$	0	0.04962180835857	100.0000
$P_{13}$	0	0.00339941490416	100.0000
$P_{14}$	-0.0355144888	-0.10342461981940	65.6615
$P_{15}$	0	-0.00085846903721	100.0000
$P_{16}$	0	0.00893371292985	100.0000

## Appendix C

# DC Motor Transfer Function

DC servomotors comprise part of the existing drive system used on the proposed C-BASS antenna. An analytical description of the transfer function is derived and can be used in future simulations of the system and analysis of the antenna characteristics.

A typical DC servomotor is shown in Figure C.1.

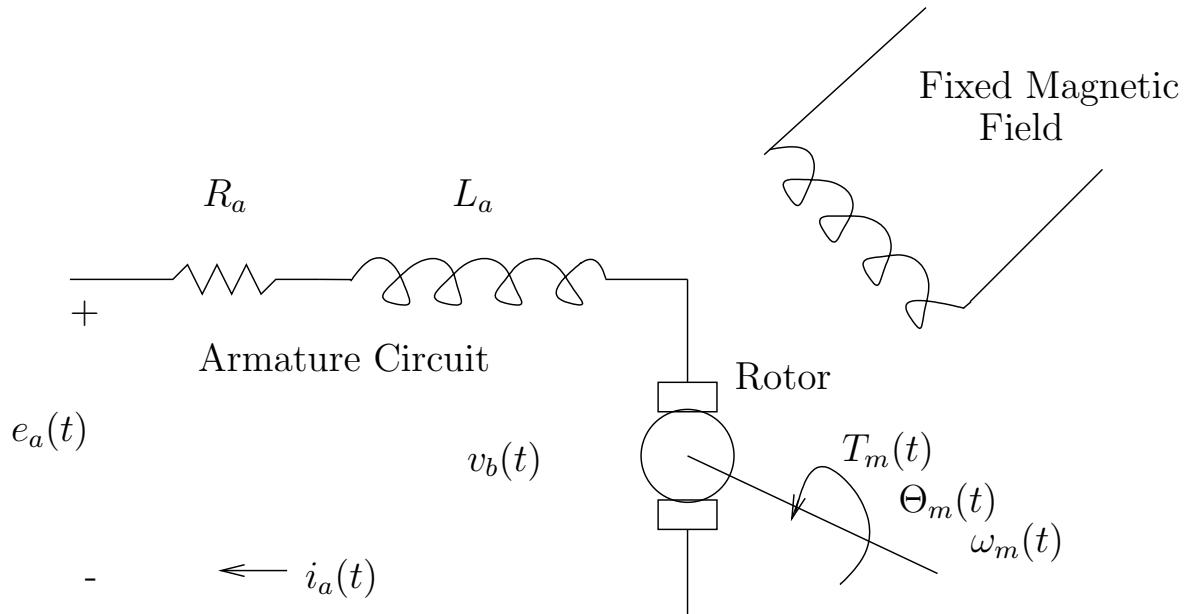


Figure C.1: Schematic of a DC motor

A DC motor works since a current passing at right angles through a magnetic field will feel a force. This is given by

$$F = Bl i_a(t) \quad (\text{C.1})$$

where  $B$  is the magnetic field strength,  $l$  is the length of the conductor passing through the magnetic field, and  $i_a(t)$  is the current passing through the conductor. In a motor the conductor is a rotating circuit known as an *armature*, and the shaft connected to this armature is known as the *rotor*.

Furthermore, a conductor moving at right angles to a magnetic field generates a voltage known as the *back electromotive force* or the *back e.m.f* at its terminals. This is given by

$$v_b(t) = Blv \quad (\text{C.2})$$

where  $v_b(t)$  is the back e.m.f and  $v$  is the velocity of the conductor normal to the magnetic field. In the case of an armature, this voltage is proportional to its speed and we can write

$$\begin{aligned} v_b(t) &= K_b \frac{d\Theta_m(t)}{dt} \\ &= K_b \omega_m(t) \end{aligned} \quad (\text{C.3})$$

$K_b$  is a constant of proportionality known as the *back emf constant*. Taking the Laplace transform this becomes

$$V_b(s) = K_b \omega_m(s) \quad (\text{C.4})$$

By forming a loop equation around the armature circuit shown in Figure C.1 (in the frequency domain) we obtain

$$R_a I_a(s) + L_a s I_a(s) + V_b(s) = E_a(s) \quad (\text{C.5})$$

The torque,  $T_m$ , produced by the motor is proportional to the current giving

$$T_m(s) = K_t I_a(s) \quad (\text{C.6})$$

where  $K_t$  is the *motor torque constant*. This leads to

$$I_a(s) = \frac{1}{K_t} T_m(s) \quad (\text{C.7})$$

Substituting Equation C.4 and C.7 into Equation C.5 we arrive at

$$\begin{aligned} \frac{R_a}{K_t} T_m(s) + \frac{L_a s}{K_t} T_m(s) + K_b \omega_m(s) &= E_a(s) \\ \frac{R_a + L_a s}{K_t} T_m(s) + K_b \omega_m(s) &= E_a(s) \end{aligned} \quad (\text{C.8})$$

In a rotating mechanical system

$$T_m(s) = (J_m s + D_m) \omega_m(s) \quad (\text{C.9})$$

where  $J_m$  is the effective inertia 'seen' by the rotor and  $D_m$  is the drag constant experienced by the rotor. Equation C.8 then becomes

$$\begin{aligned} E_a(s) &= \frac{R_a + L_a s}{K_t} (J_m s + D_m) + K_b \omega_m(s) \\ &= \omega_m(s) \left[ \frac{(R_a + L_a s)(J_m s + D_m) + K_b K_t}{K_t} \right] \end{aligned} \quad (\text{C.10})$$

This leads to a transfer function given by

$$\begin{aligned} \frac{\omega(s)}{E_a s} &= \frac{K_t}{(J_m s + D_m)(R_a + L_a s) + K_t K_b} \\ &= \frac{K_t}{s^2 + \frac{D_m L_a + R_m J_m}{L_a J_m} s + \frac{D_m R_a + K_b K_t}{L_a J_m}} \end{aligned} \quad (\text{C.11})$$

We can simplify this further by making the assumption that the armature inductance,  $L_a$ , is small compared to the armature resistance,  $R_a$  i.e  $R_a + L_a s \approx R_a$ . Equation C.11 then becomes

$$\begin{aligned}
 \frac{\omega(s)}{E_a s} &= \frac{K_t}{(J_m s + D_m)(R_a) + K_t K_b} \\
 &= \frac{K_t}{J_m R_a s + D_m R_a + K_t K_b} \\
 &= \frac{\frac{K_t}{R_a J_m}}{s + \frac{1}{J_m} (D_m + \frac{K_t K_b}{R_a})} \\
 &= \frac{K}{s + \alpha}
 \end{aligned} \tag{C.12}$$

where  $\alpha = \frac{1}{J_m} \left( D_m + \frac{K_t K_b}{R_a} \right)$

## Appendix D

# Frequency Analysis

### D.1 Bode Plots

Frequency analysis techniques may be used to obtain an estimate of the transfer function. The frequency response of a system is represented in a *bode plot*, in which both the dB gain and phase difference is plotted against frequency. This system is convenient since the effects of the poles and zeros will be additive in the dB scale.

The Bode Plot shown in Figure D.1 uses Equation 8.6 with  $\zeta = 0.25$  and  $\omega_n = 12$ . This chapter describes how the Bode Plot can be used to arrive at these values independently.

Equation 8.6 is repeated for clarity.

$$G(s) = \frac{K}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (\text{D.1})$$

Substituting  $s = j\omega$  into Equation D.1 and multiplying the denominator by its complex conjugate we obtain the magnitude of the transfer function as

$$M(\omega) = \frac{K}{\sqrt{(\omega_n^2 - \omega^2)^2 + 4\zeta^2\omega_n^2\omega^2}} \quad (\text{D.2})$$

To evaluate the frequency at which the magnitude of the transfer function,  $M(\omega)$ , is a maximum, we square  $M(\omega)$  (to remove the root).

$$M^2(\omega) = \frac{K}{(\omega_n^2 - \omega^2)^2 + 4\zeta^2\omega_n^2\omega^2} \quad (\text{D.3})$$

$$= \frac{K}{\omega_n^4 + (4\zeta^2\omega_n^2 - 2\omega_n^2)\omega^2 + \omega^4} \quad (\text{D.4})$$

$$= \frac{f(\omega^2)}{g(\omega^2)} \quad (\text{D.5})$$

We now differentiate  $M^2(\omega)$  with respect to  $\omega^2$  to arrive at

$$\begin{aligned} \frac{dM^2(\omega)}{d\omega^2} &= \frac{f'(\omega^2)g(\omega^2) - g'(\omega^2)f(\omega^2)}{g(\omega^2)^2} \\ &= \frac{-K[2\omega^2 + [-2\omega_n^2 + 4\zeta^2\omega_n^2]]}{g(\omega^2)^2} \end{aligned} \quad (\text{D.6})$$

Setting the numerator equal to zero we have

$$\begin{aligned} 0 &= 2\omega_p^2 - 2\omega_n^2 + 4\zeta^2\omega_n^2 \\ \omega_p^2 &= \omega_n^2 [1 - 2\zeta^2] \\ \omega_p &= \omega_n \sqrt{1 - 2\zeta^2} \end{aligned} \quad (\text{D.7})$$

Substituting this back into Equation D.2 we arrive at the an expression for the peak magnitude,  $M_p$

$$M_p = \frac{1}{2\zeta\sqrt{1-\zeta^2}} \quad (\text{D.8})$$

An estimate of  $\zeta$  can be obtained by examining the magnitude of the resonance peak and using Equation D.8.

$M_p$  is the difference between the peak value of the transfer function and the starting value. In the case of Figure D.1,we see this is  $((-36.9) - (-43.1)) \text{ dB} = 6.2 \text{ dB}$ . This is a peak magnitude of 2.0417 times greater than the starting magnitude. Equation D.8 can be rewritten as

$$-4\zeta^4 + 4\zeta^2 - \frac{1}{M_p^2} = 0 \quad (\text{D.9})$$

and the roots of this equation can be determined. In this case (using  $M_p = 2.0417$ ) we obtain roots at 0.9674 and 0.2531 with  $\zeta = 0.2531$ . The Bode Plot was constructed using  $\zeta = 0.25$  and this agrees well.

The natural frequency,  $\omega_n$ , can be estimated from the first peak frequency,  $\omega_p$ , at 11.2 rad/s using Equation D.7. Using the estimate of  $\zeta = 0.2532$  gives  $\omega_n = 11.97 \text{ rad/s}$  (as compared to 12 rad/s).

Using this method estimates of many of the important antenna parameters can be obtained fairly easily.

Bode plots can be used to estimate the bandwidth,  $\omega_{BW}$  of the rate-loop system. This is defined as the frequency at which the frequency response curve drops below 3 dB of its starting value and gives an idea of the response speed. Large bandwidth allows for shorter system response times. Relationships between the bandwidth and other properties of a second order system are given below

$$\omega_{BW} = \omega_n \sqrt{(1 - 2\zeta^2) + \sqrt{4\zeta^4 + 4\zeta^2 + 2}} \quad (\text{D.10})$$

$$= \frac{4}{T_s \zeta} \sqrt{(1 - 2\zeta^2) + \sqrt{4\zeta^4 + 4\zeta^2 + 2}} \quad (\text{D.11})$$

$$= \frac{\pi}{T_p \sqrt{1 - \zeta^2}} \sqrt{(1 - 2\zeta^2) + \sqrt{4\zeta^4 + 4\zeta^2 + 2}} \quad (\text{D.12})$$

It is important to note that should the transfer function include additional poles or zeros, then we cannot expect the quadratic term to contribute entirely towards the bode plot. Further approximations can be made by looking for evidence of additional poles or zeros on the bode plot (Ogata, 1970) and incorporating these into the experimental transfer function. The process can continue iteratively until reasonable agreement between predictions and experiment is reached.

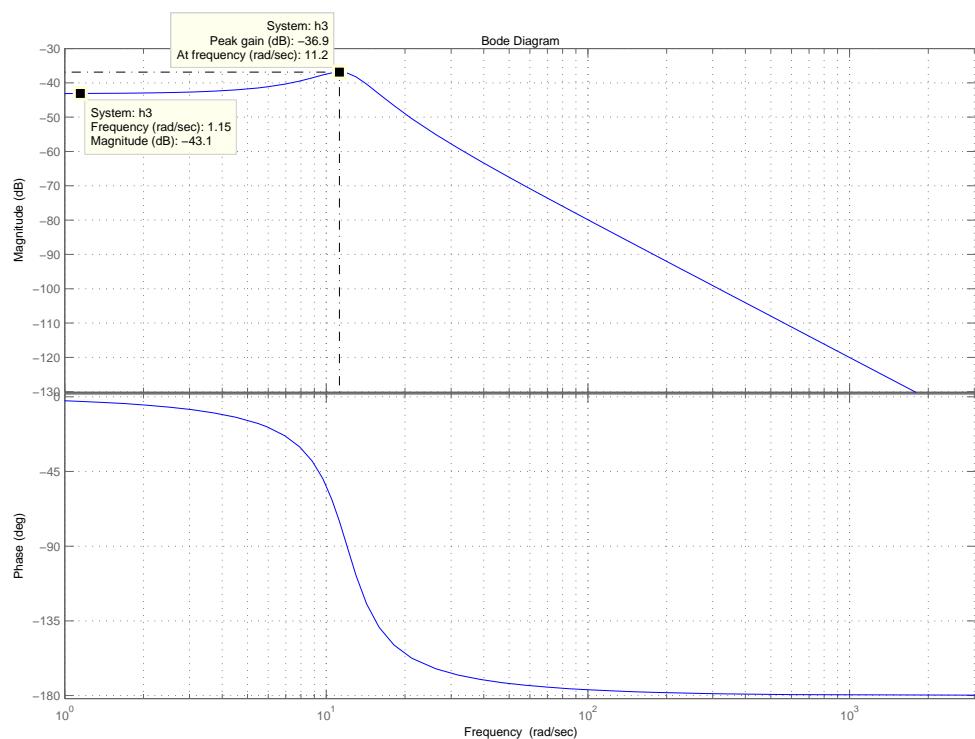


Figure D.1: Bode plots of the the second order transfer function given in Equation 8.6  
with  $\omega_n = 12$  and  $\zeta = 0.25$

## Appendix E

# Ethernet Communication

### E.1 Ethernet Communication

Ethernet communication is based on the ISO OSI Reference Model. This comprises seven layers as shown in Figure E.1.

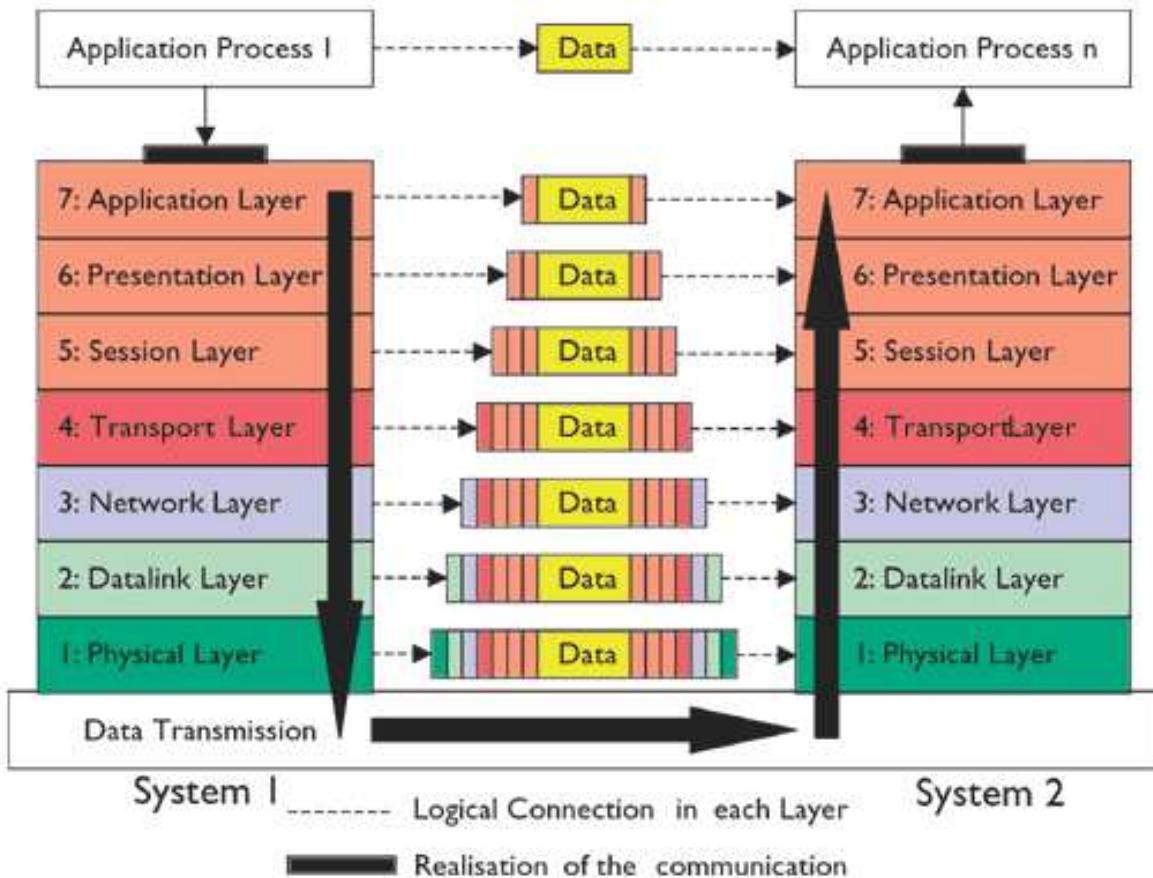


Figure E.1: The seven layers that comprise the OSI Reference Model (Schwab, 2004)

In the OSI model, data is passed down from the upper layers until it reaches the bottom (physical layer) with information being added to the original data packet at each layer. This is known as *encapsulation*. The resulting data packet is then sent over the physical connection. When it reaches its destination the data is moved up through the layers until it again reaches the top layer where the data can be read. Generally each layer adds (on transmitting) or removes (on receiving) information to/from the packet. This layer information is used for

addressing information and providing other checks to ensure data delivery and fidelity.

The TCP/IP Reference Model was used as the protocol in this study. This forms a subset of the more general OSI Reference Model. This is the model that is primarily used for communication across the internet today and has been standardised. This is a four layer Reference Model where each layer adds (or removes) a header from the data packet. The headers provide various degrees of packet information depending on which protocols are implemented, however all protocols define addressing and data size information together with an error checking algorithm. Figure E.2 shows the TCP/IP Reference Model together with each protocol's data header as implemented with the UDP transport protocol.

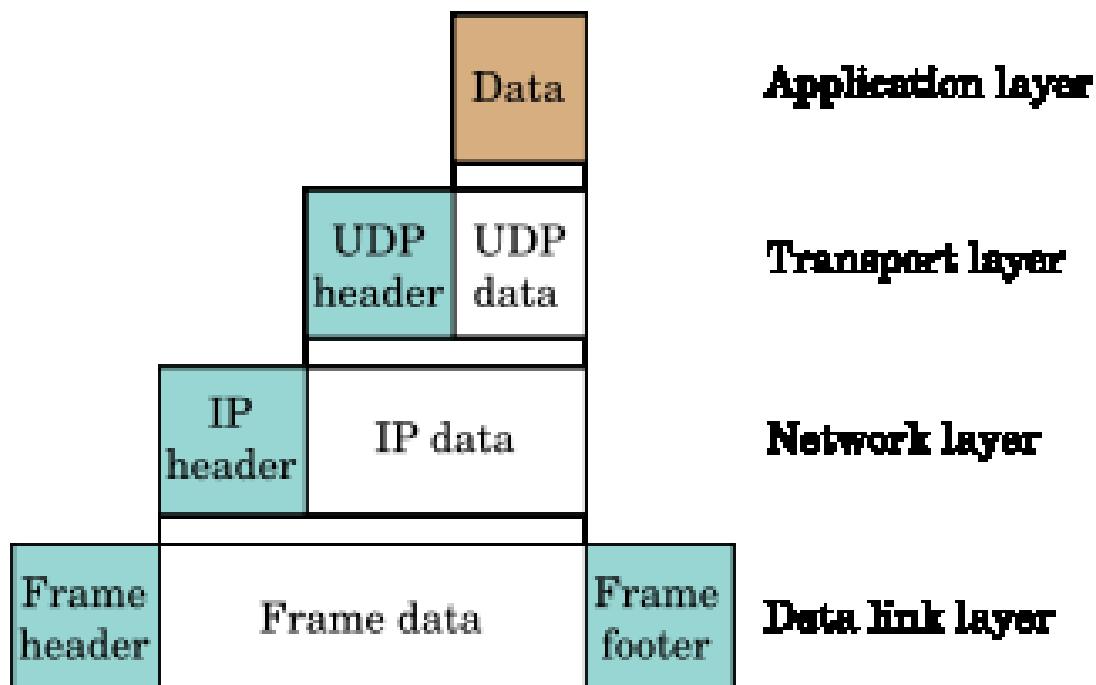


Figure E.2: A diagrammatic representation of data encapsulation using UDP (Wikipedia, 2007b)

TCP/IP packets need to be appropriately constructed before they can be sent. Similarly a received packet needs to be processed before the data payload is arrived at. This takes a certain number of CPU clockcycles and can require large memory resources. Since a microcontroller has severely limited resources compared to a PC, a careful implementation of the TCP/IP stack is required. In the case of a control system, packets need to be processed rapidly and RAM constraints need to be considered throughout.

## E.2 Data Frames

### E.2.1 Ethernet Frames (Datalink Layer)

A typical Ethernet frame must define a 6 byte source a 6 byte destination MAC address and a 2 byte Ethernet type definition as shown in Figure E.3. MAC addresses are the hardware addresses of the network cards involved in the data transaction and this forms a simple method of addressing packets. With these defined it is possible to send a data packet from one computer to another on a local network. The packet addressing is simple and requires little processing power, making it suitable for microcontroller interfacing. Unfortunately PC side software to support packets with only these Ethernet headers prepended is not extensively

tested and used. The potential speed and overhead advantages were unlikely to outweigh the difficulty in designing the necessary computer systems and I chose to rather pursue the standard TCP/IP protocol. This allows for standard libraries to be used in designing the PC side communication.



Figure E.3: A typical Ethernet frame (Wikipedia, 2007a)

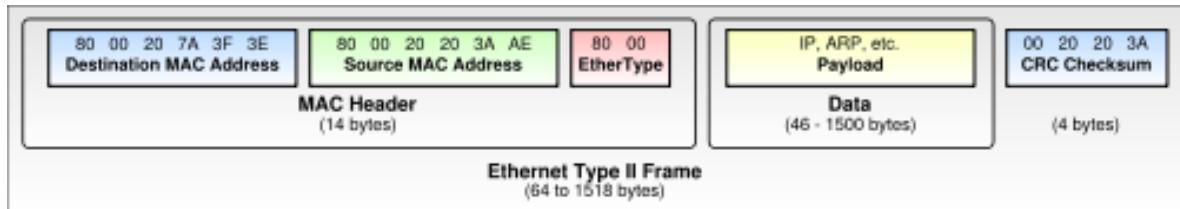


Figure E.4: A commonly used Ethernet Type II frame (Wikipedia, 2007a)

### E.2.2 IP and ARP (Internet Layer)

IP (Internet Protocol) is a means of addressing data packets for transmission over various Datalink Layers (i.e Ethernet, Wi-Fi and others). IP allows data packets to be sent over heterogeneous Datalink Layers with addressing information intact. ARP is a means of determining destination addresses (almost always IP addresses) and mapping these to hardware addresses (i.e MAC addresses). The abstraction from the Datalink Layer allows ARP to work over any type of physical network. In general, given the prevalence of IP addressing, IP and ARP are closely related with ARP being a protocol for mapping IP addresses to MAC addresses. In the case of this project the implemented ARP protocol assumes IP to MAC address mapping. ARP frame examples are included in Appendix E.4, Figure E.11 and E.12

### E.2.3 TCP/UDP Frames (Transport Layer)

The TCP/IP Reference Model allows for connection-oriented (TCP) and connectionless (UDP) connections. The difference between the two is significant and a brief explanation of both is necessary.

The TCP Transport Protocol ensures guaranteed delivery. This is achieved primarily through a mechanism known as 'flow control' which can change to accommodate different network loads, computer loads and many other variables that are inherent to an Ethernet network. TCP 'flow control' works by reducing the transmission frequency by 50% every time a packet is not received. While the speed does gradually increase subsequently, the rate is controlled by the TCP software and is difficult to control. TCP ensures error free and sequential data transmission over the network as shown in Figure E.5a. A TCP frame construction is shown in Figure E.5c. The frame allows for many definitions all lending themselves to making sure that the data arrives safely at the destination. TCP allows for transparent communication.

