UK Astronomy Technology Centre

HERSCHEL SPIRE

SPIRE BSM
Interface Control Document

Issue 3.1

26th November 2003

Approval Type	Role	Person	Signature & Date
Prepared by	Mechanical Engineer	Tom Paul	
Prepared by	Electronic / Control Engineer	Brian Stobie	
Release	BSM Project Manager	Phil Parr-Burman	
Approval	ATC Checking Engineer	Colin Cunningham	
Approval	LAM Systems Engineer	Dominique Pouliquen	
Approval	UoC,W Systems Engineer	Peter Hargrave	
Approval	MSSL Systems Engineer	Berend Winters	
Approval	SPIRE Systems Engineer	Doug Griffin	
Approval	SPIRE Instrument Development manager	Eric Sawyer	

DISTRIBUTION LIST

SPIRE-Project	Doug Griffin	
	Bruce M. Swinyard	
	Matt Griffin	
UK ATC	Colin Cunningham	
	Gillian Wright	
	Phil Parr-Burman	
	Brian Stobie	
	Tom Paul	
UoC	Peter Hargrave	
MSSL	Berend Winter	
LAM	Didier Ferrand	
	Dominique Pouliquen	
	Patrick Levacher	

RECORD OF ISSUE

Date	Index	Remarks
10 Jan 2002	0.1	Creation of the document (break out from Design description presented art DDR)
10.Jan.02	1.0	Minor updates and first Issue
07.Feb.02	2.0	Changes to structure mass and inertia properties
24 Nov 03	3.0	Major re-write: Re-structured document to include all interfaces, including electronics in one document
26 Nov 03	3.1	Corrected error in wiring table

Table of Contents

1. Documents ____ *4*

1.1 Applicable documents ____ **4**

1.2 Reference documents ____ **4**

1.3 Glossary ____ **5**

2. ICD Philosophy ____ *6*

3. Dimensions ____ *7*

4. Mass Properties ____ *13*

5. Warm Electronics Interfaces ____ *15*

5.1 BSM Assembly to Warm Electronics ____ **15**

5.2 BSM to Warm Electronics Interface ____ **15**

5.3 Position Sensor Interface ____ **15**

5.3.1 Current Source ____ 15

5.3.2 Position sensor amplifier ____ 16

5.3.3 Motor Interface ____ 16

5.4 Electronics to SMEC Processor Interface ____ **16**

5.4.1 A-D and D-A Signals ____ 16

5.4.2 Chop and Jiggle A-D ____ 16

5.4.3 Chop and Jiggle D-A ____ 16

5.5 PCAL TO WARM ELECTRONICS INTERFACE ____ **16**

5.6 ISOLATION ____ **16**

5.7 CONNECTOR TYPE ____ **16**

6. SOFTWARE REQUIREMENTS ____ *19*

CHANGE RECORD (SOFTWARE) ____ **19**

6.1 INTRODUCTION ____ **19**

6.2 WAVEFORM COMMAND REQUIREMENTS ____ **19**

6.3 CONTROL SYSTEM DIAGNOSTIC DATA ____ **20**

6.4 CONTROL ALGORITHMS ____ **20**

6.4.1 Description ____ 20

6.5 Chop Control ____ **21**

6.5.1 Chop Slew-rate Limiter ____ 21

6.5.2 Chop Position Sensor Linearisation ____ 21

6.5.3 Chop Control Code ____ 22

6.5.4 Chop Control Pseudo-Code ____ 23

6.6 JIGGLE CONTROL ____ **25**

6.6.1 Description ____ 25

6.6.2 Jiggle Slew-rate Limiter ____ 25

6.6.3 Jiggle Sensor Linearisation ____ 26

6.6.4 Jiggle Control Code ____ 27

1. Documents

1.1 Applicable documents

	Title	Author	Reference	Date
AD1	Not used			
AD2	BSM ICD drawing	T. Paul	SPIRE-BSM-061-002 Rev 1. Sheets 1 to 4 Included in this document for information.	24th Nov 2003
AD3	ICD drawing: Structure: BSM	MSSL	Drg No A2/5264/907 issue 9	3rd July 2003
AD4	N/A			
AD5	SPIRE Harness Definition	D.K.Griffin	SPIRE-RAL-PRJ-000608 v1.1	5th March 2003
AD6	PCAL ICD		SPIRE-BSM-020-001-004 Rev 1 (note ATC drawing is the definitive one: UCL ref is SPIRE-UCF-PRJ-001150, dated 6th Feb 2002)	June 2001