UDP does not concern itself with guaranteed delivery. This can be a problem if the network is congested, as collisions may occur and the packet may not arrive at its destination.

There are advantages to using UDP over TCP. The rigorous error checking, sequencing and other built in checks found in the TCP protocol are not always necessary. It is possible to build the required functionality into a UDP frame with a little more time spent configuring both sides of the connection and removing the inevitable bugs. As a result of this, it is possible to reduce the overhead involved in constructing the data packets which is particularly useful in microcontroller applications. UDP offers significant speed advantages over TCP primarily because the flow control is removed. UDP however allows for configuring at this level.

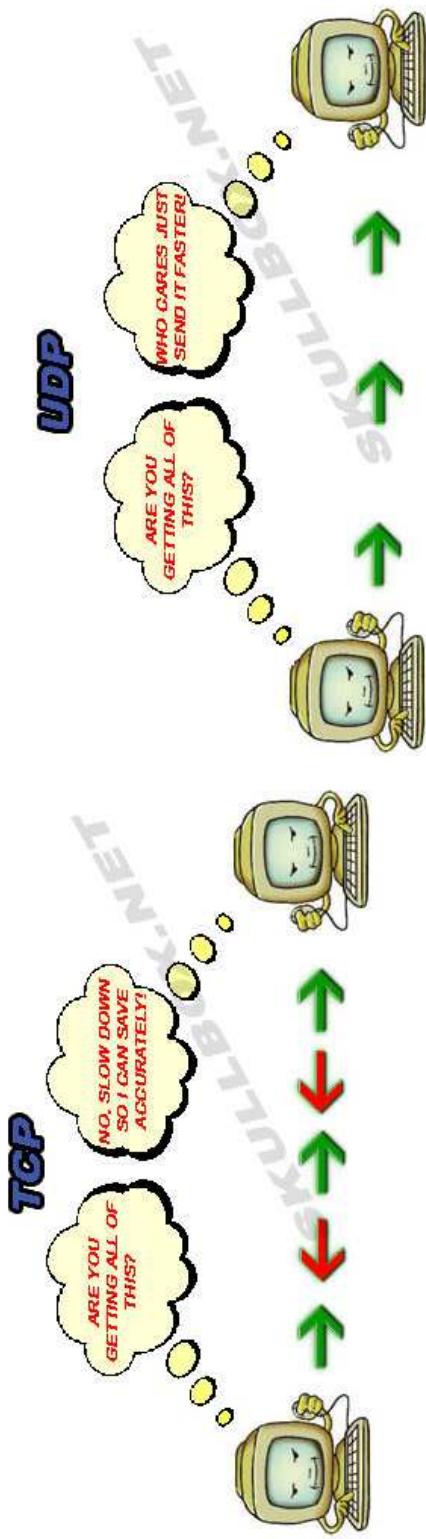
UDP thus offers the following advantages over TCP:

- Speed
- Overhead
- Configurability

These are achieved together with the following disadvantages:

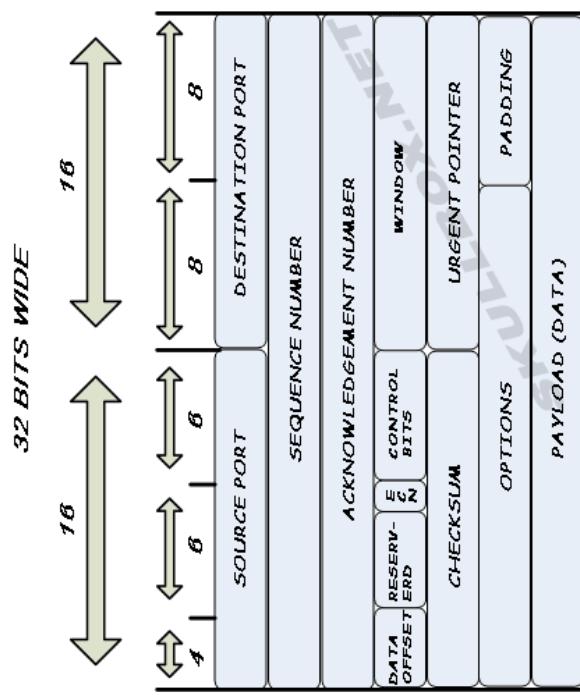
- Unreliability
- Unsequential Data

The relative size of the TCP and UDP headers shown in Figure E.5c and E.5d respectively, demonstrate these advantages and disadvantages clearly. It should be noted that the primary disadvantages of UDP are largely negligible on a dedicated local network where congestion is low.



(a) Depiction of typical TCP data transmission. The data transmission uses flow control and acknowledgements to ensure that all data is received in the same order in which it is sent

(b) Depiction of typical UDP data transmission. No acknowledgements allow faster data transfer with lower overheads, however data delivery is not guaranteed.



(c) The layout of the TCP header used in data transmission.

(d) The layout of the UDP header used in data transmission.

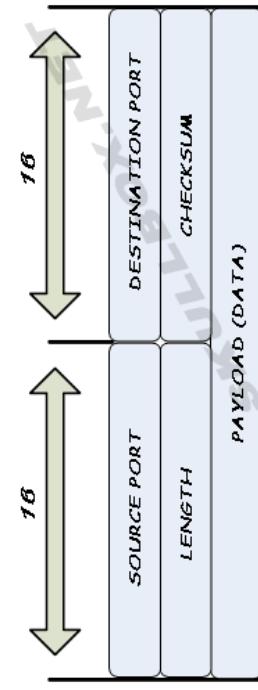


Figure E.5: Comparison between UDP and TCP packets and data transmission (Rodriguez). These figures show the difference between the two transport protocols.

## E.3 Time Keeping

Maintaining synchronisation between the microcontroller and PC time is essential to any control loop which relies on both sets of equipment. It is especially important since the Ethernet connection has variable transmission latencies which needs to be overcome.

The time standard adopted for the microcontroller was the same as that used on a UNIX system. The start epoch is the 1/1/1970 and the time is stored as an unsigned long integer allowing a maximum of 4 294 967 295 seconds to pass. Since more accurate time keeping was required, I used a 16 bit timer counter set to a maximum resolution of 62500 (chosen as a convenient fraction of the 16MHz system clock crystal). This allowed timing with a potential accuracy of 1/62500 of a second or 16  $\mu$ s. A 1 pulse-per-second (1PPS) signal was used to provide a reset to this timer which allows good synchronisation with other systems using the same 1PPS signal.

### E.3.1 NTP/SNTP

The standard method of time synchronisation in networked systems implementing the TCP/IP Reference Model is the SNTP or NTP server. This relies on timestamped packets being sent between an SNTP or NTP server and a client. Since a 1PPS signal should be available it should only be necessary to synchronise the computer and the microcontroller to the nearest second using SNTP/NTP. This presents no problems and synchronisation of the order of 10 ms is obtainable using SNTP.

NTP operates in a series of layers known as strata. A Stratum 1 server is the highest NTP server and is usually synchronised to an atomic clock. A Stratum 2 server is then synchronised to a Stratum 1 server and so synchronisation is maintained. Generally any network will have from 1 to 5 NTP servers, each of which synchronises its time to a server of a higher (lower number) Stratum than itself. Computers on the network will then synchronise with their local NTP servers (see Figure E.6)

NTP/SNTP can in be implemented in two 'flavours':

- Unicast time updates
- Multicast time updates

The first of these requires a request from the client (i.e the microcontroller). The second assumes no such request to be forthcoming and time update packets are sent at a predefined frequency to a multicast IP group (Holmeide and Skeie, 2001).

The first method allows better synchronisation and makes use of the following system.

1. Client generates a reference time  $T_0$  and puts this into a time request packet. It then composes a time request packet and stores the local time  $T_1$  when the packet is sent.
2. Server copies  $T_0$  into the time reply packet. This allows pairing of requests and responses. It generates a server timestamp when the packet is received  $T_2$  and when the packet is sent  $T_3$  and includes these in the time reply packet.
3. Packet arrives at client containing  $T_1$  and  $T_2$ . Client stores the arrival time  $T_4$  of the packet according to the local clock

From these times it is possible to calculate the propagation time of the packet  $T_{pd}$  and the clock offset  $T_{offset}$  using the following equations (Holmeide and Skeie, 2001):

$$T_1 + T_{offset} + T_{pd} = T_2 \quad (\text{E.1})$$

$$T_3 - T_{offset} + T_{pd} = T_4 \quad (\text{E.2})$$

$$T_{pd} = \frac{(T_2 + T_3) - (T_4 + T_1)}{2} \quad (E.3)$$

$$T_{offset} = \frac{(T_2 - T_1) + (T_3 - T_4)}{2} \quad (E.4)$$

### E.3.2 NTP Strata

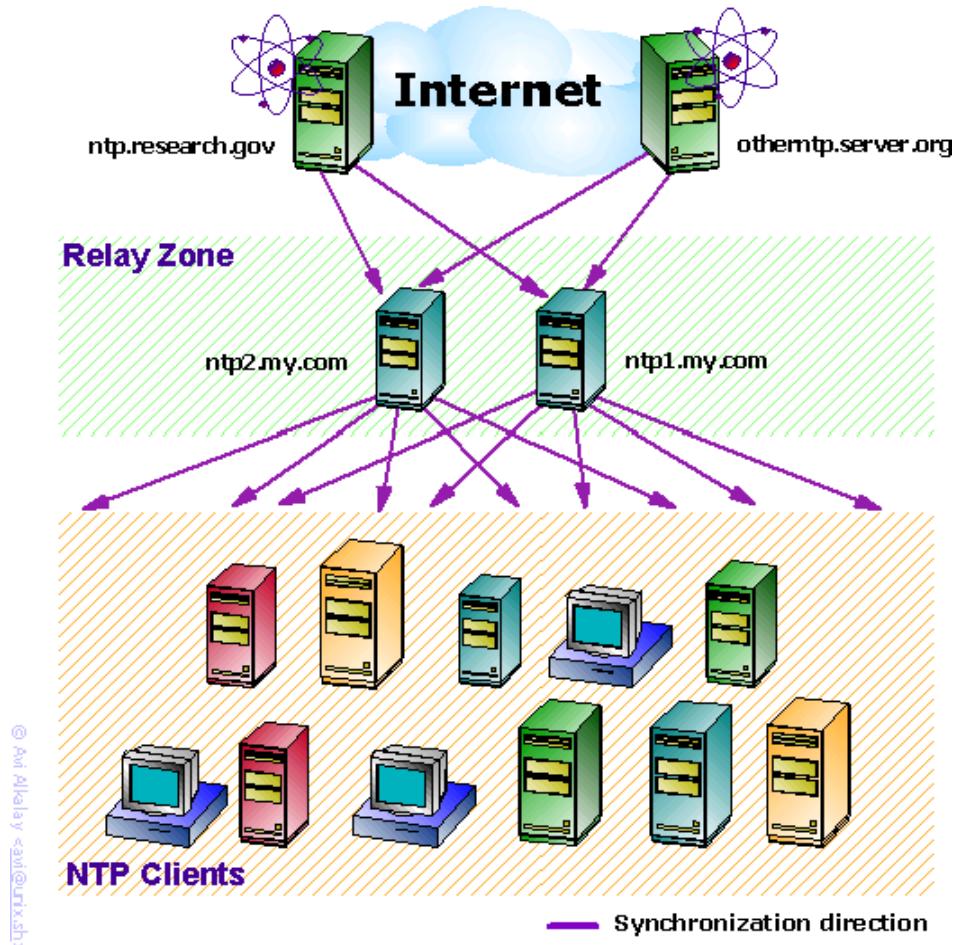


Figure E.6: An overview of the NTP time synchronisation process

### E.3.3 Different Time Stamping Options for SNTP

#### Time stamping in the SNTP/NTP application layer

A typical SNTP/NTP application will include the time stamping in the application layer. This is represented in Figure E.8. This does take a certain amount of time to process and thus limits the potential accuracy. Since a 1PPS signal is used this shouldn't be a problem as the delay is not of the same order as one second.

#### Time stamping in other layers

It is possible to perform the timestamping in other layers. The major problem involved in doing this is that PC side software is written to look for the timestamping in the application layer. The effort that would be involved in writing a customised version would not be worth it since the 1PPS signal would allow synchronisation down to the accuracy required without

the need for PC software customisation. An added advantage of keeping the timestamp in the application layer is that it is in keeping with the standard and would allow any NTP source worldwide to be used for synchronisation.

0	8	16	24	31
LI	VN	Mode	Stratum	Poll
Root Delay (32)				
Root Dispersion (32)				
Reference Identifier (32)				
Reference Timestamp (64)				
Originate Timestamp (64)				
Receive Timestamp (64)				
Transmit Timestamp (64)				
Authenticator (optional) (96)				

Figure 4. NTP Message Header

Figure E.7: An example of a typical NTP packet

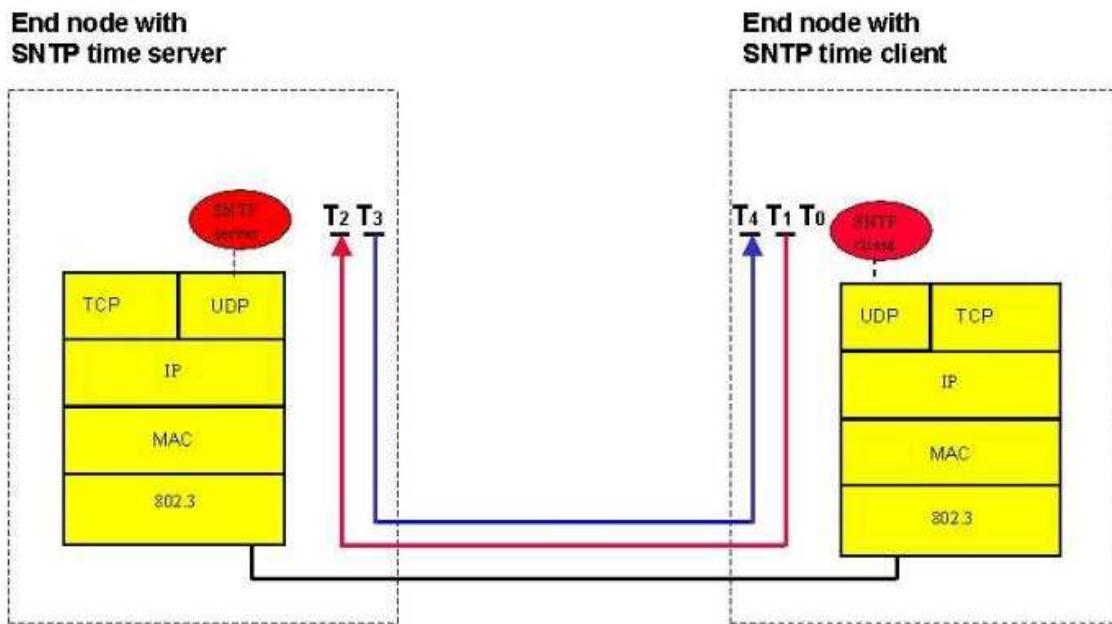


Figure E.8: Time stamping at the application layer

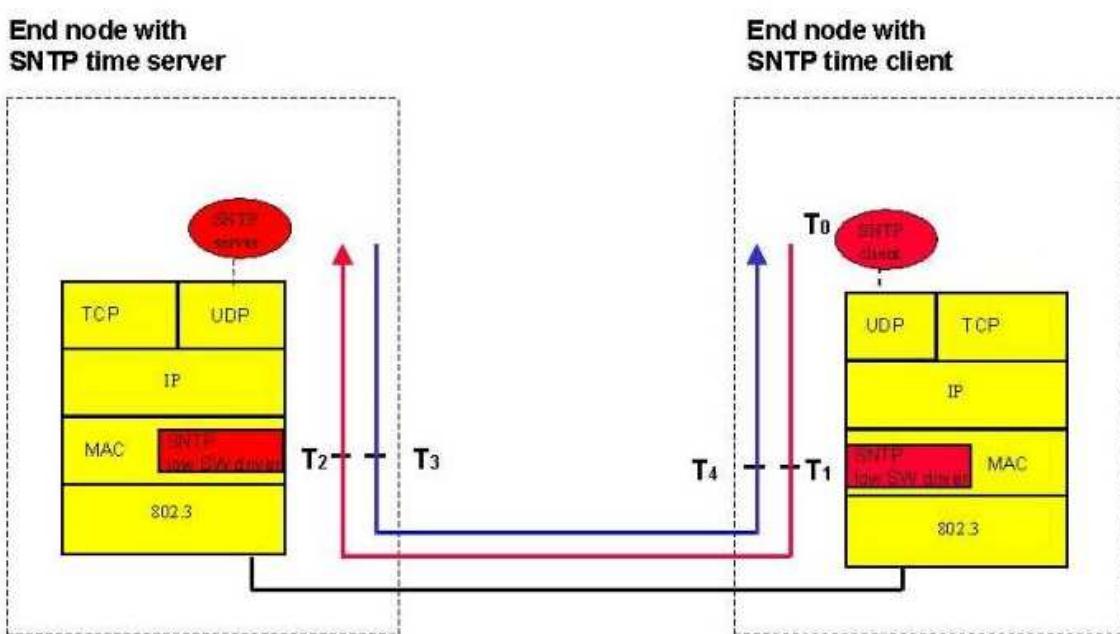


Figure E.9: Time stamping at the Ethernet level

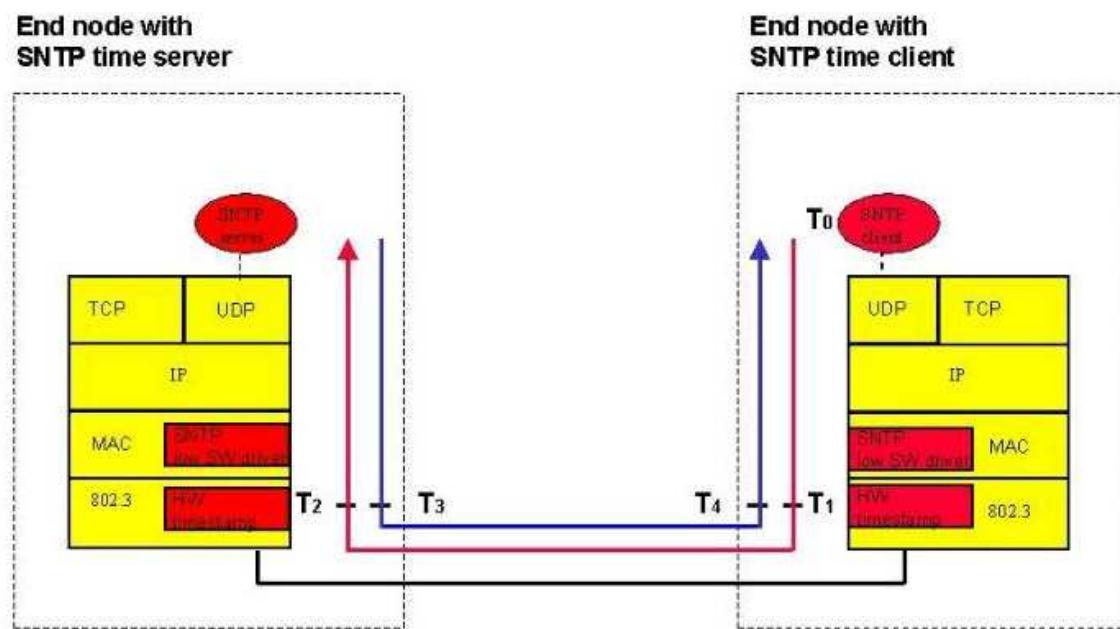


Figure E.10: Timestamp at the datalink/physical layer level

## E.4 ARP Frame Examples

Address Recognition Protocol (ARP) is a protocol that is used to determine the appropriate MAC address for hardware located at a particular IP address. This is done by sending ARP request packets (Figure E.11) and ARP reply packets (Figure E.12).

<b>+</b>	<b>Bits 0 - 7</b>	<b>8 - 15</b>	<b>16 - 31</b>
<b>0</b>	Hardware type = 1		Protocol type = 0x0800
<b>32</b>	Hardware length = 6	Protocol length = 4	Operation = 1
<b>64</b>		SHA (first 32 bits) = 0x000958D8	
<b>96</b>	SHA (last 16 bits) = 0x1122		SPA (first 16 bits) = 0xA0A
<b>128</b>	SPA (last 16 bits) = 0xA7B		THA (first 16 bits) = 0x0000
<b>160</b>		THA (last 32 bits) = 0x00000000	
<b>192</b>		TPA = 0xA0A0A0A8C	

Figure E.11: A typical ARP request frame

<b>+</b>	<b>Bits 0 - 7</b>	<b>8 - 15</b>	<b>16 - 31</b>
<b>0</b>	Hardware type = 1		Protocol type = 0x0800
<b>32</b>	Hardware length = 6	Protocol length = 4	Operation = 2
<b>64</b>		SHA (first 32 bits) = 0x000958D8	
<b>96</b>	SHA (last 16 bits) = 0x33AA		SPA (first 16 bits) = 0xA0A
<b>128</b>	SPA (last 16 bits) = 0xA8C		THA (first 16 bits) = 0x0009
<b>160</b>		THA (last 32 bits) = 0x58D81122	
<b>192</b>		TPA = 0xA0A0A0A7B	

Figure E.12: A typical ARP reply frame

## **Appendix F**

# **Description of the Existing System**

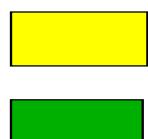
### **F.1 Layout**

#### **F.1.1 Terminal Blocks**

The terminal blocks provide a junction between the DCPA CPU Board and the rest of the DCPA Box. All voltages necessary for successfully monitoring the control system, as well as all voltages necessary for controlling the servopacks are interfaced to the DCPA CPU Board through the terminal blocks.

**104 TB**

0	20
1	21
2	22
3	23
4	24
5	25
6	26
7	27
8	28
9	29
10	30
11	31
12	32
13	33
14	34
15	35
16	36
17	37
18	38
19	39



**Z57**  
**LIMIT SWITCHES**

Figure F.1: Terminal Block 104 TB

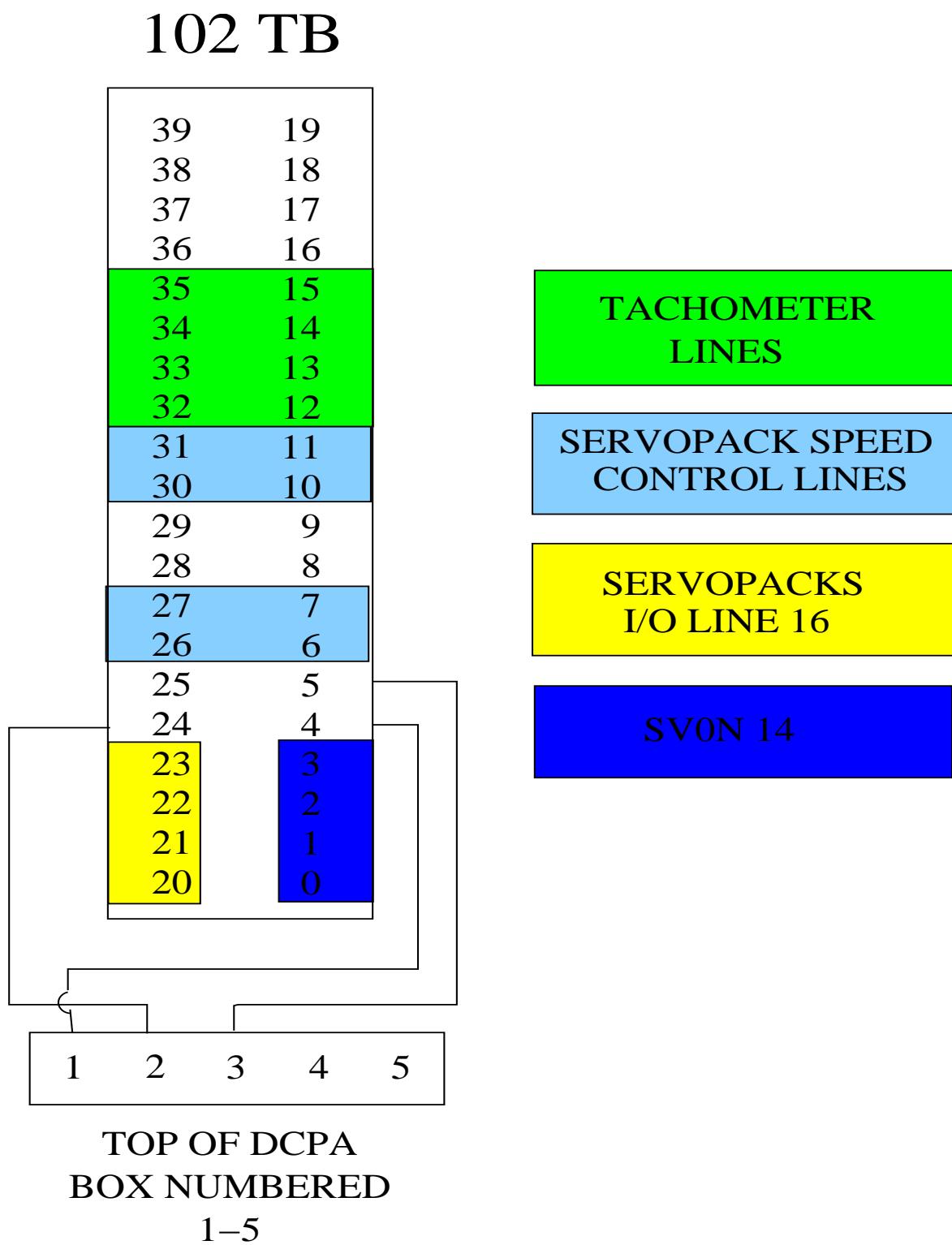


Figure F.2: Terminal Block 102 TB

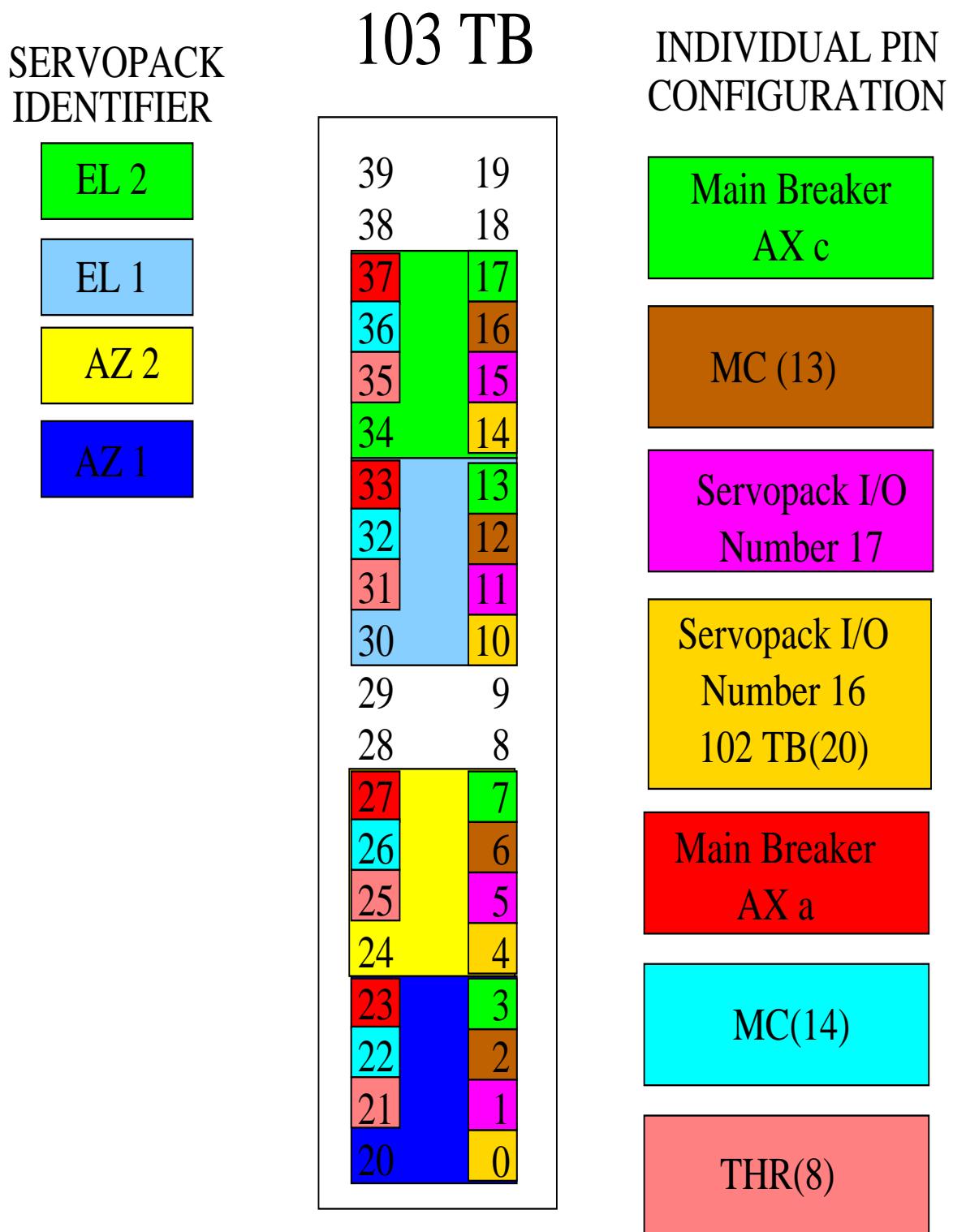


Figure F.3: Terminal Block 103 TB

# 108 TB

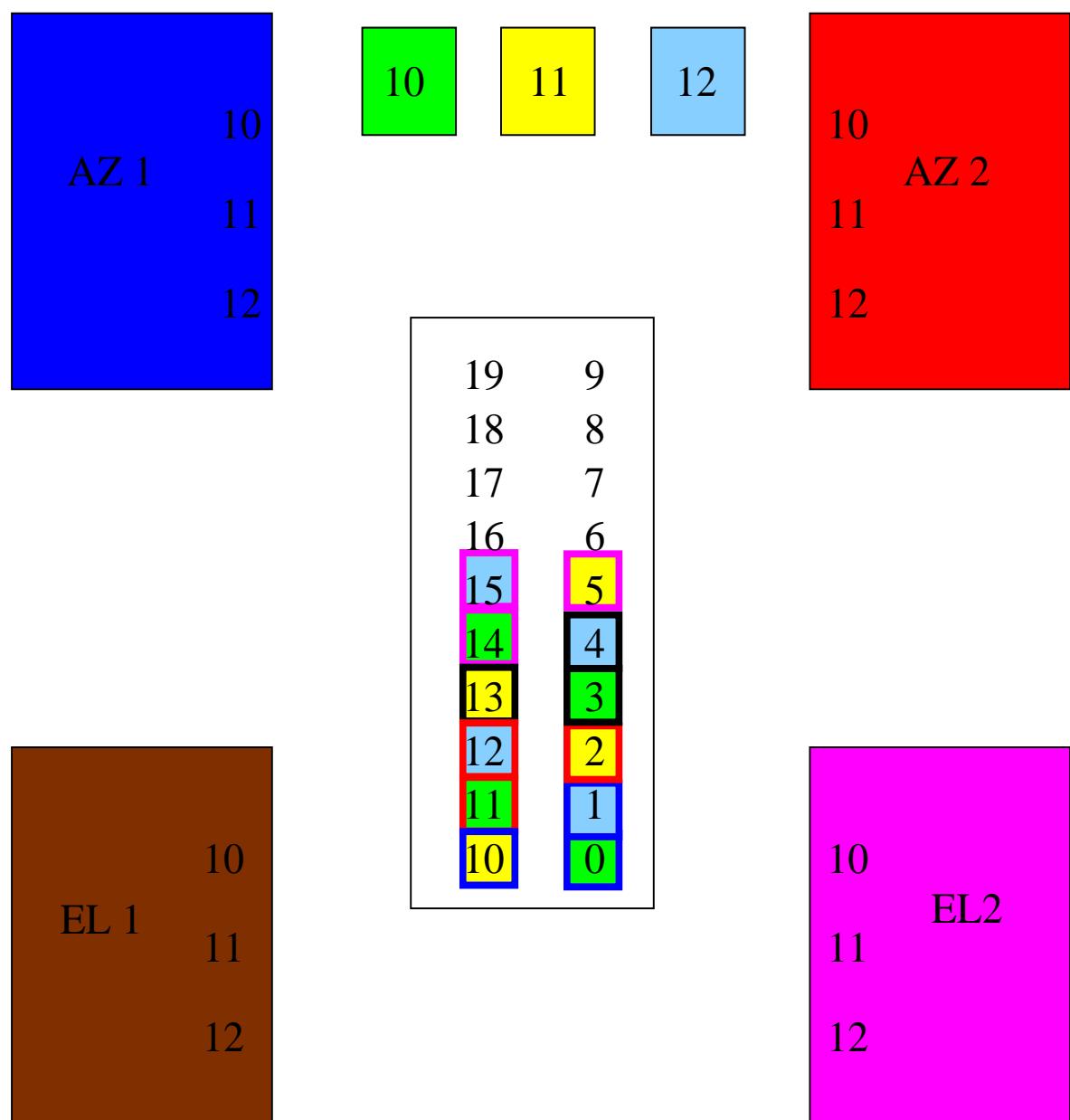


Figure F.4: Terminal Block 108 TB

RFT Training Materials

**NEC**

# DCPA CPU Board Component Layout

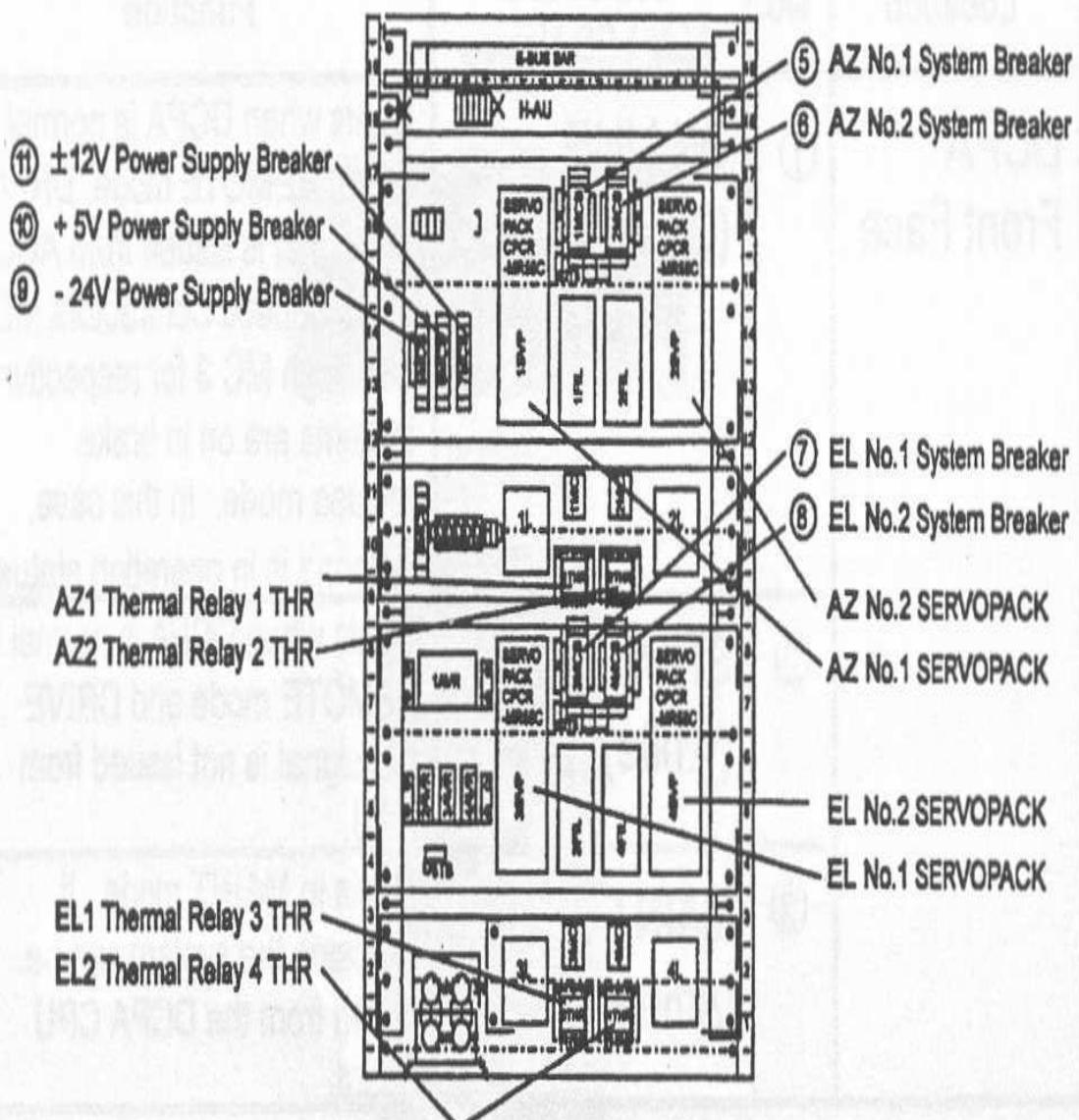


Figure F.5: Layout of the inside of the DCPA box (CBA, 1998)

RFT Training Materials

**NEC**

# BLOCK DIAGRAM OF ANTENNA DRIVE AND SERVO SUBSYSTEM

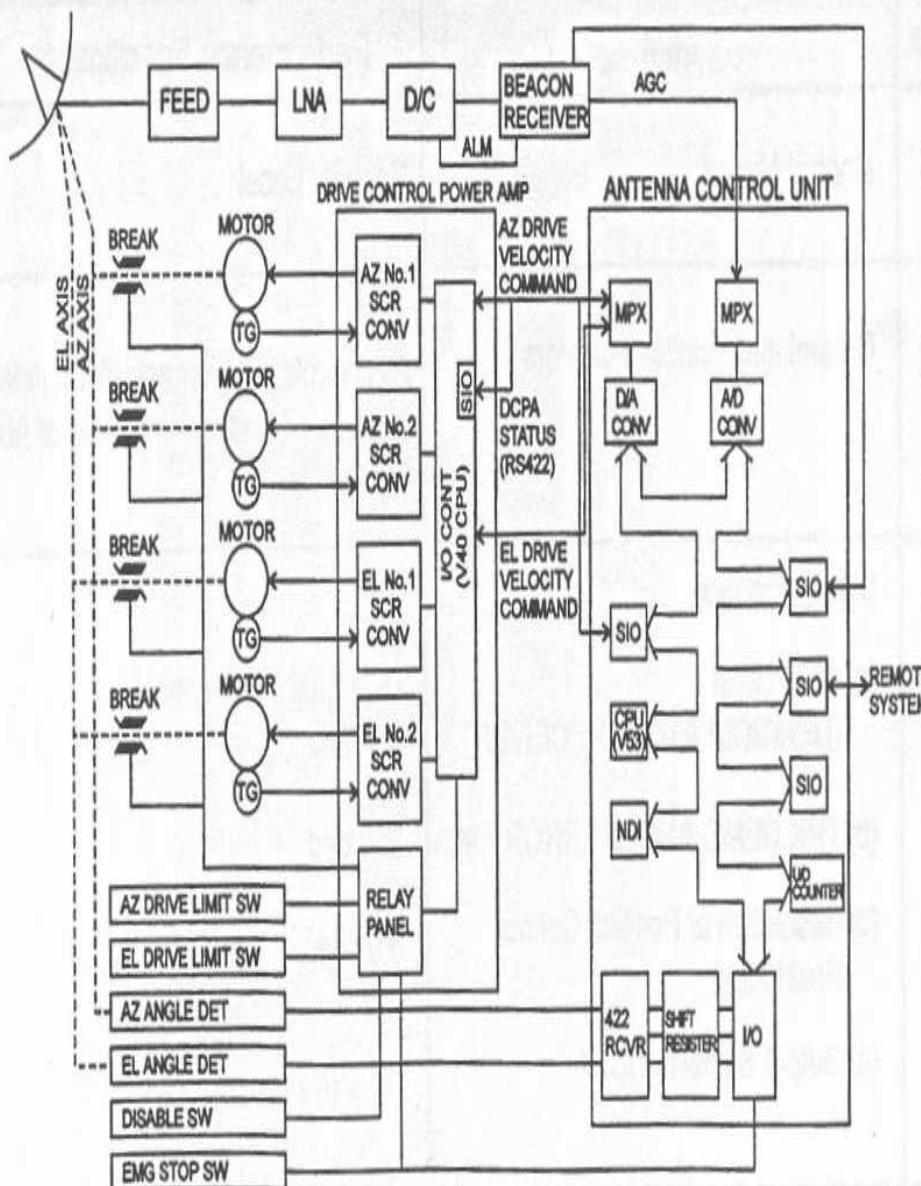


Figure F.6: Overview of the existing control system (CBA, 1998)

RFT Training Materials

**NEC**

# DCPA Simplified Block Diagram

(applicable to antenna on EL axis.)

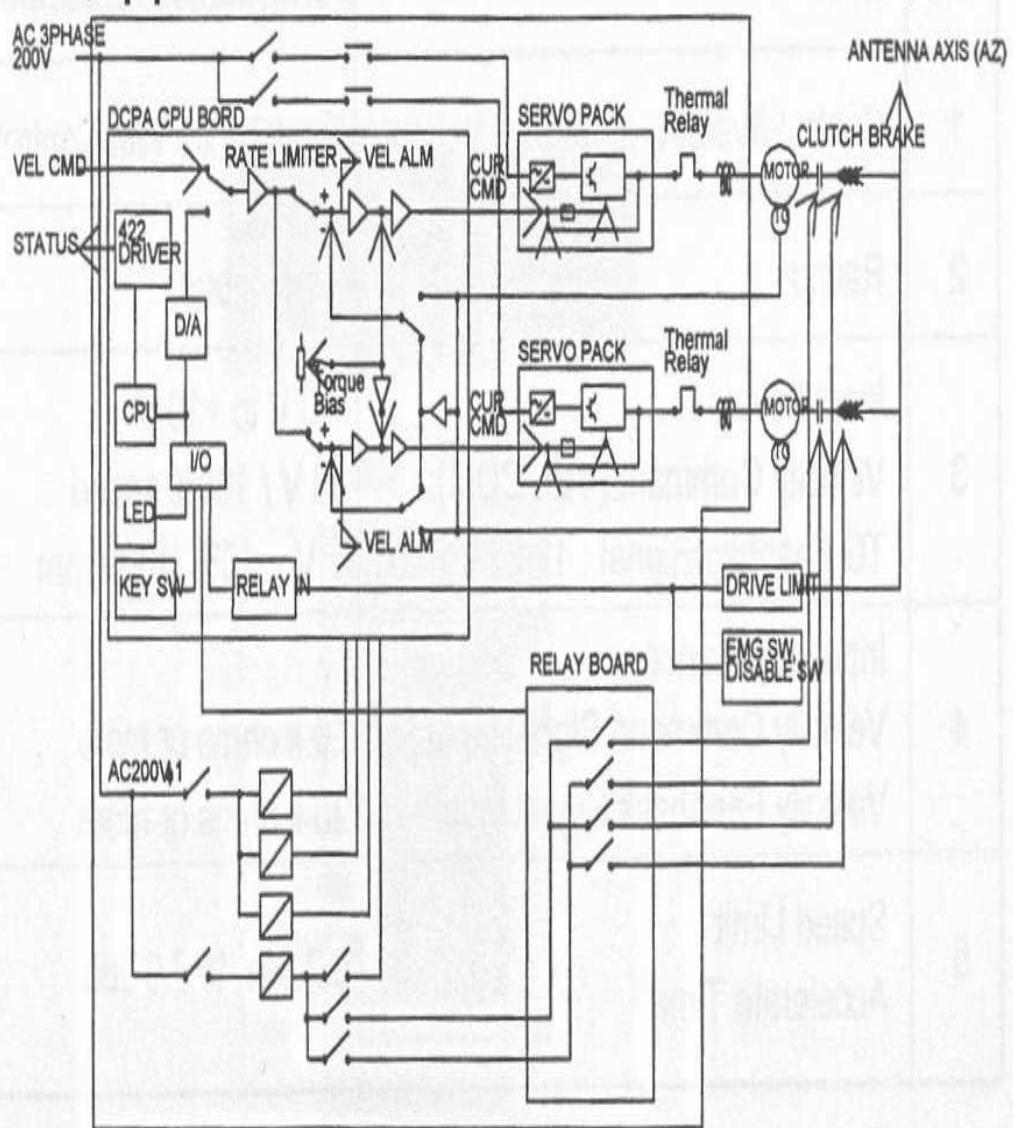


Figure F.7: Diagram of the existing DCPA Control Board and Relay Board (CBA, 1998)

## F.2 Hardware

### F.2.1 DCPA CPU Board

The DCPA CPU Board interfaces to the ACU and, through the Z57 cable, interfaces to a distribution panel. This is the DCPA Status depicted in Figure F.6. This distribution panel is impossible to understand given the knowledge that we have.

Due to this fact, I decided against trying to use the DCPA CPU Board and I didn't spend long examining it. I did gain a good understanding of exactly what its purpose is, and what we should do to replace it.

The purpose of the DCPA CPU Board is to interface with the ACU and provide the necessary command voltages to the servopacks. Tachometer feedback is directed to the DCPA CPU Board (through Z102) as shown in Figure F.6. Command voltages for the servopacks are then returned along Z102 and on 102TB up to the servopacks.

The layout of the board is shown in Figure F.8

### F.2.2 Relay Board

Subject to further investigation, I believe that we could use the Relay Board in the future control system. As such I examined it carefully and came up with the I/O schematic shown in Figure F.9

### F.2.3 Servopack

The servopacks are Yaskawa CPCR-MR55C DC servo amplifiers. Documentation is available on their specifications and wiring diagrams.

### F.2.4 Drive System

#### Motors

There are 4 DC Servomotors. They are positioned with two motors in opposition on each axis so as to minimise backlash effects.

#### Brakes

Each axis is equipped with a disc brake. They are electric brakes which engage if current stops flowing through them. This means that a power failure automatically engages the brakes.

#### Clutch

A clutch is installed on each motor. It is engaged and disengaged in the same way as the brakes are. If the power (and current) is off the clutches will disengage.

### F.2.5 Angle Encoders

The existing angle encoders are Heidenhain ROC416 rotary angle encoders (ROC). These are 16-bit absolute angle encoders that work on an SSI interface. This is a RS485 differential line interface where data is transmitted serially on the rising edge of a data clock. The first ROC416 encoder that we attempted to interface with would not respond. As such we purchased a FRABA OCD-SL00B-0016-S10S-CRW 16 Bit angle encoder with the same SSI interface.

The interface allows for a data transmission clock frequency between 100 KHz and 1 MHz over a distance of up to 100 m.

Table F.1: Description of the DCPA Relay Board I/O Cables

Cable Name	Destination
Z211-1	1 CP Breaker
Z211-2	4 TB-1
Z211-3	2 CP Breaker
Z211-4	4 TB-1
Z212-1	3 CP Breaker
Z212-2	4 TB-2
Z212-3	4 CP breaker
Z212-4	4 TB-2
Z201-Z204	Clutch and Brake Controls
Z206-1	AZ 1 Servopack (r)
Z206-2	AZ 1 Main Breaker (S2)
Z206-3	AZ 2 Servopack (r)
Z206-4	AZ 2 Main Breaker (S2)
Z207-1	EL 1 Servopack (r)
Z207-2	EL 1 Main Breaker (S2)
Z207-3	EL 2 Servopack (r)
Z207-4	EL 2 Main Breaker (S2)

The wiring diagram is shown in Appendix K.2, Figure K.2, K.3 and K.4. The wiring diagram for the FRABA encoder is shown in K.1.