1.2 Reference documents

	Title	Author	Reference	Date
RD 1	Thermal Configuration Control Document	S.Heys	SPIRE-RAL-PRJ-000560	18.Apr.01

1.3 Glossary

AD	Applicable Document	MAC	Multi-Axis Controller
CEA	Commissariat à l' Energie Atomique	MCE	Mechanism Control Electronics
CDR	Critical Design Review	MGSE	Mechanical Ground Support Equipment
CNES	Centre National des Etudes Spatiales	MPIA	Max Planck Institute for Astronomy
CoG	Center of Gravity	MSSL	Mullard Space Science Laboratory
CQM	Cryogenic Qualification Model	NA	Not Applicable
DDR	Detailed Design Review	OGSE	Optical Ground Support Equipment
DESPA	Département des Etudes SPAtiales	PFM	ProtoFlight Model
DM	Development Model	RAL	Rutherford Appleton Laboratory
DRCU	Digital Read-out and Control Unit	RD	Reference Document
EGSE	Electrical Ground Support Equipment	BSM	Beam Steering Mirror
Herschel	Far InfraRed Space Telescope	UK ATC	United Kingdom Astronomy Technology Centre
FPU	Focal Plane Unit	BSM	Beam Steering Mirror
FS	Flight Spare model	SPIRE	Spectral and Photometric Imaging REceiver
LAM	Laboratoire d'Astrophysique de Marseille	TBC	To Be Confirmed
FTS	Fourier Transform Spectrometer	TBD	To Be Defined
		WE	Warm Electronics