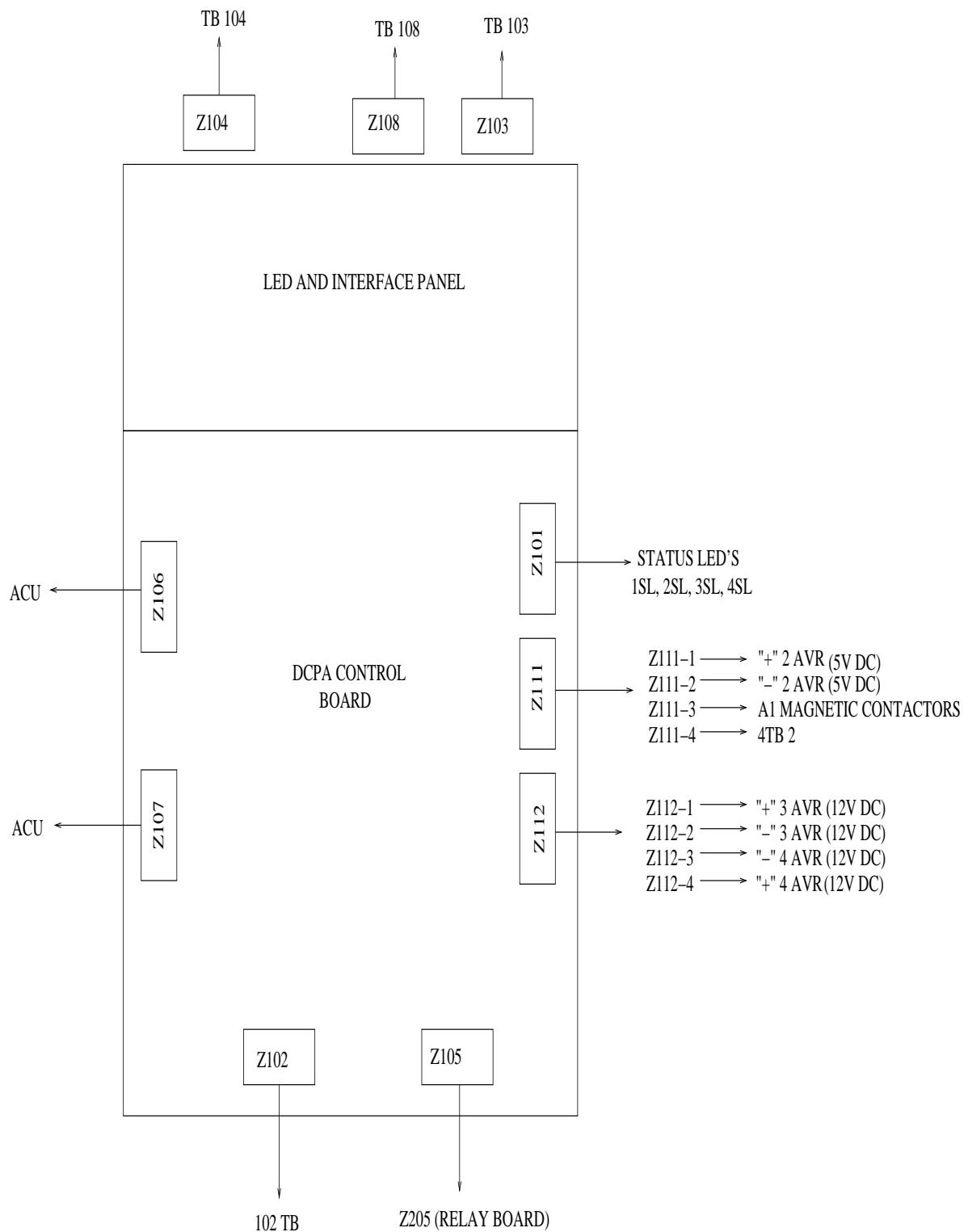


Figure F.8: The layout of the DCPA Control Board

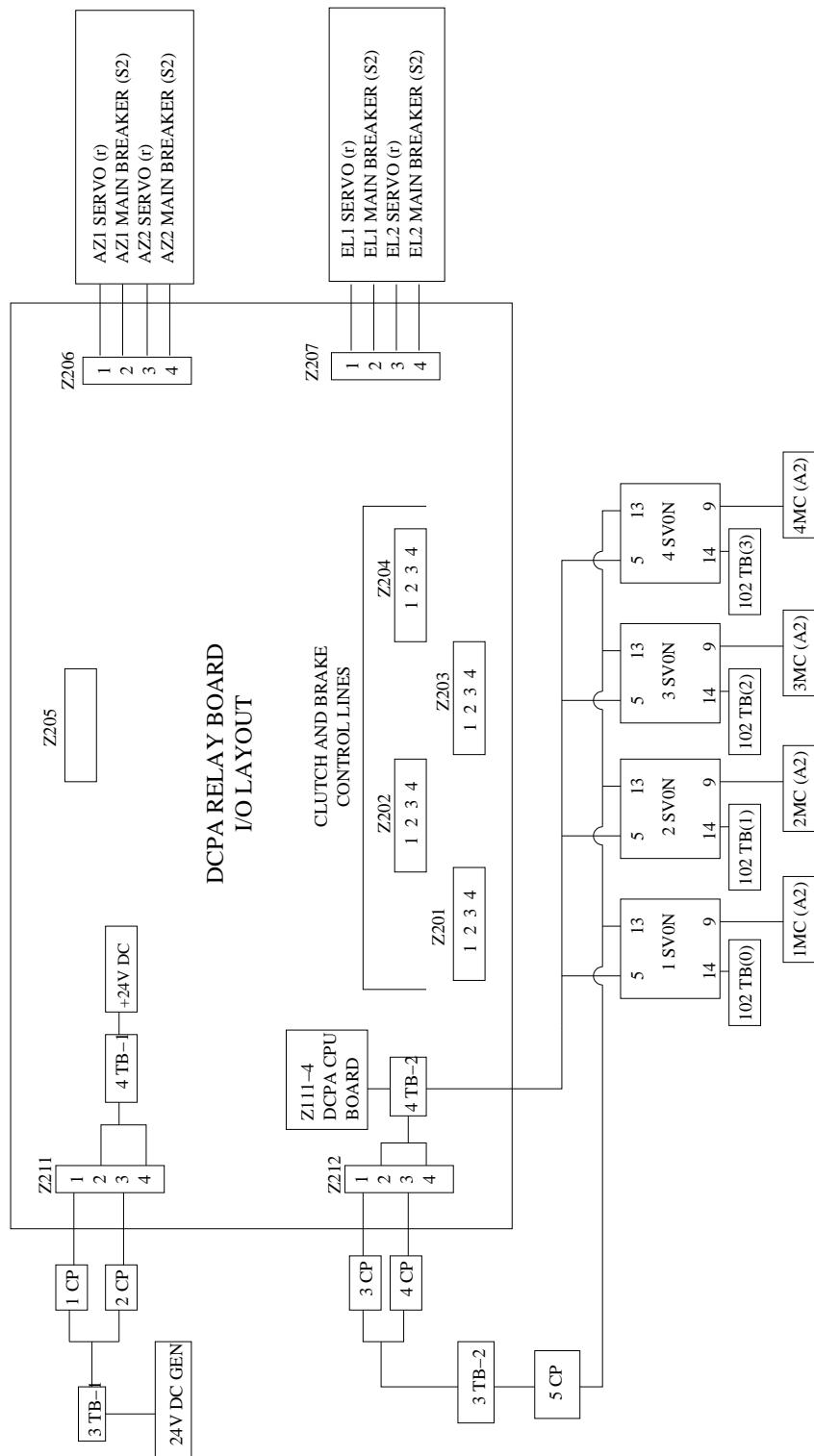


Figure F.9: Diagram of the DCPA Relay Board I/O Ports

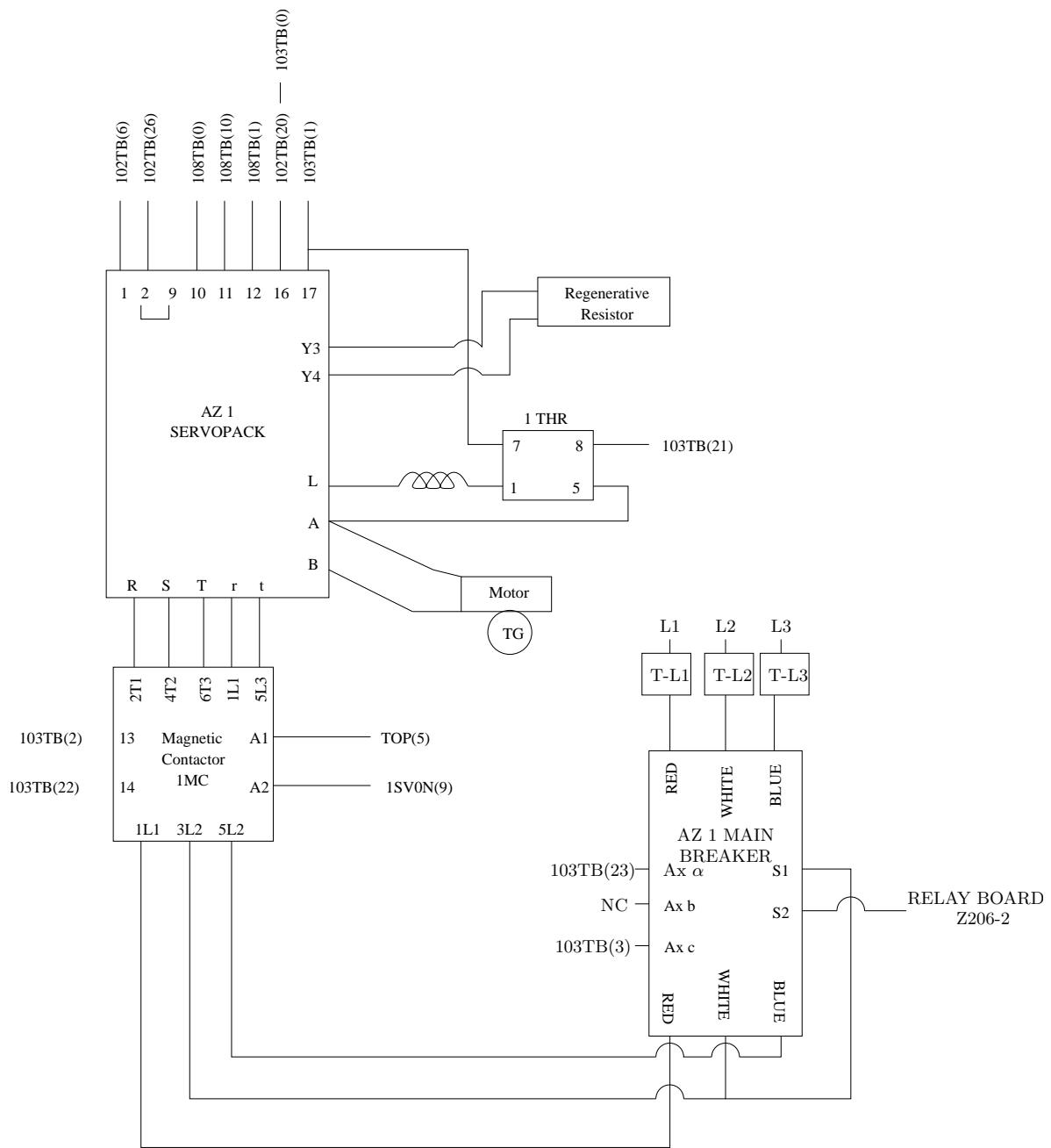


Figure F.10: Wiring Diagram for the AZ1 Servopack

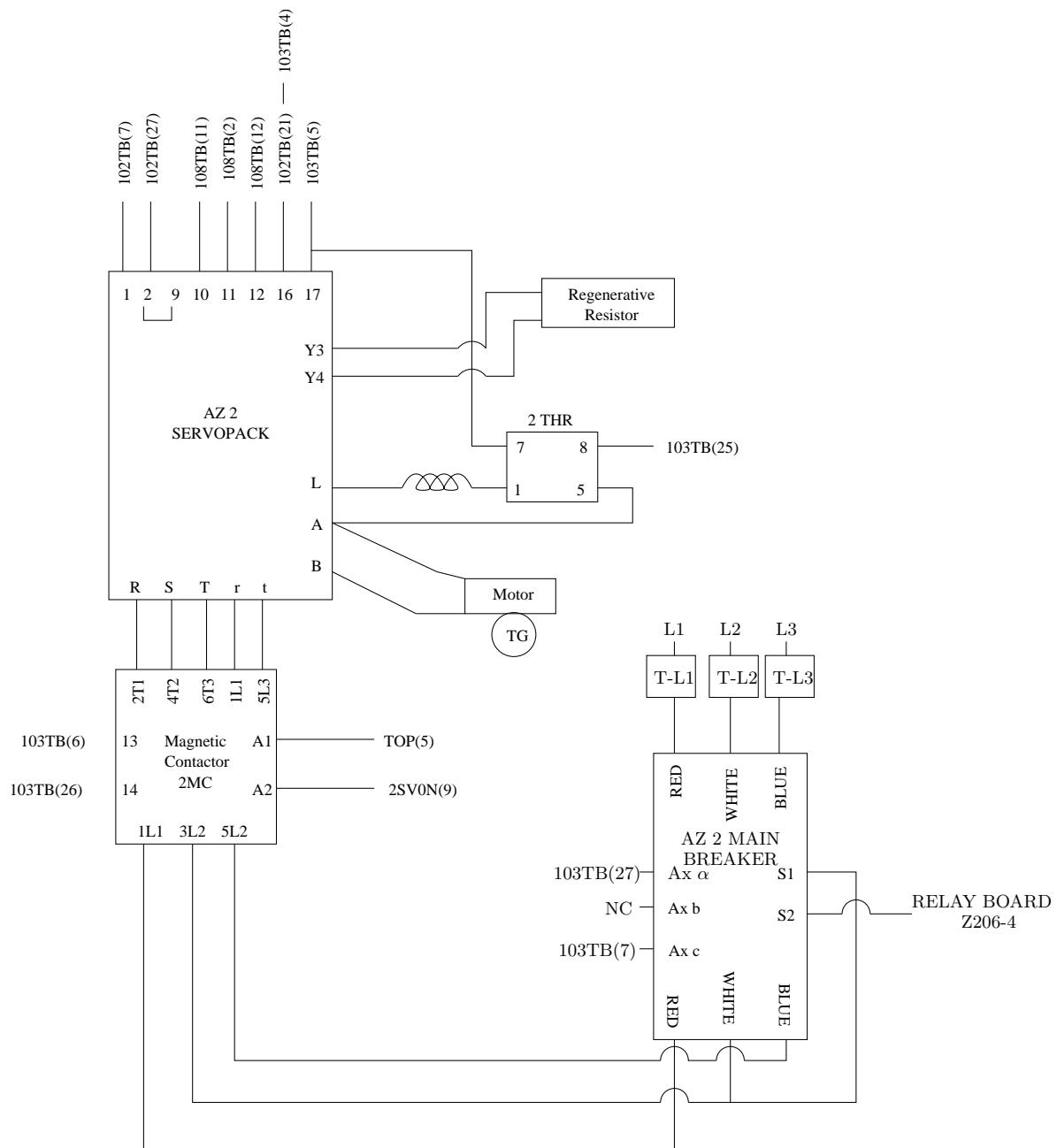


Figure F.11: Wiring Diagram for the AZ2 Servopack

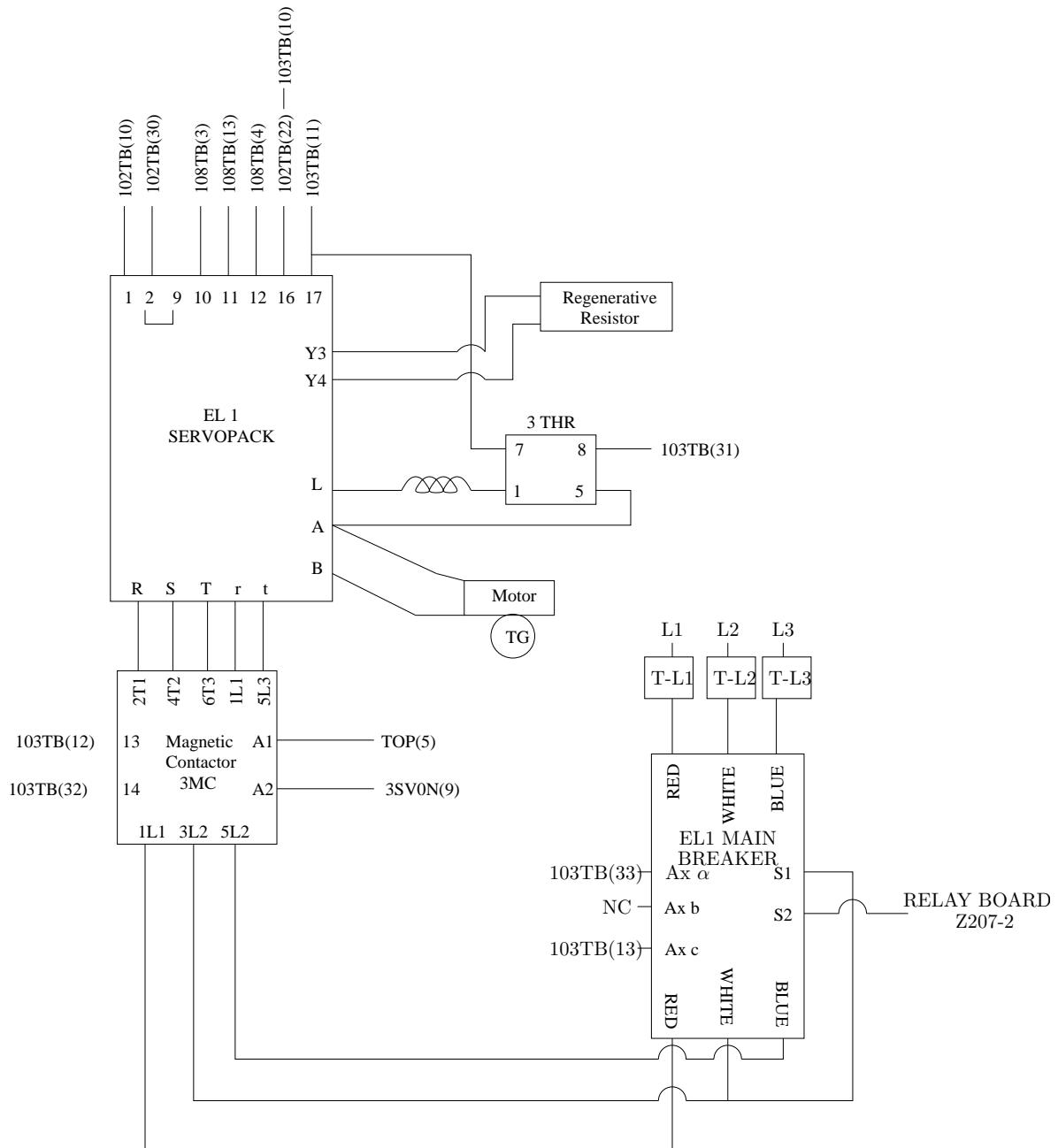


Figure F.12: Wiring Diagram for the EL1 Servopack

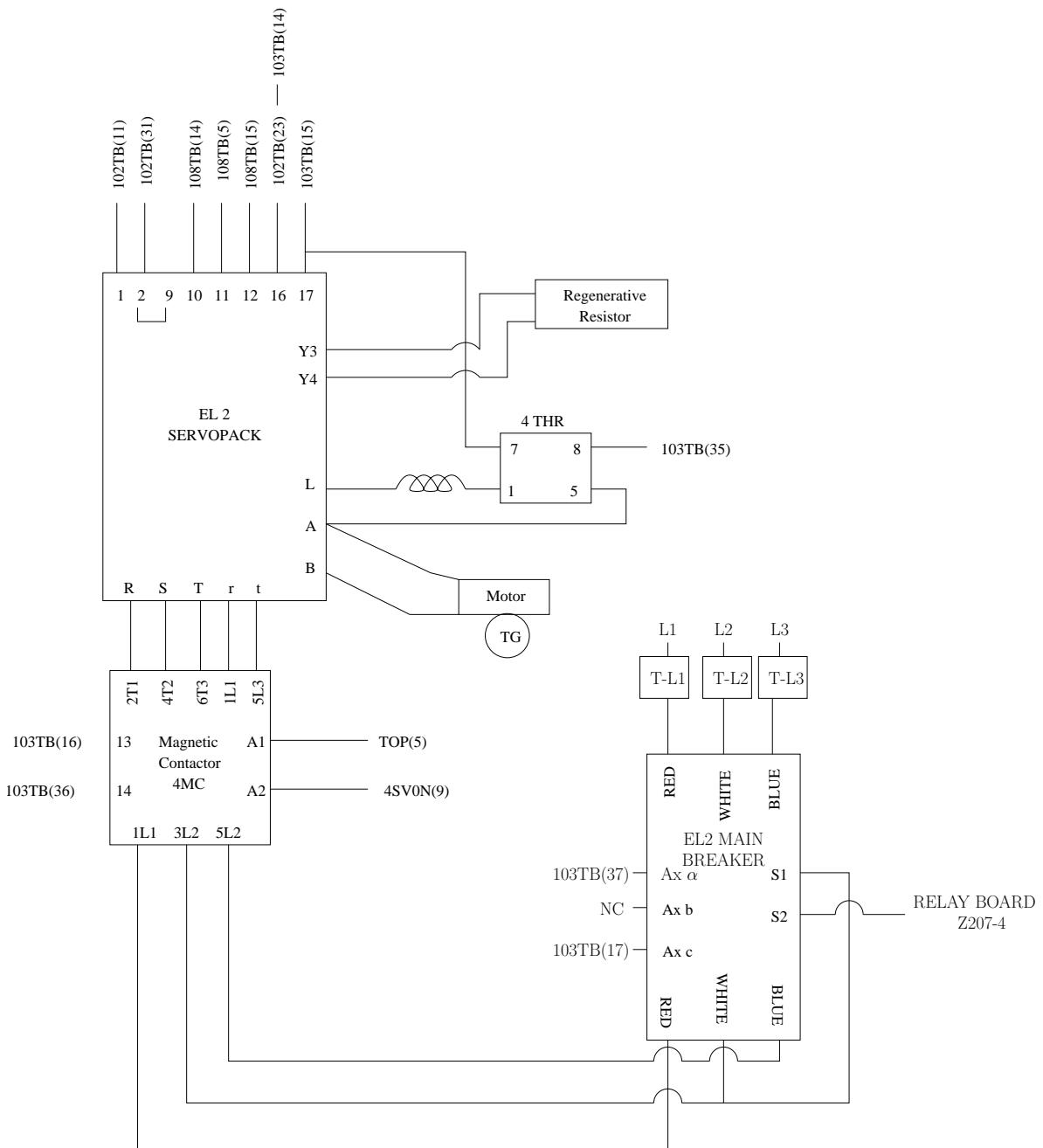


Figure F.13: Wiring Diagram for the EL2 Servopack

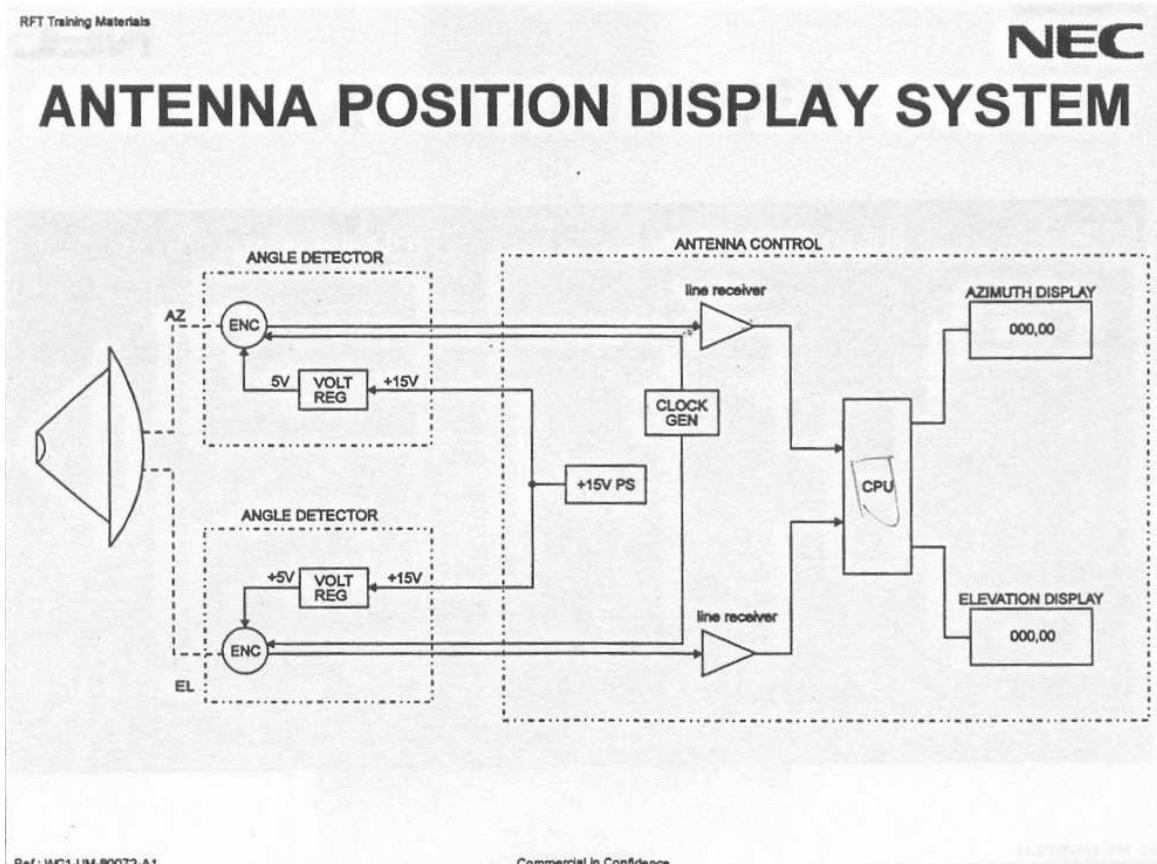


Figure F.14: The existing angle encoder layout (CBA, 1998)