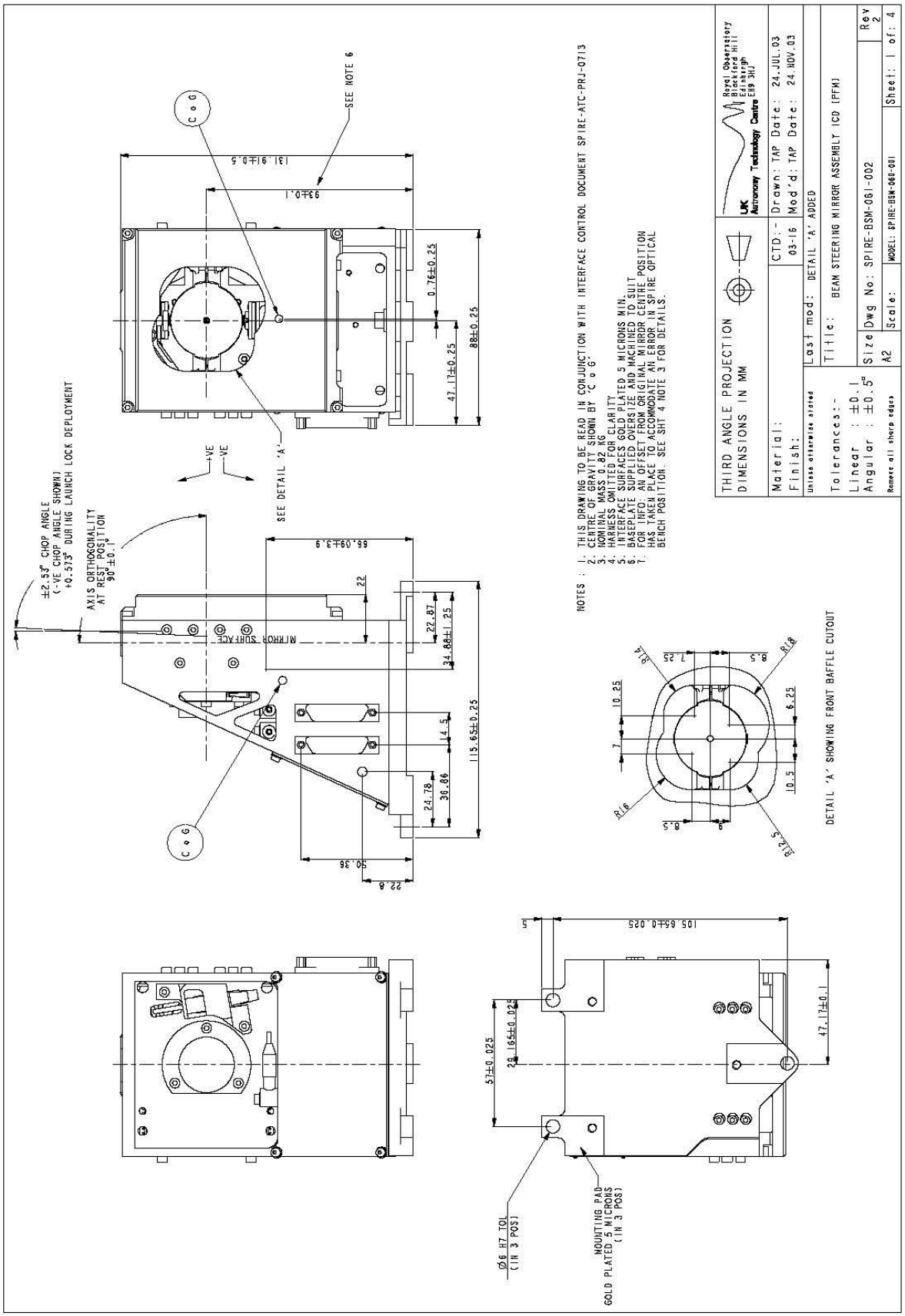
2. ICD Philosophy

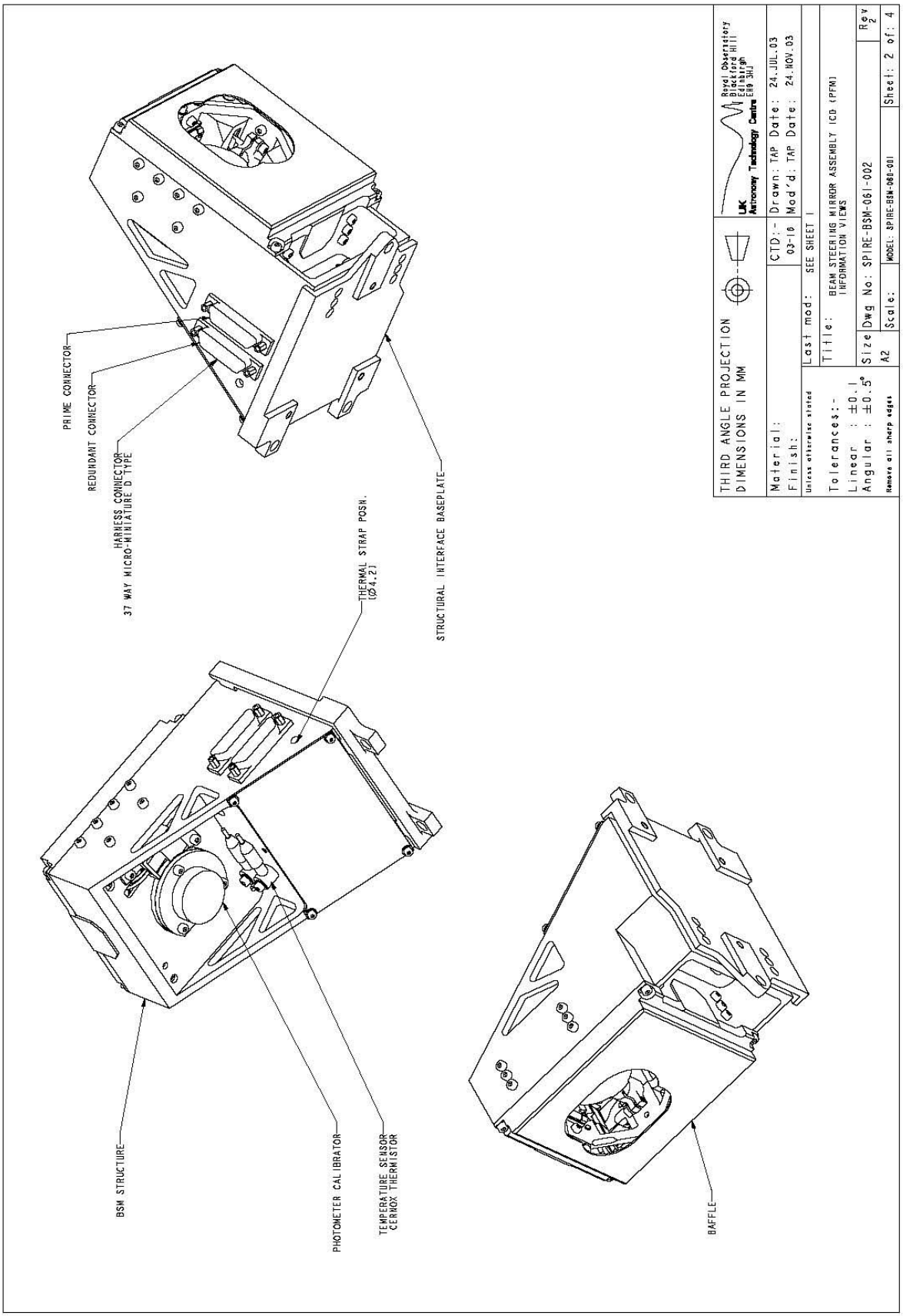
The BSMm interfaces with other subsystems in the SPIRE instrument.
This document defines the interface characteristics of the BSMm for the structure and electronics.

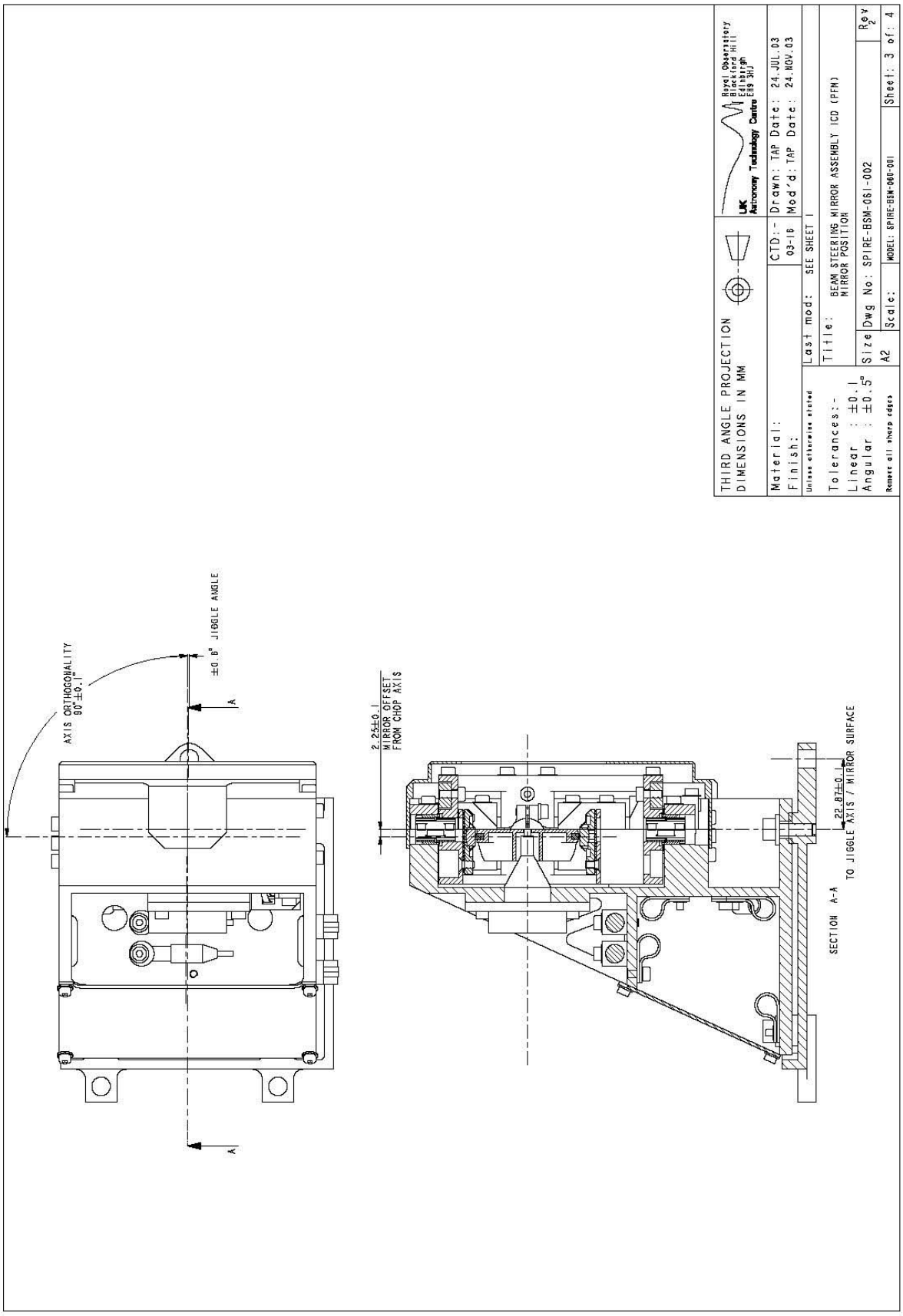
The optical interface (controlled by MSSL) is complex and is not easy to represent in a single ICD. In practice this has been done using 3D files (IGES). In order to have a controlled interface:

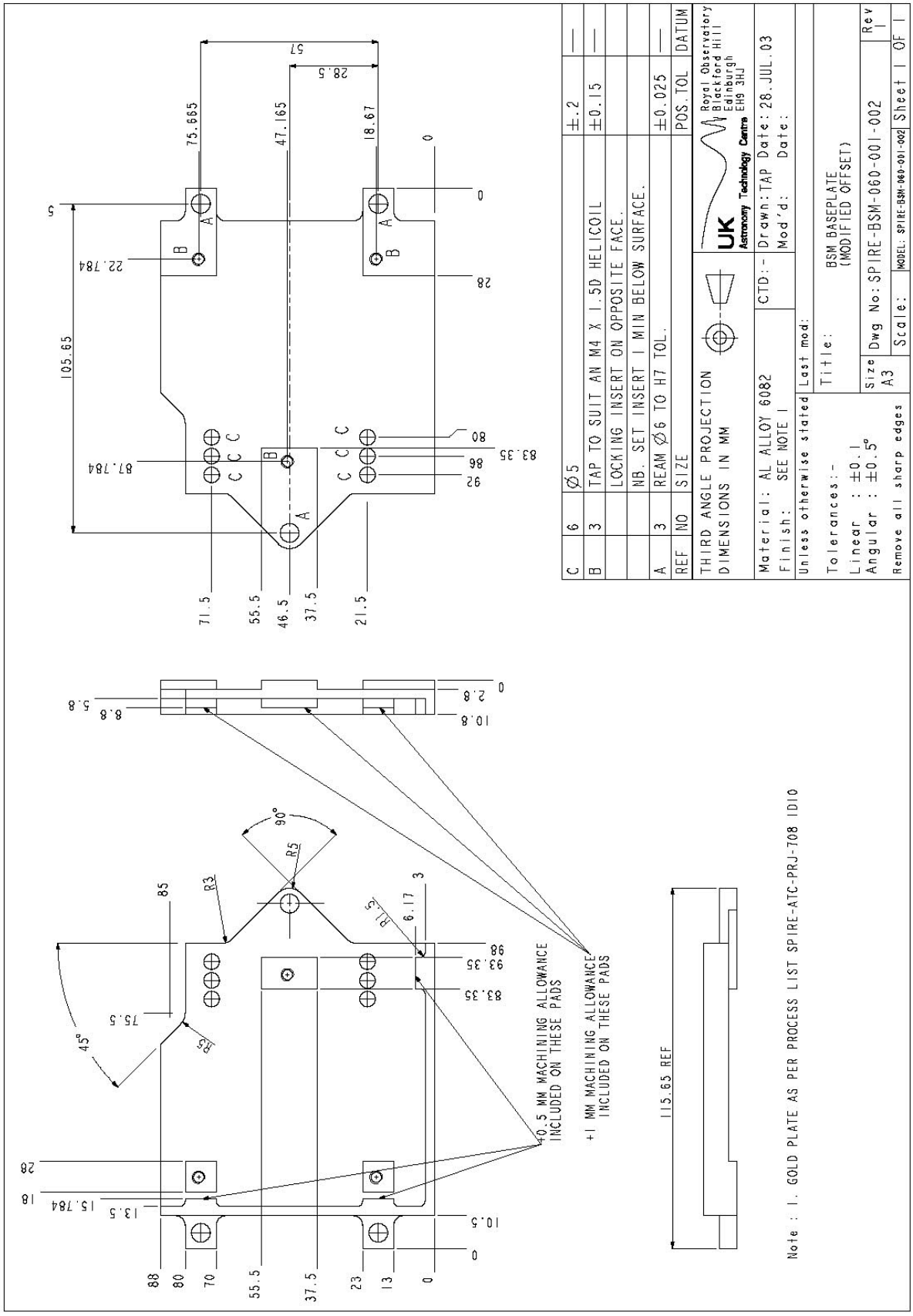
3. Dimensions

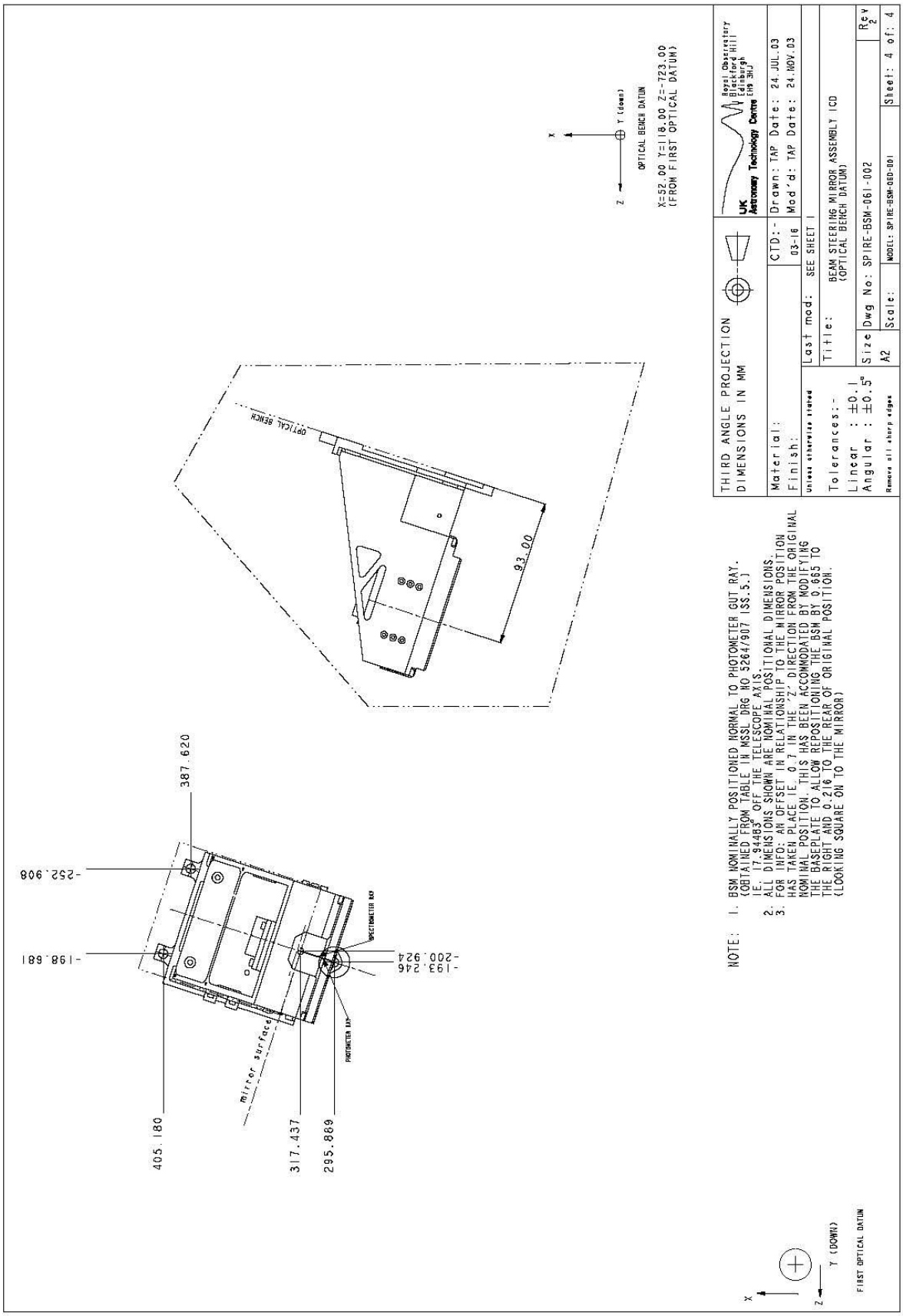
These are defined in:
BSMm interface: drawing no SPIRE-BSM-061-002 Rev 2. 24th Nov 2003. Sheets 1 to 4
Shoe interface: drawing no SPIRE-BSM-060-001-002 Rev 1, 28th July 2003