## **Appendix G**

# **Prototype Control Board Screenshots**

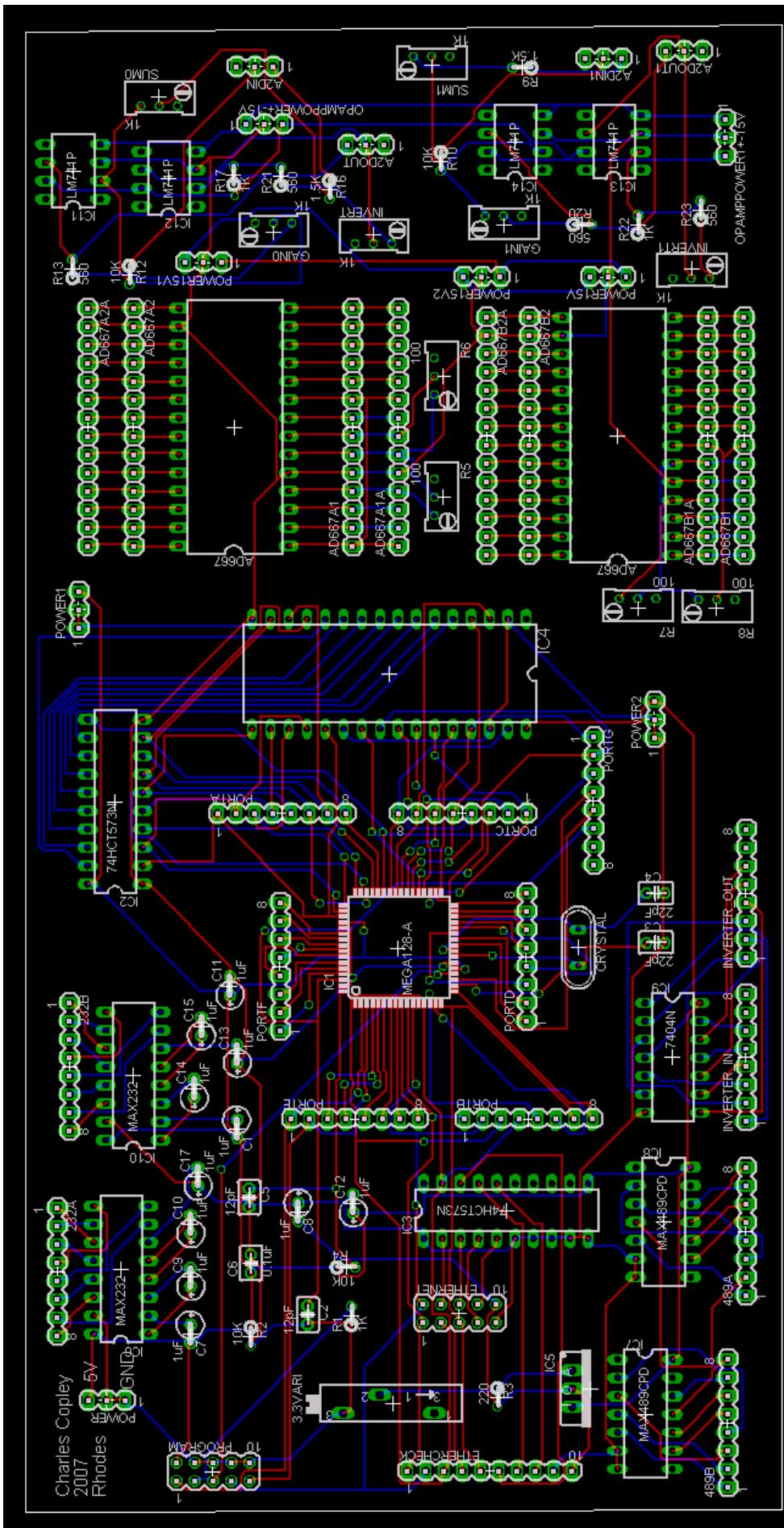


Figure G.1: Both sides of the control board

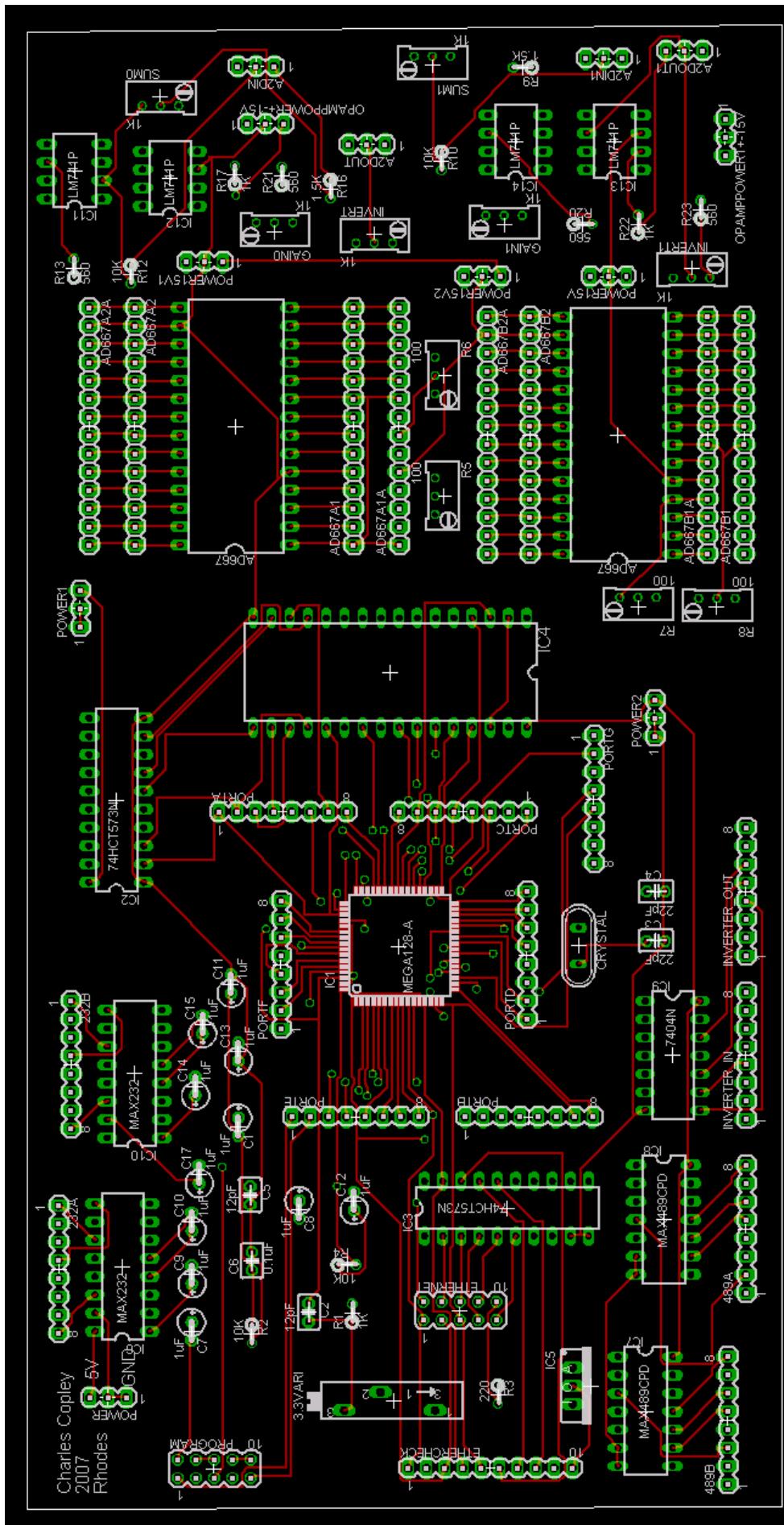


Figure G.2: Top side of the control board

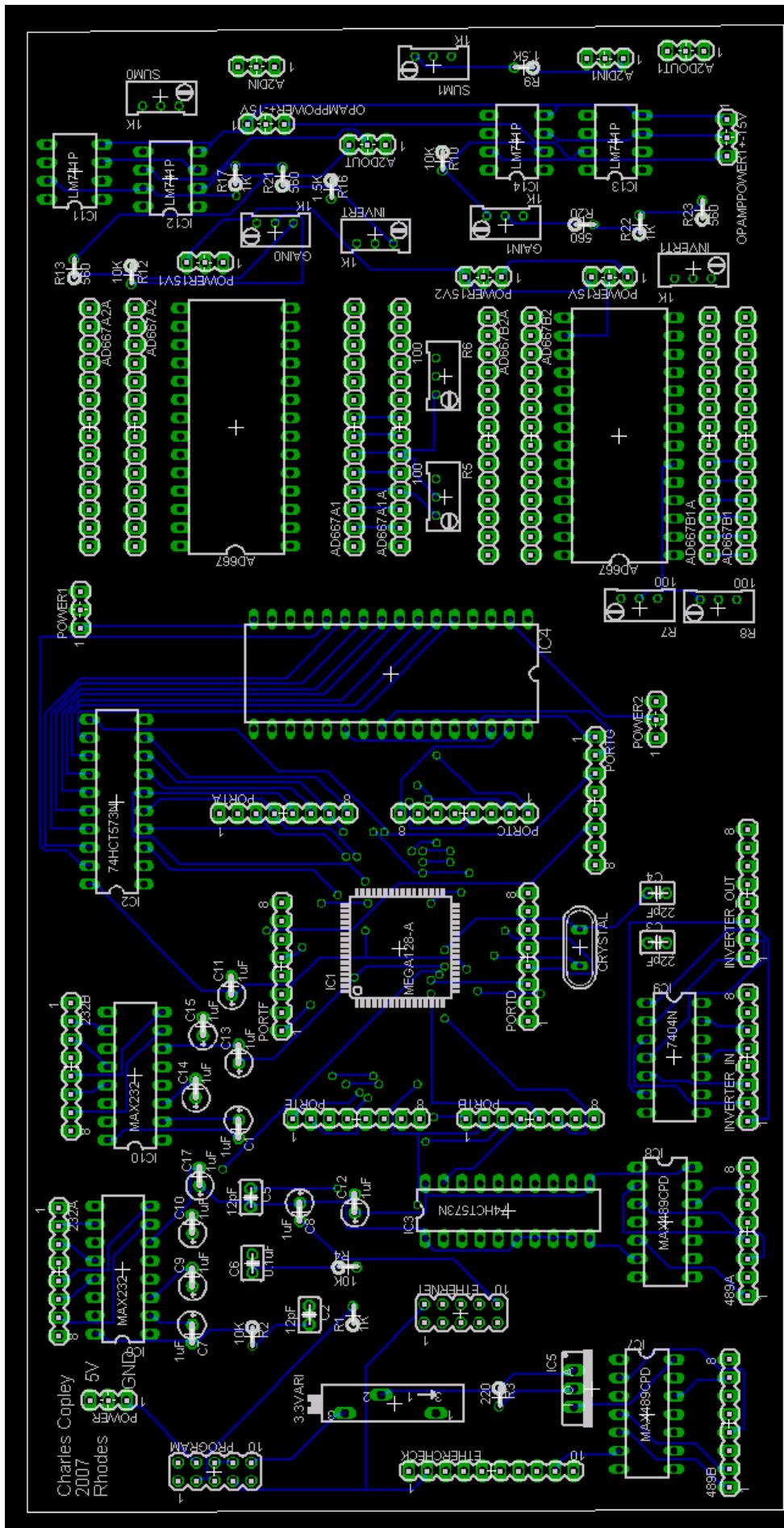


Figure G.3: Bottom side of the control board

## Appendix H

# Required External Jumpers for the Prototype Board

Diagrams showing the required external jumpers for the prototype board are shown below.

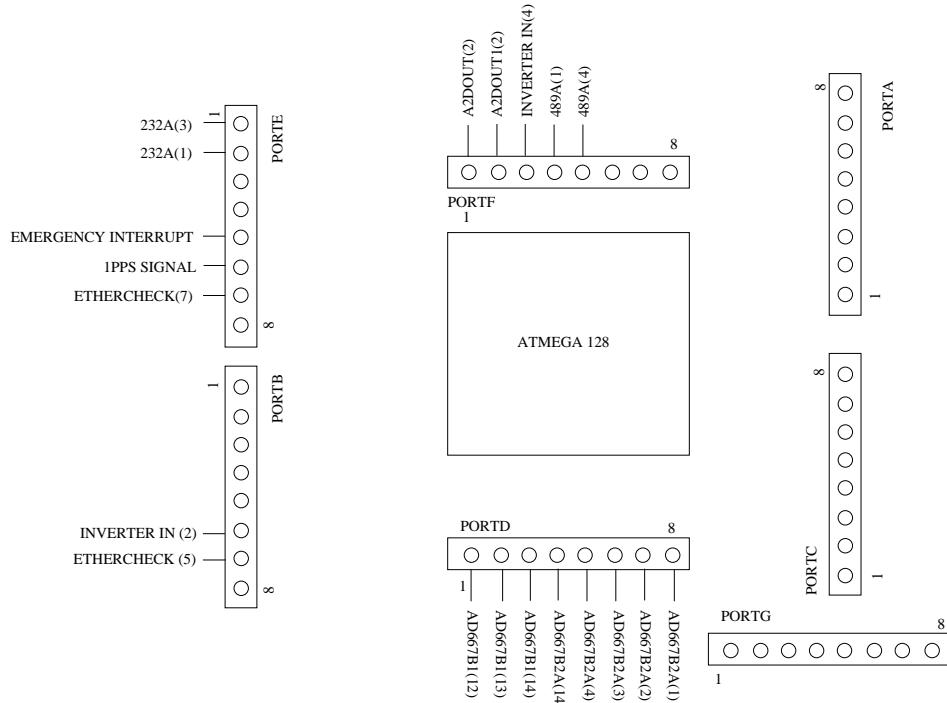


Figure H.1: Description of the external jumpers needed to complete the prototype board.

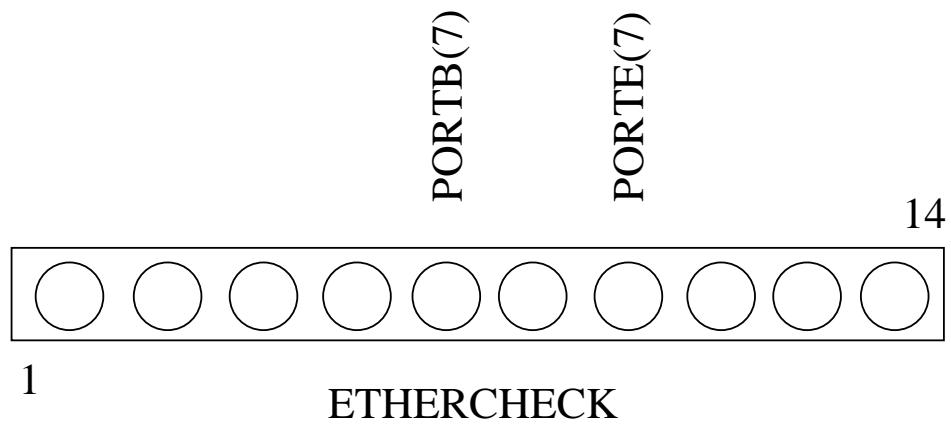


Figure H.2: The external jumpers required to *ETHERCHECK*.

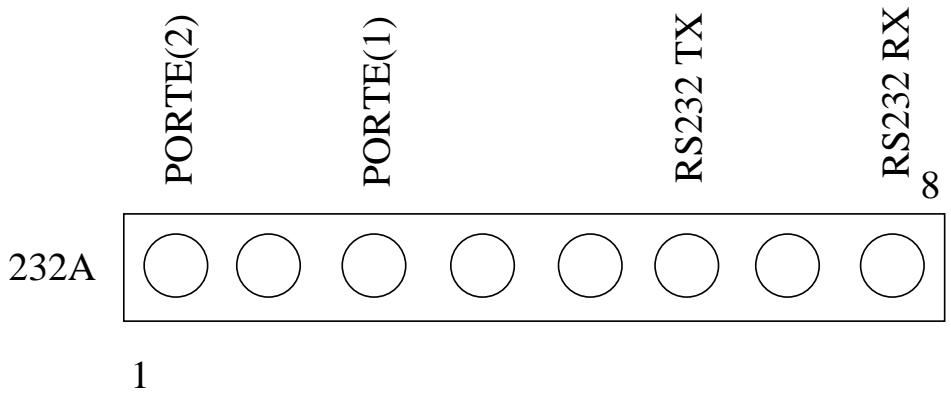


Figure H.3: Jumpers required to *232A*

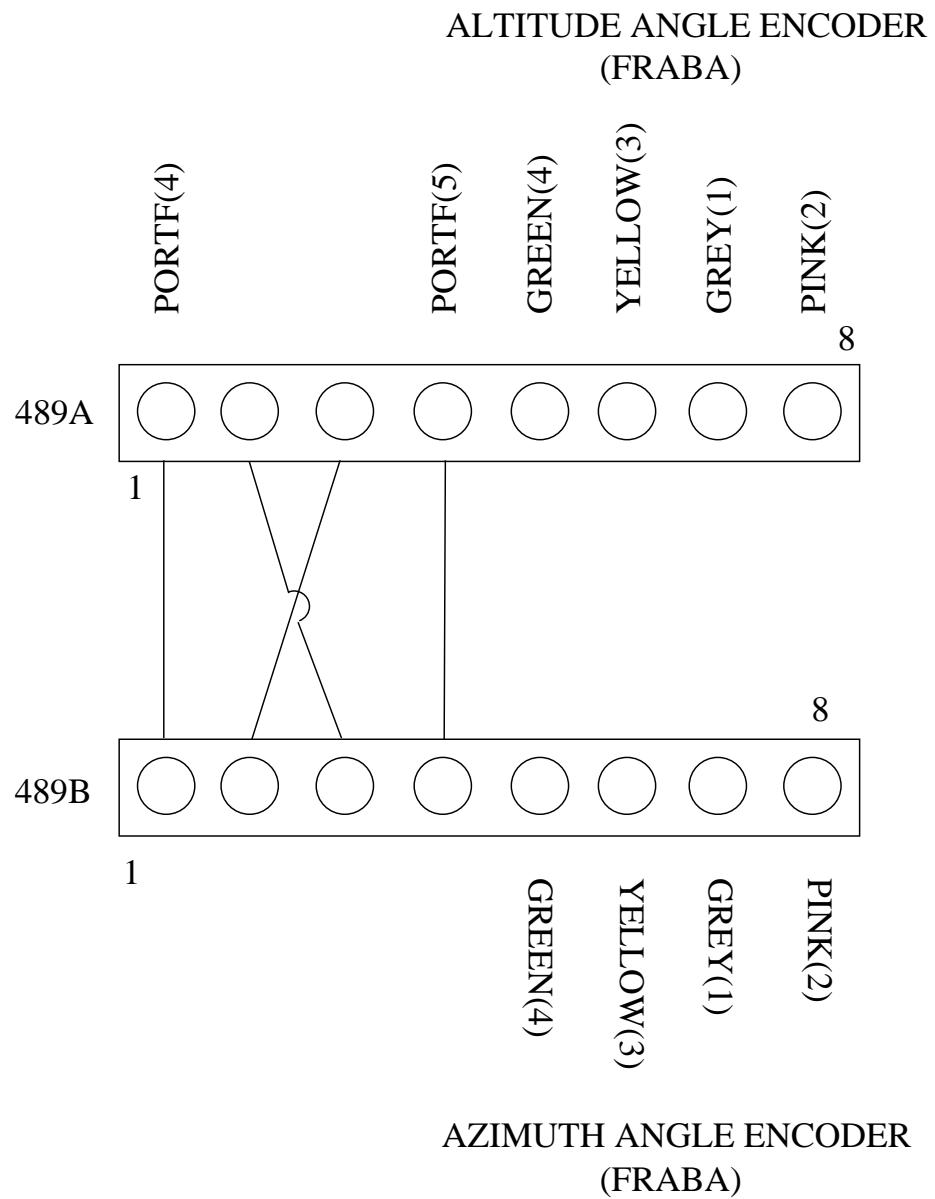


Figure H.4: The external jumpers needed to complete the wiring to pins 489A and 489B.

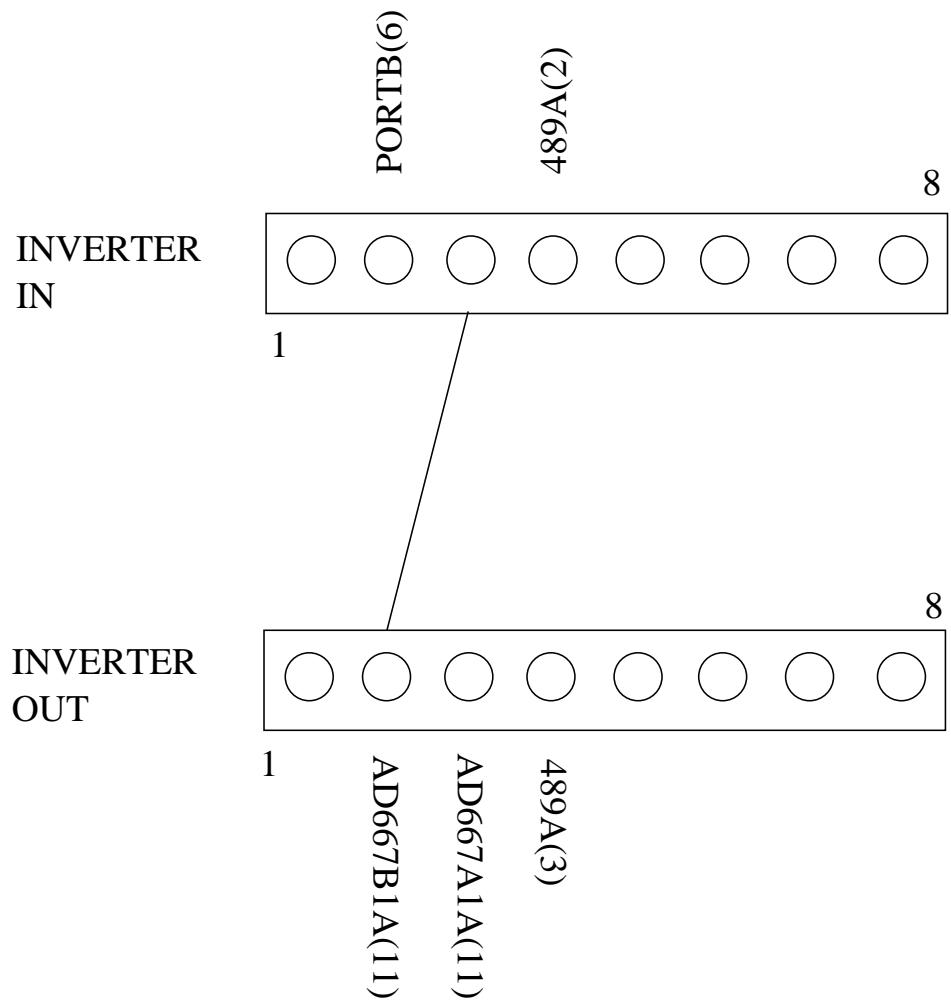


Figure H.5: Jumpers needed to complete the wiring to *INVERTER IN* and *INVERTER OUT*

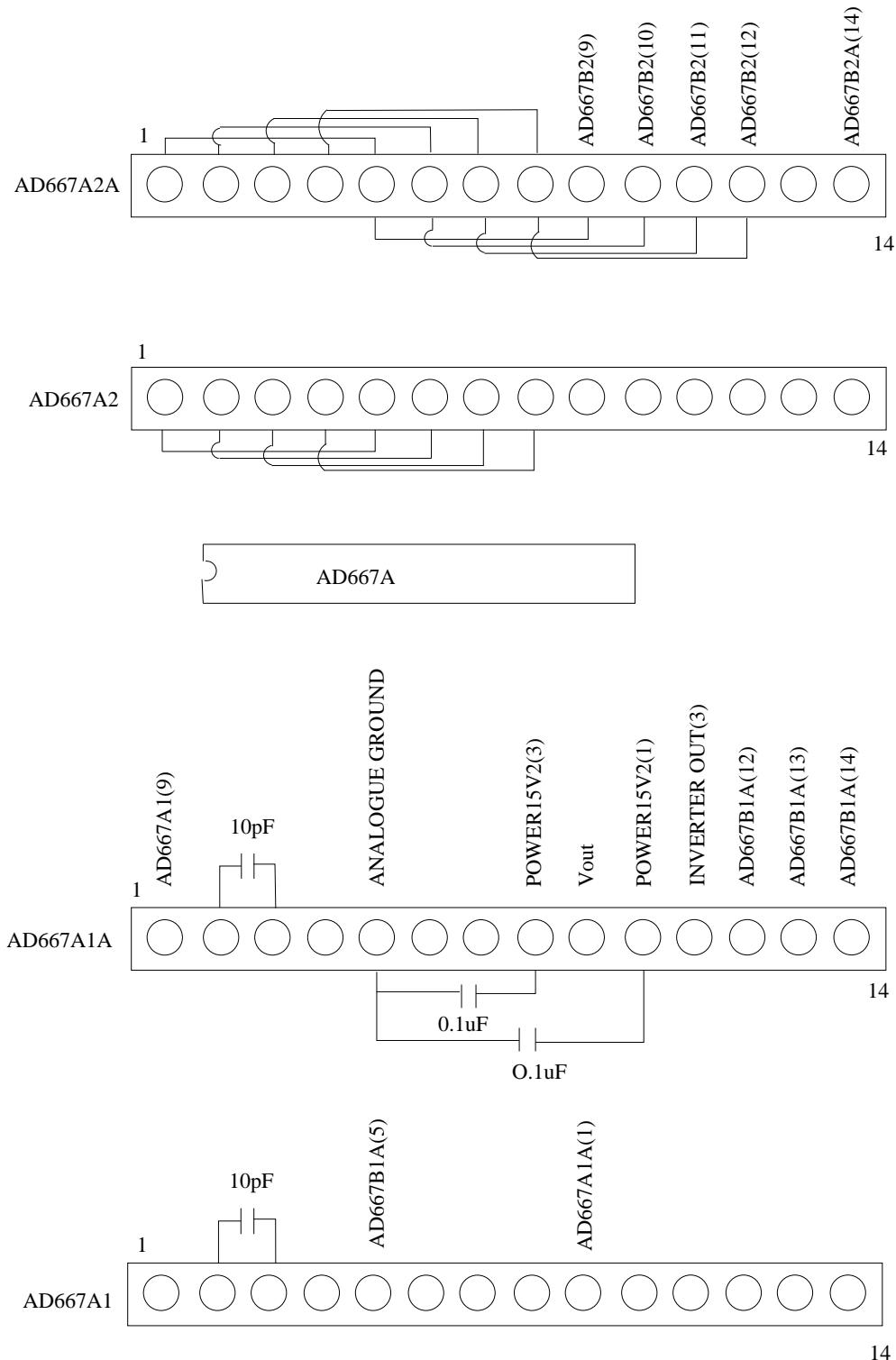


Figure H.6: The external jumpers required for AD667(a)- the altitude DAC

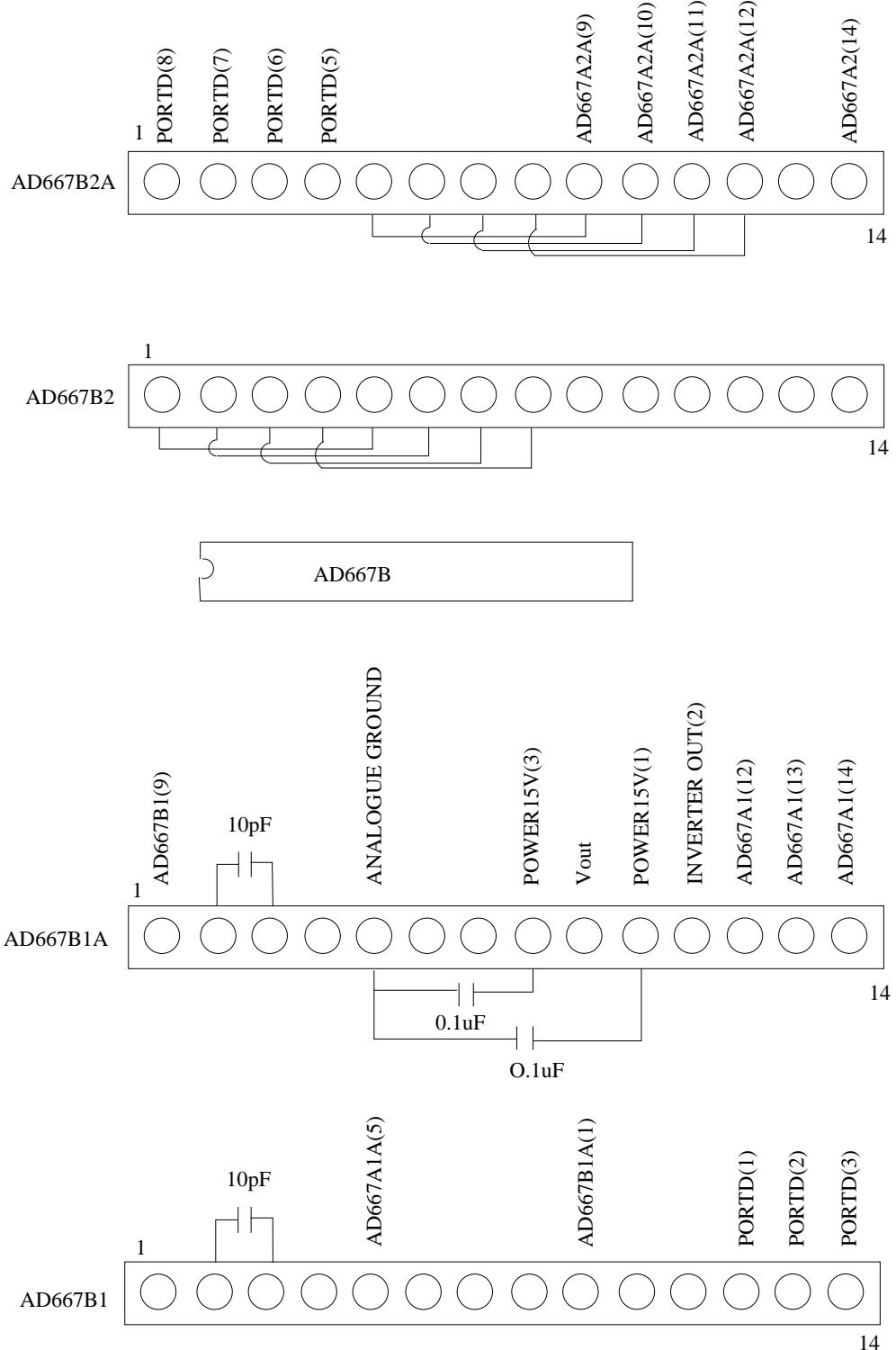


Figure H.7: The external jumpers required for AD667(b)- the azimuth DAC

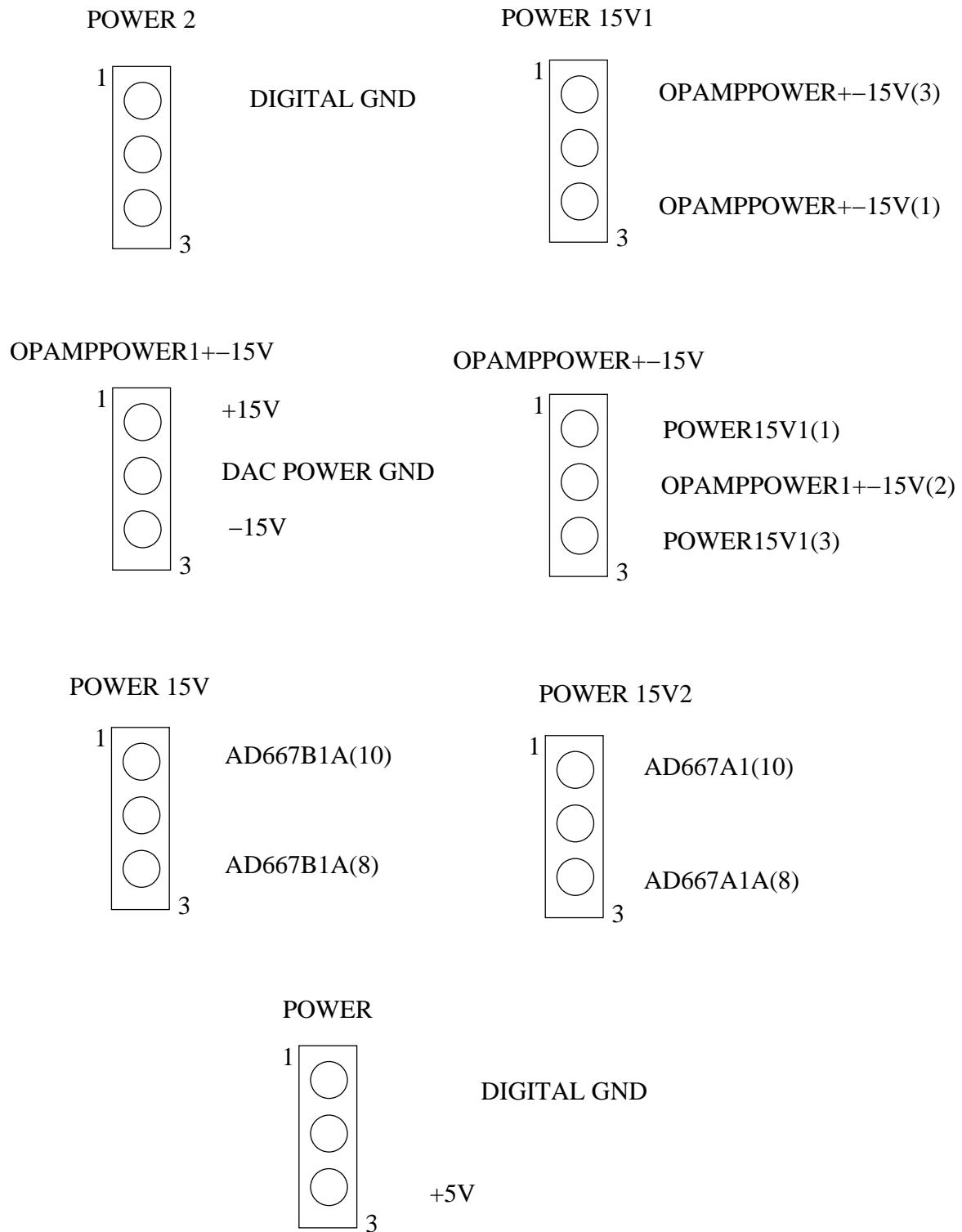


Figure H.8: External jumpers and power connections required.

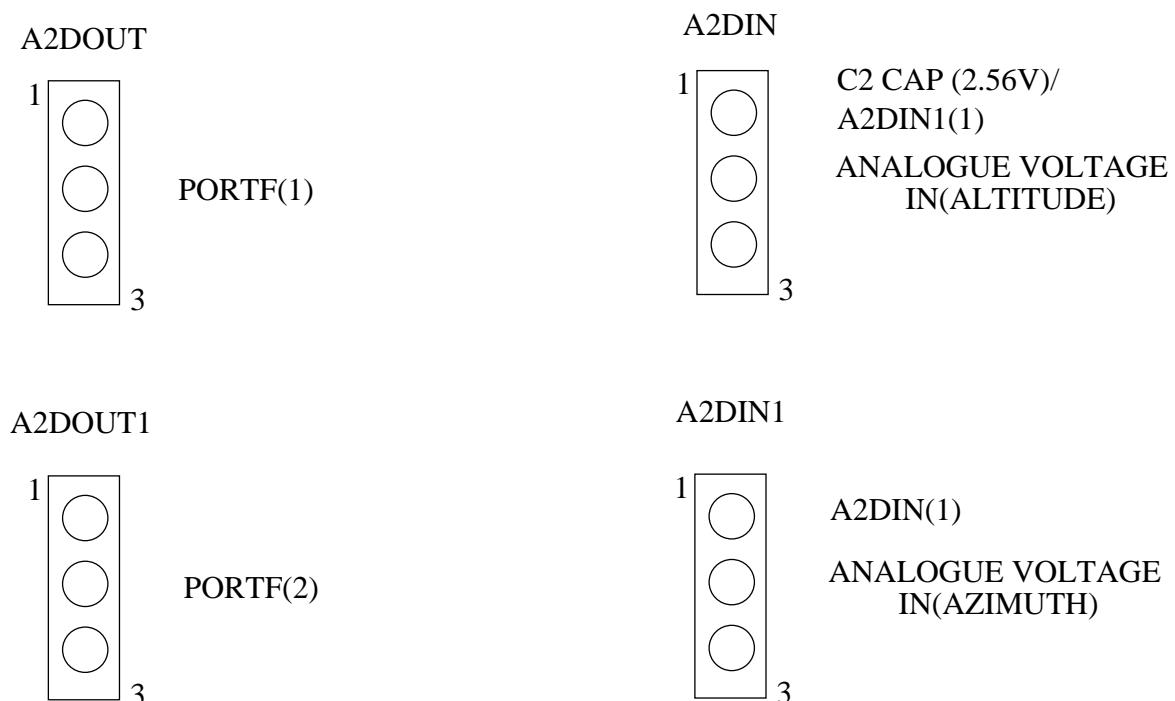


Figure H.9: The external jumpers and voltages required for the A/D analogue circuitry.

## Appendix I

# Additional Prototype Board Information

### I.1 ENC28J60 SPI Commands

Instruction Name and Mnemonic	Byte 0						Byte 1 and Following							
	Opcode			Argument			Data							
Read Control Register (RCR)	0	0	0	a	a	a	a	a	a	a	a	a	a	N/A
Read Buffer Memory (RBM)	0	0	1	1	1	0	1	0						N/A
Write Control Register (WCR)	0	1	0	a	a	a	a	a	a	d	d	d	d	d
Write Buffer Memory (WBM)	0	1	1	1	1	0	1	0		d	d	d	d	d
Bit Field Set (BFS)	1	0	0	a	a	a	a	a	a	d	d	d	d	d
Bit Field Clear (BFC)	1	0	1	a	a	a	a	a	a	d	d	d	d	d
System Command (Soft Reset) (SC)	1	1	1	1	1	1	1	1	1					N/A

Legend: a = control register address, d = data payload.

Figure I.1: Available SPI commands for the ENC28J60 Ethernet Controller (ENC, 2004)

### I.2 ENC28J60 Initialisation

```
from enc28j60.c with headers in enc28j60.h
.....
void enc28j60Init(void)
{
    // initialise I/O
    sbi(ENC28J60_CONTROL_DDR, ENC28J60_CONTROL_CS);
    sbi(ENC28J60_CONTROL_PORT, ENC28J60_CONTROL_CS);
    // enable the reset pin and drive it high
    sbi(ENC28J60_CONTROL_DDR, ENC28J60_RESET);
    // setup SPI I/O pins
    sbi(ENC28J60_CONTROL_PORT, ENC28J60_RESET);
    // set SCK hi
```

```

    sbi(ENC28J60_SPI_PORT, ENC28J60_SPI_SCK);
    // set SCK as output
    sbi(ENC28J60_SPI_DDR, ENC28J60_SPI_SCK);
    // set MISO as input
    cbi(ENC28J60_SPI_DDR, ENC28J60_SPI_MISO);
    // set MOSI as output
    sbi(ENC28J60_SPI_DDR, ENC28J60_SPI_MOSI);
    // SS must be output for Master mode to work
    sbi(ENC28J60_SPI_DDR, ENC28J60_SPI_SS);

    // initialise SPI interface
    // master mode
    sbi(SPCR, MSTR);
    // select clock phase positive-going in middle of data
    cbi(SPCR, CPOL);
    // Data order MSB first
    cbi(SPCR, DORD);
    // switch to f/4 2X = f/2 bitrate
    cbi(SPCR, SPR0);
    cbi(SPCR, SPR1);
    sbi(SPSR, SPI2X);
    // enable SPI
    sbi(SPCR, SPE);

    // perform system reset
    enc28j60WriteOp(ENC28J60_SOFT_RESET, 0, ENC28J60_SOFT_RESET);

    delay_us(400);
    // check CLKRDY bit to see if reset is complete
    while(!(enc28j60Read(ESTAT) & ESTAT_CLKRDY)){

};

    // do bank 0 stuff
    // initialise receive buffer
    // 16-bit transfers, must write low byte first
    // set receive buffer start address
    NextPacketPtr = RXSTART_INIT;
    enc28j60Write(ERXSTL, RXSTART_INIT&0xFF);
    enc28j60Write(ERXSTH, RXSTART_INIT>>8);
    // set receive pointer address
    enc28j60Write(ERXRDPTL, RXSTART_INIT&0xFF);
    enc28j60Write(ERXRDPTH, RXSTART_INIT>>8);
    // set receive buffer end
    // ERXND defaults to 0x1FFF (end of ram)
    enc28j60Write(ERXNDL, RXSTOP_INIT&0xFF);
    enc28j60Write(ERXNDH, RXSTOP_INIT>>8);
    // set transmit buffer start
    // ETXST defaults to 0x0000 (beginning of ram)
    enc28j60Write(ETXSTL, TXSTART_INIT&0xFF);
    enc28j60Write(ETXSTH, TXSTART_INIT>>8);

    // do bank 2 stuff
    // enable MAC receive

```

```

enc28j60Write(MACON1, MACON1_MARXEN|MACON1_TXPAUS|\
MACON1_RXPAUS);
// enc28j60Write(MACON1, MACON1_MARXEN);

// bring MAC out of reset
enc28j60Write(MACON2, 0x00);
// enable automatic padding and CRC operations
enc28j60WriteOp(ENC28J60_BIT_FIELD_SET, MACON3, \
MACON3_PADCFG0|MACON3_TXCRCEN|MACON3_FRMLNEN);
// enc28j60Write(MACON3, MACON3_PADCFG0|MACON3_TXCRCEN|\
MACON3_FRMLNEN);
// set inter-frame gap (non-back-to-back)
enc28j60Write(MAIPGL, 0x12);
enc28j60Write(MAIPGH, 0x0C);
// set inter-frame gap (back-to-back)
enc28j60Write(MABBIPG, 0x12);
// Set the maximum packet size which the controller will \
accept
enc28j60Write(MAMXFLL, MAX_FRAMELEN&0xFF);
enc28j60Write(MAMXFLH, MAX_FRAMELEN>>8);

// do bank 3 stuff
// write MAC address
// NOTE: MAC address in ENC28J60 is byte-backward
enc28j60Write(MAADR5, ENC28J60_MAC0);
enc28j60Write(MAADR4, ENC28J60_MAC1);
enc28j60Write(MAADR3, ENC28J60_MAC2);
enc28j60Write(MAADR2, ENC28J60_MAC3);
enc28j60Write(MAADR1, ENC28J60_MAC4);
enc28j60Write(MAADR0, ENC28J60_MAC5);

// no loopback of transmitted frames
enc28j60PhyWrite(PHCON2, PHCON2_HDLDIS);

// switch to bank 0
enc28j60SetBank(ECON1);
// enable interrupts
enc28j60WriteOp(ENC28J60_BIT_FIELD_SET, EIE, EIE_INTIE|\
EIE_PKTIE|EIE_TXIE|EIE_TXERIE|EIE_RXERIE);
// enc28j60WriteOp(ENC28J60_BIT_FIELD_SET, EIE, EIE_INTIE|\
EIE_PKTIE|EIE_TXERIE|EIE_RXERIE);
// enable packet reception
enc28j60WriteOp(ENC28J60_BIT_FIELD_SET, ECON1, ECON1_RXEN);

// enc28j60PhyWrite(PHLCON, 0x0AA2);

// setup duplex -----
// Disable receive logic and abort any packets currently being \
transmitted
enc28j60WriteOp(ENC28J60_BIT_FIELD_CLR, ECON1, ECON1_TXRTS|\
ECON1_RXEN);

// {
    u16 temp;
    // Set the PHY to the proper duplex mode
    temp = enc28j60PhyRead(PHCON1);
    temp &= ~PHCON1_PDPXMD;
}

```

```

    enc28j60PhyWrite(PHCON1, temp);
    // Set the MAC to the proper duplex mode
    temp = enc28j60Read(MACON3);
    temp &= ~MACON3_FULDPX;
    enc28j60Write(MACON3, temp);

// }

// Set the back-to-back inter-packet gap time to IEEE \
specified
// requirements. The meaning of the MABBIPG value changes \
with the duplex
// state, so it must be updated in this function.
// In full duplex, 0x15 represents 9.6us; 0x12 is 9.6us in \
half duplex
enc28j60Write(MABBIPG, 0x15);

// setup the ENC28J60 filters 0xe0 will set the controller up \
to only work with the host PC i.e no ARP broadcast packets \
will be accepted! 0xa1 accepts all packets with a matching \
MAC address, correct CRC, or Broadcast packets - 0xa1 seems to \
have more issues with Ethernet interrupts - it is better to \
run the system in a closed environment to maintain as few \
errors as possible
enc28j60Write(ERXFCON, 0xe0);
// Reenable receive logic
enc28j60WriteOp(ENC28J60_BIT_FIELD_SET, ECON1, ECON1_RXEN);

```

Listing I.1: Initialisation code for the ENC28J60 Ethernet Controller

### I.3 ATMega128 Fuse Settings

The ATMega128 has 3 fuse bytes which control clock settings, EEPROM characteristics, programming, watchdog timers, start-up times, brown-out detection and the ATMega103 compatibility mode. These settings are detailed in (ATMEL). The fuse settings used are included in Table I.1.

Table I.1: Fuse settings of the ATMega 128

Fuse	Setting	Description
lfuse	0xef	Brown-out detection disabled 4.1 ms startup time Crystal Oscillator >8 MHz
hfuse	0xc1	Disable OCD No JTAG Programming >8 MHz Clock Bootsize =4096 words
efuse	0xee	Watchdog timer always on ATMega103 Compatibility mode disabled

### I.4 A/D Analogue Circuitry

In the  $-12V \rightarrow +12V$  case we used the resistors shown in Table I.2. These set an output of between  $0V \rightarrow 2.56V$ .

In order to set the variable resistors correctly the following process should be followed:

1. Ground  $V_{in}$ . Adjust SUM0 and SUM1 until A2DOUT Pin3 is  $-1.28$  V (i.e  $2.56$  V/2)

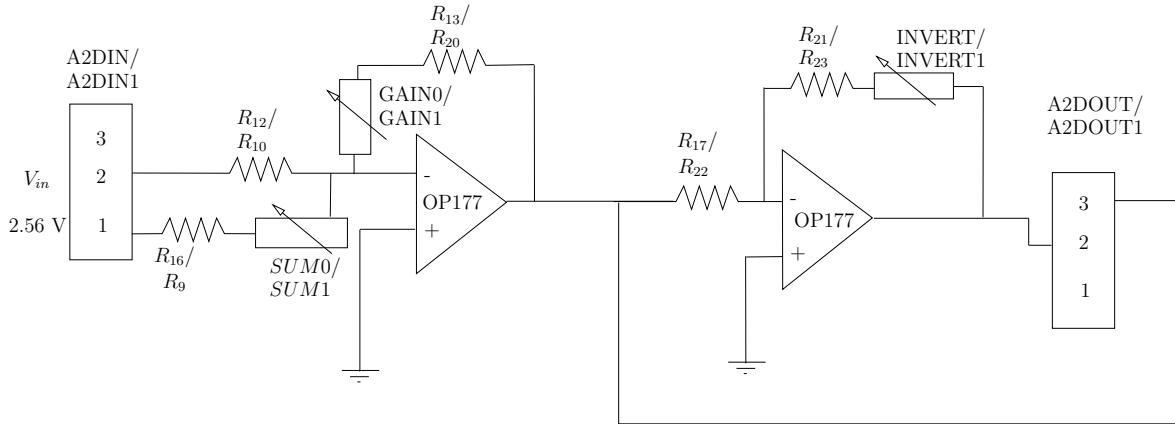


Figure I.2: A schematic of the analogue input circuitry to the A/D.

Table I.2: Analogue input circuitry resistor values used

Resistor	Value
$R_{16} \text{ & } R_9$	1.5 K
$R_{12} \text{ & } R_{10}$	10 K
$R_{13} \text{ & } R_{20}$	560
$R_{17} \text{ & } R_{22}$	1 K
$R_{21} \text{ & } R_{23}$	560

2. Set  $V_{in}$  to high voltage i.e +12V. Adjust GAIN0 and GAIN1 until A2DOUT Pin3 is at -2.56 V.
3. Now adjust INVERT and INVERT1 until A2DOUT Pin2 is at +2.56 V.
4. Check that the low voltage i.e  $V_{in} = -12$  V gives 0 V output on A2DOUT Pin2 and Pin3

If a different input voltage range is required the resistors can be calculated as follows

1. Determine the minimum input voltage. At this point there should be zero current flowing into the first op-amp inverting input due to the 2.56 V offset. Calculate the current through  $R_{12}$  and set  $R_{16}$  such that they are roughly correct to match this current. SUM0 and SUM1 can be adjusted to fine tune this. e.g if -6 V to +6V input range is required:

- Arbitrarily choose  $R_{12} = 10K$
- $I_{R_{12}} = \frac{-6V}{10K} = 0.6 \text{ mA}$
- As such we need this current to flow through  $R_{16}$  to cause the offset
- So  $\frac{V}{I} = R \rightarrow \frac{2.56}{0.6 \text{ mA}} = 4.27K$
- If we set  $R_{16}$  to the readily available 3.3K resistor then we will could use 2K potentiometers for SUM0 to make up the additional 966 ohms and produce 4.27 K
- .
- 2. We now have between 0 and  $2V_{in}(\max)/R_{12}$  amps going through  $R_{20}$ . We need the resistance of  $R_{13}$  and GAIN0 to produce a voltage between -2.56 V and 0 V.
- Using  $\frac{V}{I} = R$ , we can say that  $\frac{2.56}{2V_{in}(\max)/R_{12}} = R$

- If we use the -6 V to +6 V range and use  $R_{12} = 10K$  as before we arrive at  $R=2.133K$ . As before we set the rough resistance with  $R_{13}$  and then finetune it using GAIN0.
3. Arbitrarily choose  $R_{17}$  and then set  $R_{21}$  to the next lowest available resistor value. INVERT can then be adjusted to provide unity gain through inverting amplifier op-amp circuit.  $R_{21}$  and INVERT should have the same resistance as  $R_{17}$ . This system allows for the potentially large resistor uncertainties to be removed.

## I.5 Commands Available to the Control Board

Extract from process\_packet.h

```
#define COMMAND 0x1f
#define COMMAND_REPLY 0xf1
#define UDP_PACKET 0xe1
#define ROUGH_TIME_SYNC 0xe0
#define NTP_REQUEST 0xe2
#define RESET 0x08
#define NTP_STATUS 0xe3
#define BEGIN_DATA_DOWNLOAD 0xe4
#define STOP_DATA_DOWNLOAD 0xe5
#define COMMAND_ANGLE 0xe6
#define PID 0xe7
#define ACK 0xe8
#define PID_DUMP 0xe9
#define D2A_SET 0xea
#define D2A_zero 0xeb
#define D2A_high 0xec
#define A2D_sample 0xef
#define A2D_calibrate 0xd0
#define Angle_update_style 0xd1
#define EMERGENCY 0xd2
#define altPID_lookup_table_update 0xd3
#define azPID_lookup_table_update 0xd4
```

Listing I.2: Definitions of the commands available to the control board

### I.5.1 Summary of Commands

#### **ROUGH\_TIME\_SYNC**

This command tells the microcontroller to process the data contained in the packet and update its clock time to that contained in the packet. This is designed to set the microcontroller clock to roughly the same time as the PC before using the SNTP protocol to refine this time further.

#### **NTP\_REQUEST**

This is a command sent from the PC to the microcontroller which instructs the microcontroller to immediately send an SNTP time synchronisation request.

#### **RESET**

This command caused the microcontroller to reset and reboot.

**NTP\_STATUS**

This was originally used for checking the SNTP protocol while it was being developed. Presently this is not being used.

**BEGIN\_DATA\_DOWNLOAD**

This starts the 1/100 s timer and begins transmitting command voltages through the DAC. Packets of control information will begin streaming to the PC at a rate of 5 packets per second. These packets contain information recorded every 1/100 s.

**STOP\_DATA\_DOWNLOAD**

This stops the 1/100 s timer and sets the command voltages to zero.

**COMMAND\_ANGLE**

This identifies a packet as containing command angle information. A packet containing this information should arrive once every second. There are two options as to how the microcontroller will process these packets. These were detailed in Section 10.2.3.

**PID**

This identifies a packet containing new PID constant values and loads the PID values into SRAM. This sets the microcontroller so that onboard adaptive PID is not used.

**ACK**

This is a tag used by the microcontroller to confirm receipt of a packet sent by the PC or vice-versa.

**PID\_DUMP**

This will transmit the PID constants stored on EEPROM to the PC. Sets the microcontroller so that onboard adaptive PID is not used. It is still possible to perform adaptive control from the PC side if this is desired.

**D2A\_SET**

This allows one to manually set the DAC to a certain voltage. This can be used when checking the DAC configuration and adjusting trim resistors to output the correct voltages.

**D2A\_zero**

This will manually zero the DAC and can be used to adjust the trim resistors to the zero level. This command will not set the DACs to output zero volts, but rather the minimum -10 V.

**D2A\_high**

This will set the DAC to its maximum output voltage +9.976 V, and can be used for adjusting the trim resistors.

### **A2D\_sample**

Here a sample of the two A/D inputs is taken and the reading is transmitted to the PC via Ethernet. The number is transmitted as a 2-s complement 16 bit number i.e  $-32768 \rightarrow +32767$ . This can be used together with the *A2D\_calibrate* command to remove the effects of op-amp offset voltage drift over time.

### **A2D\_calibrate**

This allows one to calibrate any known offsets out of the A/D sampling by using a fitted straight line. A series of known voltages (across the expected range) can be applied to the A/D and the output recorded. From this a gradient and an offset can be calculated by fitting a straight line. This can be loaded into the microcontroller and a correction will be applied to every sample before it is used.

A/D readings are transmitted as 16 bit numbers when the *A2D\_sample* command is used. A straight line can be constructed between the 16 bit number (i.e  $-32768 \rightarrow 32767$ ) and the known application voltage ( written as a 16 bit number). The straight line that needs to be fitted is of the form

$$Voltage_{16\ bit} = m \times A/D_{16\ bit} + Offset \quad (I.1)$$

This allows quick software compensation of offset voltages and partial elimination of grounding difficulties between devices. It was included as a means of dynamically adjusting the A/D inputs. This was necessary since the temperature dependence of the A/D input operational amplifier circuitry was not fully understood. Despite being low offset drift op-amps, the offset voltage will change over time. This allows for simple and quick adjustments to take this into account without any hardware changes.

### **Angle\_update\_style**

This command is used to change the type of data packets that the microcontroller is expecting for the command updates. The option 1 sets the microcontroller to expect a start angle and an end angle and to perform the necessary interpolation itself to achieve the 1/100s command angles. Option 2 set the microcontroller to expect a packet containing the interpolated values which were calculated by the PC. These are detailed more thoroughly in Section 10.2.3.

### **EMERGENCY**

This is a tag used by the microcontroller. When the Emergency Stop Interrupt is triggered on INT4, the microcontroller will transmit a packet to the PC containing this tag.

### **altPID\_lookup\_table\_update**

This expects a packet from the PC containing a set of PID coefficients and a command value to which these coefficients correspond. The PID coefficients are then saved into SRAM, together with the appropriate command value. This creates a lookup table which is used for onboard adaptive PID control. This command sets the microcontroller to use onboard adaptive PID control for the altitude commands.

### **azPID\_lookup\_table\_update**

Similar to *altPID\_lookup\_table\_update* except that it applies to the azimuth lookup table. This will set the microcontroller to use onboard adaptive PID control for the azimuth commands.

## I.6 Compiling the code

I have included a Makefile with the code. If AVRGCC is installed correctly this Makefile will perform all the necessary compilations. A series of files will be produced. Those of primary concern are

1. myproject.hex. ATMEGA128 flash file
2. myproject.eep EEPROM load file for the ATMEGA128

The code should be compiled using level 2 code optimisation.

## I.7 Code Layout

### I.7.1 AVRGCC

Compilation of the code depends on the installation of the default set of libraries that come with AVRGCC. I have successfully compiled the code on two separate machines which have AVRGCC installed on them and so portability should not be a problem.

### I.7.2 C source files

#### **main.c**

This file contains the heart of the control system. All interrupts are initially serviced here.

#### **interrupt\_handling.c**

Provides functions that are called when the enabled ATMEGA128 interrupts are triggered. These define actions to be taken when the following occurs:

1. 1 PPS not-present Interrupt (SIG\_OUTPUT\_COMPARE3B)
2. Control Interrupt (SIG\_OUTPUT\_COMPARE3C)
3. Emergency Stop Interrupt (INT4)
4. 1 PPS Interrupt (INT5)
5. ENC28J60 Interrupt (INT6)

#### **enc28j60.c**

This file contains functions used for interfacing to the ENC28J60 Ethernet controller. Packet sending, packet receiving, initialising the ENC28J60 are all controlled inside this function.

#### **d2a.c**

Functions to set the DACs to the required output voltages by uploading 3 separate nibbles to each 12 bit DAC.

#### **delay.c**

Generates delays in the code based on interrupts. It used Counter 1 to generate these delays so Counter 1 should be avoided elsewhere in the programming.

**udp.c**

Packet construction definitions for UDP and ARP packets. Includes the NTP packet construction and checksum calculation algorithms for UDP and IP headers.

**process\_packet.c**

This contains the primary function that filters through received packets and calls appropriate functions to take appropriate action. The appropriate action functions are defined in here.

**boot\_processes.c**

Any functions called during the microcontroller booting process are included in here. The initial ARP request to establish communication to the host PC is defined in this file.

**angle\_encoders.c**

Functions to read the SSI angle encoder interface and to return the 16 bit angle read by the angle encoder.

**uart.c**

A function file from the AVRGCC collection which establishes interrupt driven UART output.

### I.7.3 C header files

**main.h**

Function definitions for `main.c`.

**math.h**

Mathematical function headers to allow various mathematical calculations on the microcontroller.

**interrupt\_handling.h**

Function definitions for `interrupt_handling.c`.

**process\_packet.h**

Function definitions for `process_packet.c`. It contains the command definitions used to tag and filter UDP packets.

**d2a.h**

Function definitions for `d2a.c`. It contains the hardware definitions of the pins used to interface between the ATMEGA128 and the DACs as laid out on the prototype control board.

**port128.h**

Hardware definitions for the ATMEGA128.

**delay.h**

Function definitions for `delay.c`.

**boot\_processes.h**

Function definitions for `boot_process.c`.

**uart.h**

Definitions used in `uart.c`.

**enc28j60.h**

Header file for `enc28j60.c`. Includes definitions of maximum Ethernet packet sizes and the ENC28J60 receive and transmit buffers

**enc28j60conf.h**

This includes all the Ethernet and IP configuration data such as MAC Addresses, IP addresses, UDP port settings and ATMEGA128 hardware definitions for interfacing to the ENC28J60. These should all be configured correctly for the system to work. In particular the GIP values define the gateway IP, which in this case is the host PC IP address. ARP requests are originally sent to this IP address and initialisation depends on a response.

**udp.h**

Header file for `udp.c`. includes the packet definition bytes which are used to tag and identify outgoing packets and which are checked when processing incoming packets from the PC.

**angle\_encoders.h**

Header file for `angle_encoders.c`. includes hardware definitions for the ATMEGA128 and the actual pins used for the SSI clock signal, encoder select pin and the receive pin, as well as the ATMEGA128 port that needs to be initialised during the encoder interface initialisation.

**global.h**

Includes the crystal clock speed and the AVRGCC type definitions which may be used with AVRGCC functions. Enables floating point `printf()` support.

**avrlibdefs.h**

Definitions of a few bit-wise operators suitable for use on the ATMEGA128.

**avrlibtypes.h**

AVRGCC type definitions and maximum/minimum sizes of each type.

## Appendix J

# PID Tuning Examples using the prototype control board

### J.1 Step Response with no control

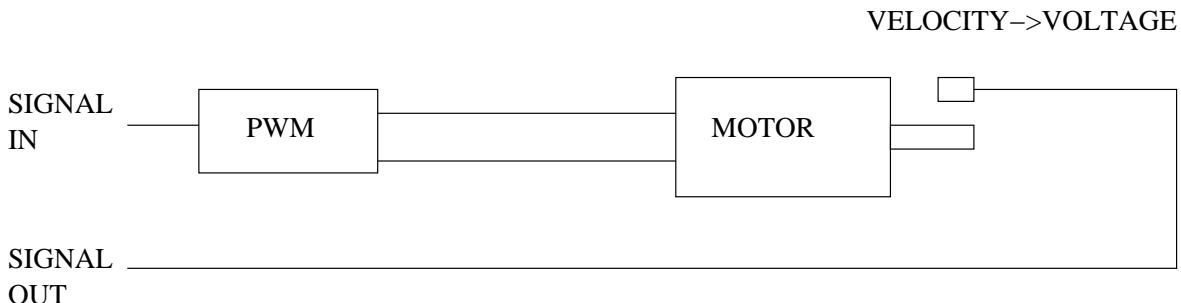
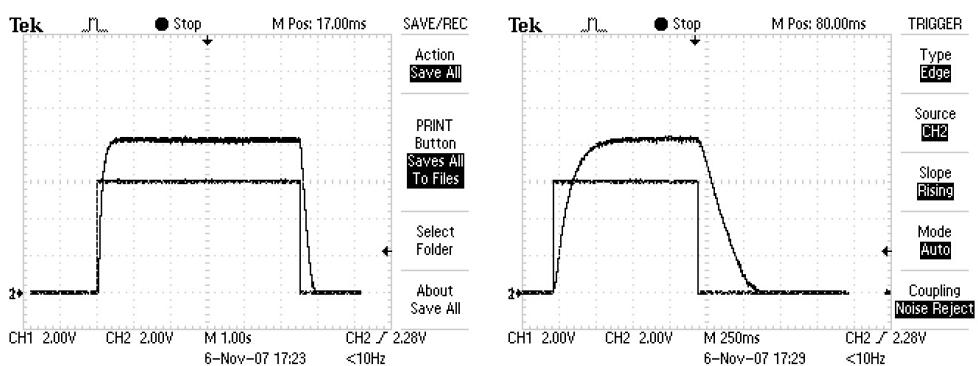


Figure J.1: Layout of the equipment used in the testing the DC motor Step Response.  
A step input from a signal generator was applied at the PWM input to generate the step input.



(a) The motor 6 V step response with no control applied at 1 second timescale

(b) The motor 6 V step response with no control applied at 250 ms timescale

Figure J.2: The motor 6 V step responses

## J.2 Control and PID Tuning

I then rearranged the equipment to include the control board as shown in Figure J.3

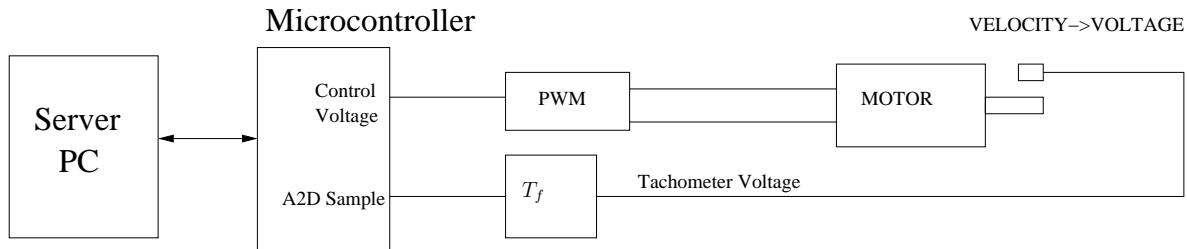


Figure J.3: Layout of the equipment used in the obtaining these results.

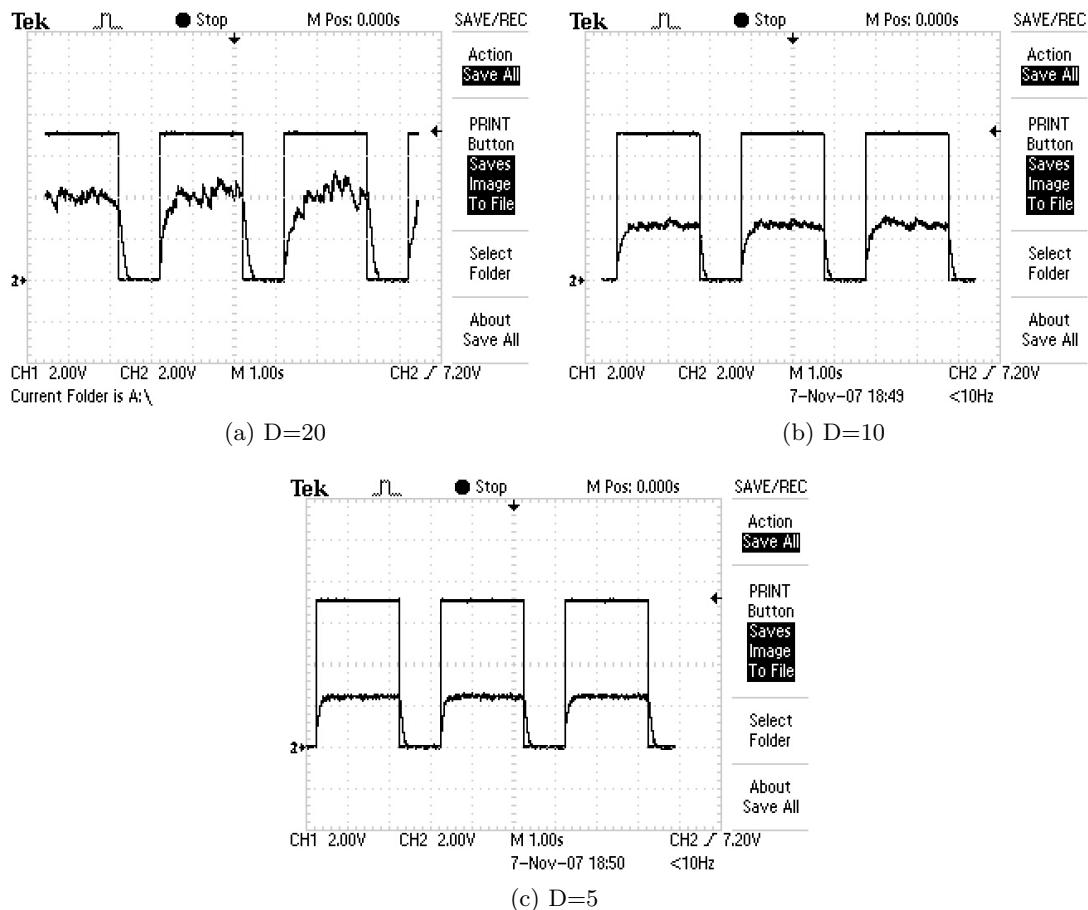
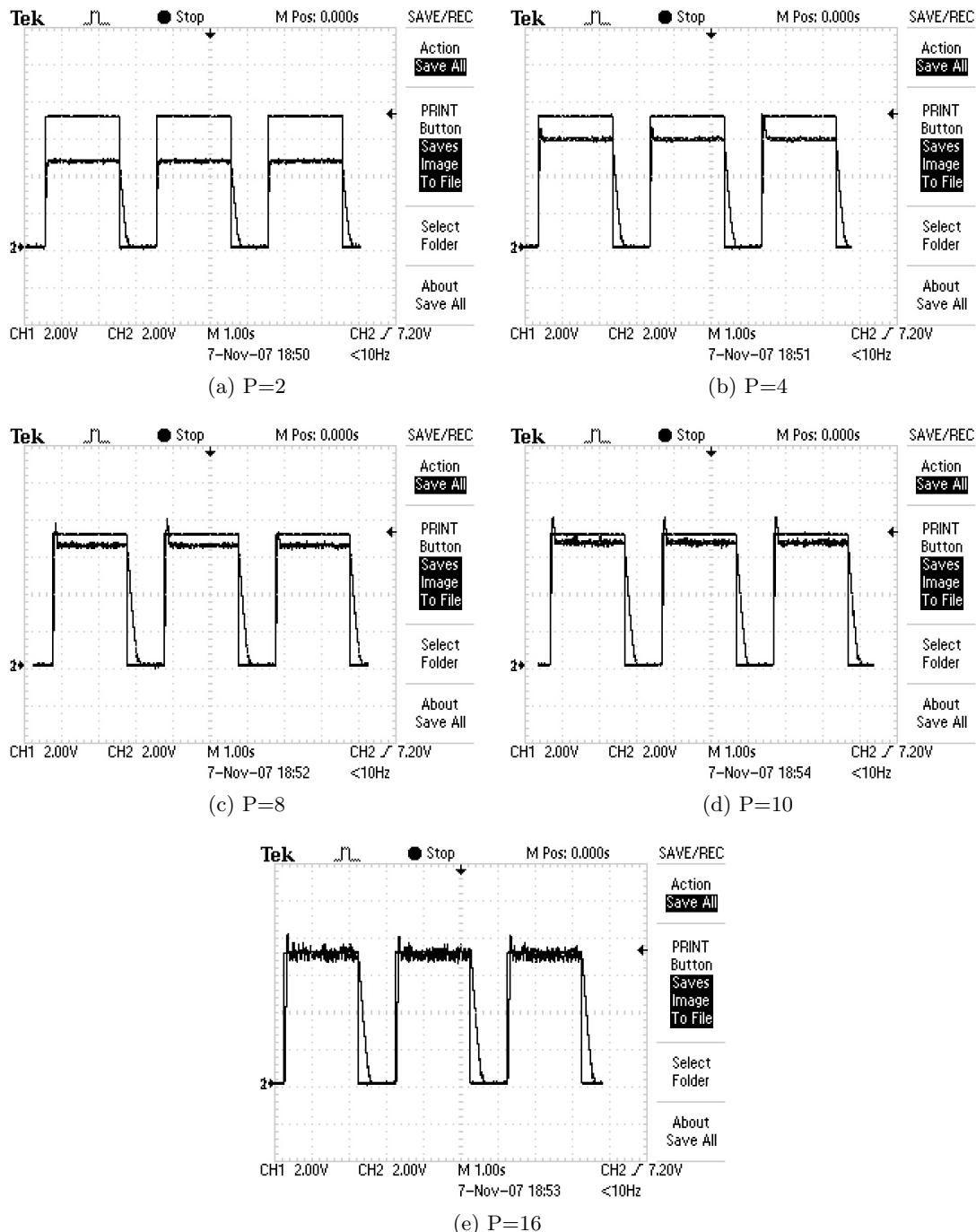


Figure J.4: Tuning the D coefficient as explained in (Westcott, 2000). Settled on D= between 5 and 10

Figure J.5: P tuning-Settled on  $P =$  between 4 and 8

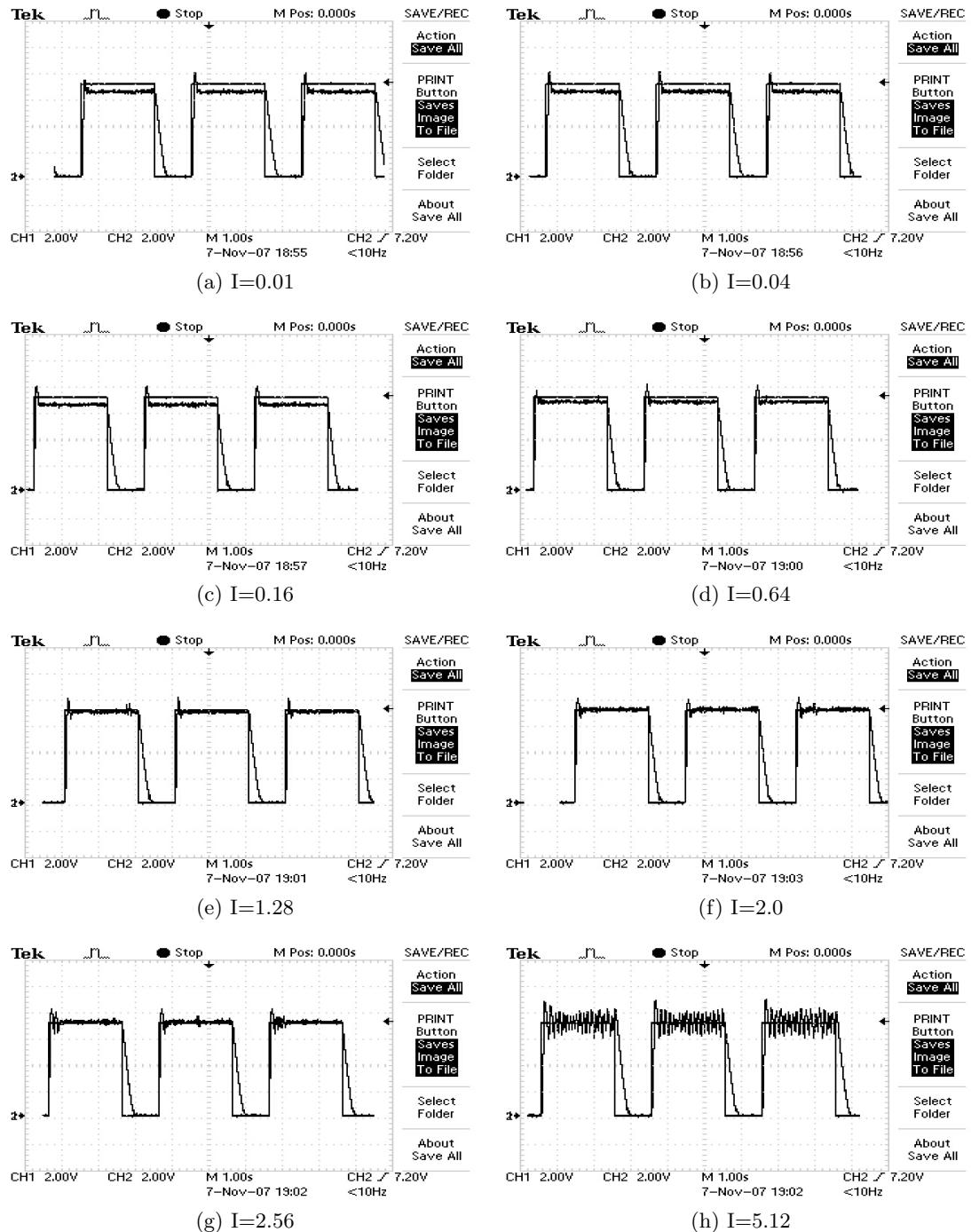


Figure J.6: I tuning-Settled on I = 2

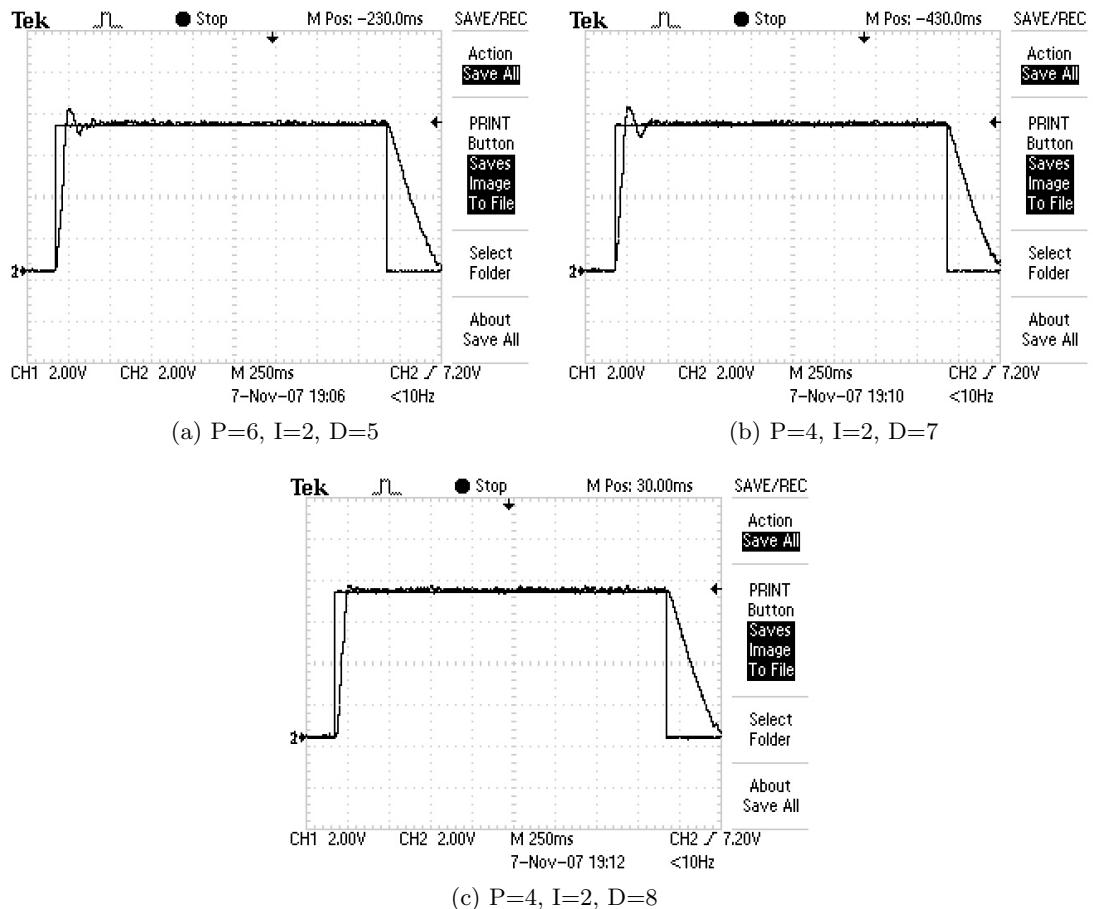


Figure J.7: Final Tuning- Command = 23000- This demonstrates the good response achieved in (c). Notice the difference between (b) and (c) which demonstrates the effect of changing the D coefficient.

### J.3 Adaptive PID Control

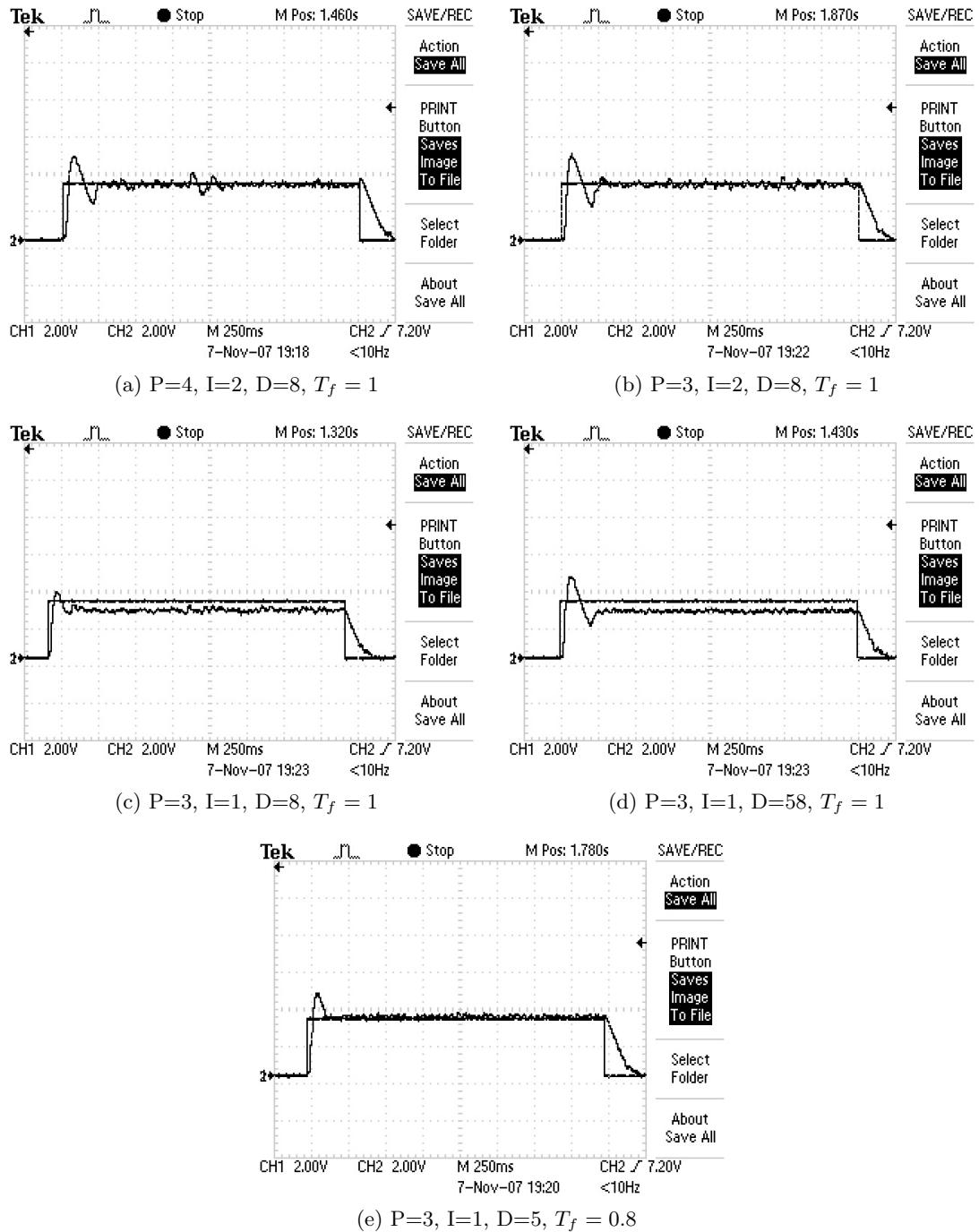


Figure J.8: Command = 10000

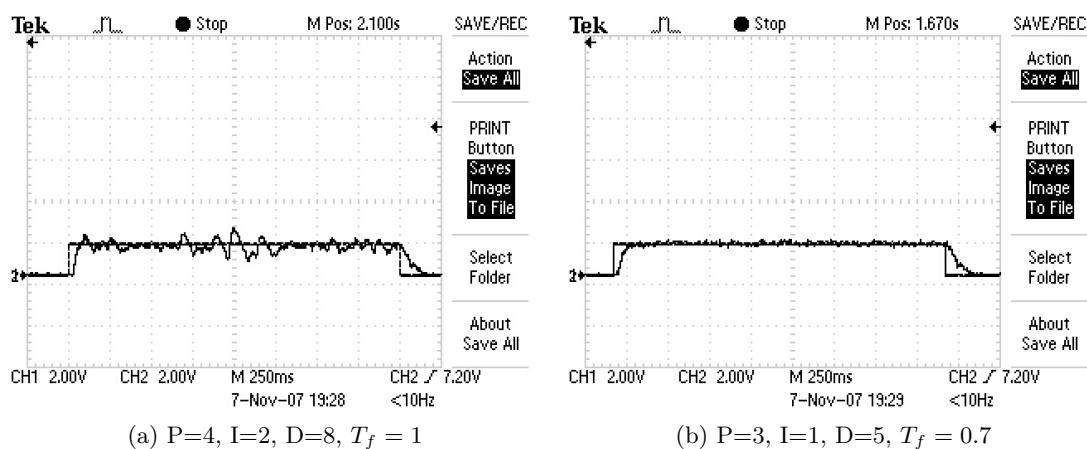


Figure J.9: Command = 5000

# Appendix K

## Datasheets

### K.1 ENC28J60

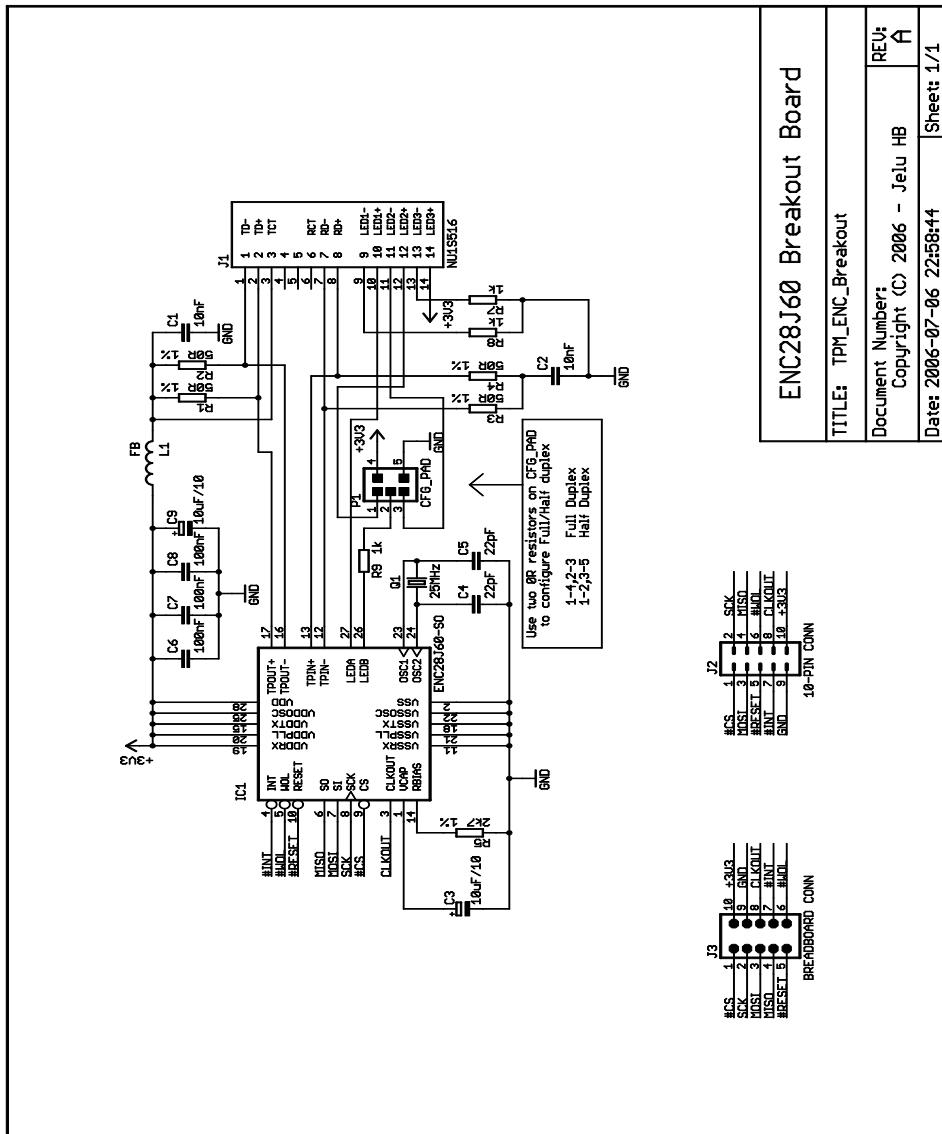


Figure K.1: Schematic for the ENC28J60 board used

## K.2 ROC 416

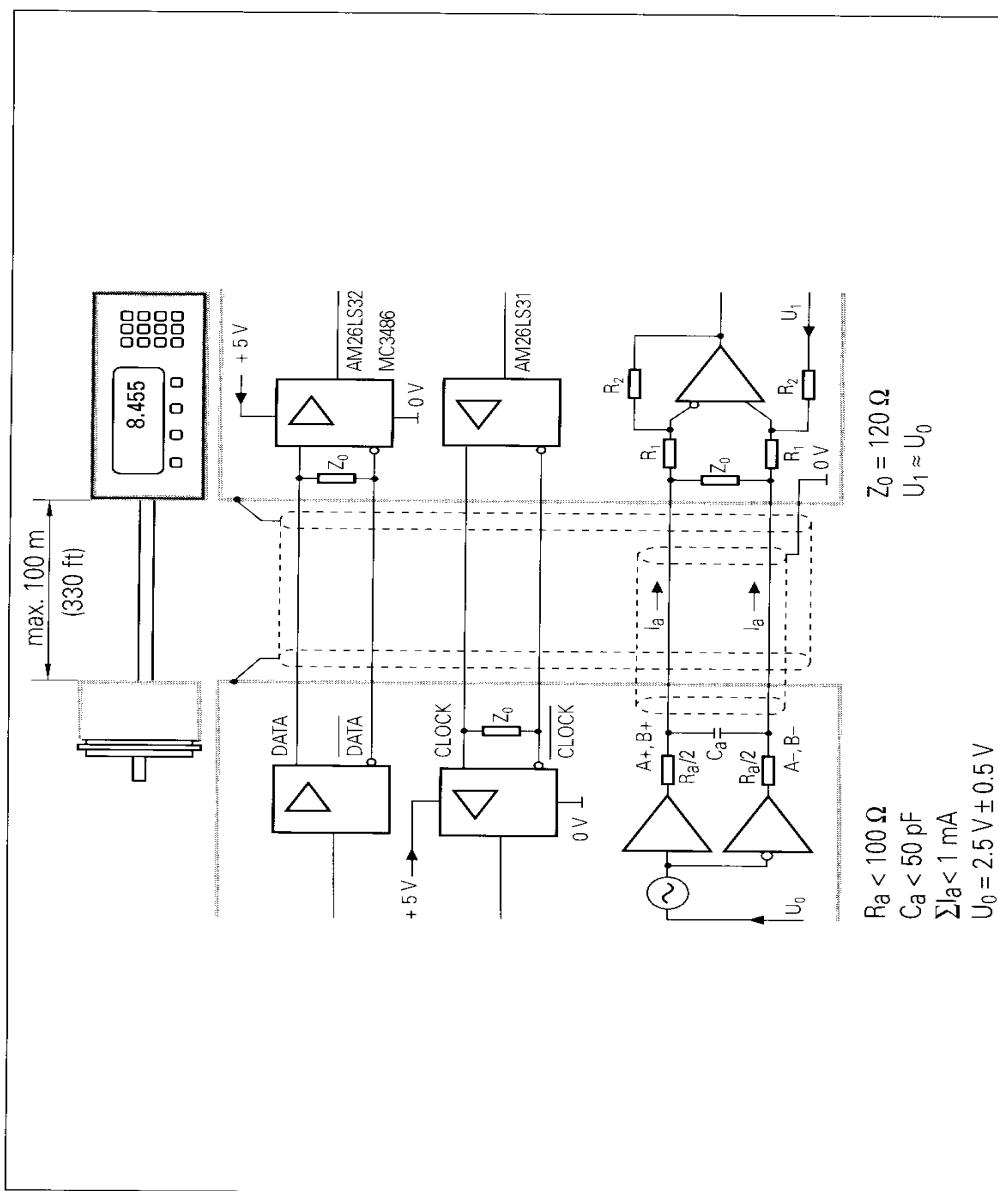


Figure K.2: Schematic of the ROC 416 interface (ROC)

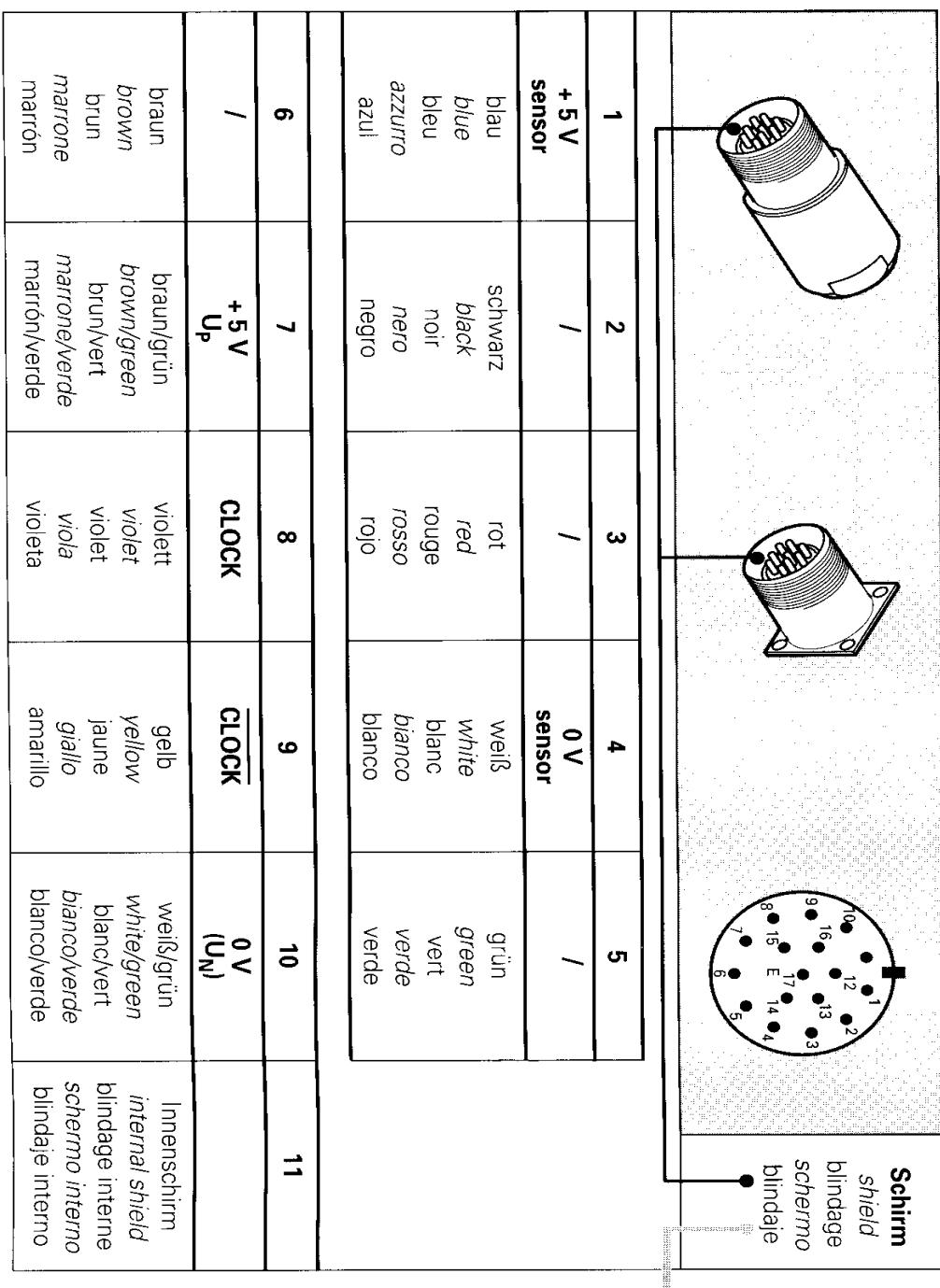


Figure K.3: Part 1 of the ROC 416 wiring diagram (ROC)

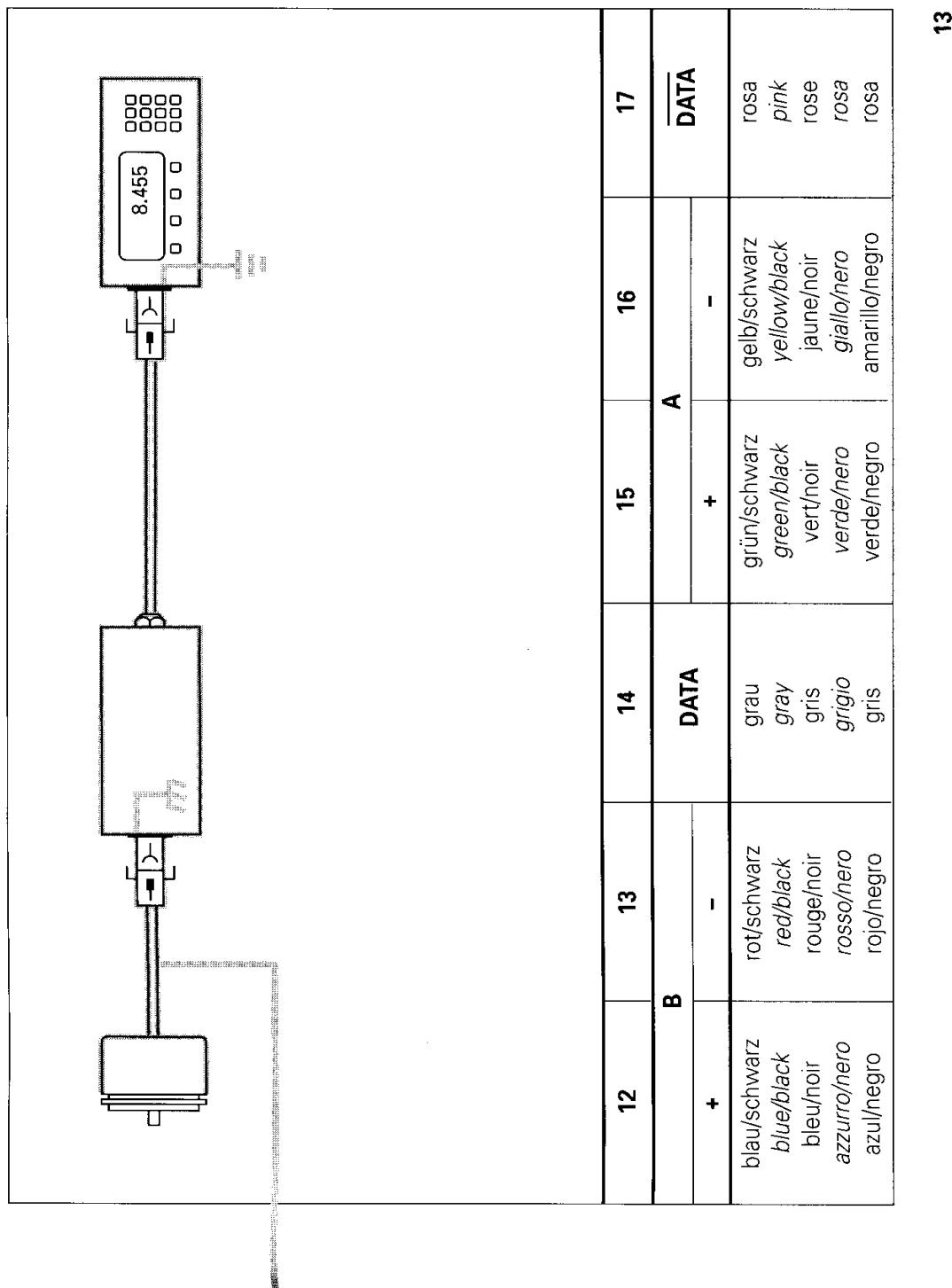


Figure K.4: Part 2 of the ROC 416 wiring diagram (ROC)

### K.3 FRABA Encoder

Table K.1: Description of the FRABA Angle encoder interface (POSITAL FRABA)

Interface	Connector Pin	Wire end
Clock -	3	Yellow
Clock +	4	Green
Data +	1	Grey
Data -	2	Pink
Complement	7	Red
$+U_b=10-30V$	8	Brown
GND	9	White