NOTES : 1. THIS DRAWING TO BE READ IN CONJUNCTION WITH INTERFACE CONTROL DOCUMENT SPIRE-ATC-PRJ-0713
2. CENTRE OF GRAVITY SHOWN BY 'C o G'
3. NOMINAL MASS 0.82 KG
4. HARNESS OMITTED FOR CLARITY
5. INTERFACE SURFACES GOLD PLATED 5 MICRONS MIN.
6. BASEPLATE SUPPLIED OVERSIZE AND MACHINED TO SUIT
7. FOR INFO: AN OFFSET FROM ORGINAL MIRROR CENTRE POSITION HAS TAKEN PLACE TO ACCOMMODATE AN ERROR IN SPIRE OPTICAL BENCH POSITION. SEE SHT 4 NOTE 3 FOR DETAILS.
THIRD ANGLE PROJECTION
DIMENSIONS IN MM
Material:
Finish:
Unless otherwise stated
Tolerances:-
Linear : ±0.1
Angular : ±0.5°
Remove all sharp edges
CTD:- 03-16
Drawn: TAP Date: 24.JUL.03
Mod'd: TAP Date: 24.NOV.03
Last mod: DETAIL 'A' ADDED
Title: BEAM STEERING MIRROR ASSEMBLY ICD (PFM)
Size A2
Dwg No: SPIRE-BSM-061-002
Rev 2
Scale:
MODEL: SPIRE-BSM-060-001
Sheet: 1 of: 4
Royal Observatory Blackford Hill Edinburgh EH9 3HJ
UK Astronomy Technology Centre
SEE NOTE 6
SEE DETAIL 'A'
+VE
-VE
±2.5° CHOP ANGLE (-VE CHOP ANGLE SHOWN)
+0.573° DURING LAUNCH LOCK DEPLOYMENT
AXIS ORTHOGONALITY AT REST POSITION 90°±0.1
MIRROR SUBSTRATE
DETAIL 'A' SHOWING FRONT BAFFLE CUTOUT
MOUNTING PAD GOLD PLATED 5 MICRONS (IN 3 POS)
Ø6 H7 TOL (IN 3 POS)


PRIME CONNECTOR
REDUNDANT CONNECTOR
HARNESS CONNECTOR
37 WAY MICRO-MINIATURE D-TYPE
THERMAL STRAP POSN.
(Ø4.2)
STRUCTURAL INTERFACE BASEPLATE
BSM STRUCTURE
PHOTOMETER CALIBRATOR
TEMPERATURE SENSOR
CERNOX THERMISTOR
BAFFLE
THIRD ANGLE PROJECTION
DIMENSIONS IN MM
Material:
Finish:
Unless otherwise stated
Tolerances:-
Linear : ±0.1
Angular : ±0.5°
Remove all sharp edges
CTD:- 03-16
Last mod: SEE SHEET 1
Drawn: TAP Date: 24.JUL.03
Mod'd: TAP Date: 24.NOV.03
Title: BEAM STEERING MIRROR ASSEMBLY ICD (PFM) INFORMATION VIEWS
Size A2
Dwg No.: SPIRE-BSM-061-002
Rev 2
Scale:
MODEL: SPIRE-BSM-060-001
Sheet: 2 of: 4
UK Astronomy Technology Centre
Royal Observatory Blackford Hill Edinburgh EH9 3HJ


THIRD ANGLE PROJECTION
DIMENSIONS IN MM
Material:
Finish:
Unless otherwise stated
Tolerances:-
Linear : ±0.1
Angular : ±0.5°
Remove all sharp edges
UK Astronomy Technology Centre
Royal Observatory Blackford Hill Edinburgh EH9 3HJ
CTD:- 03-16
Drawn: TAP Date: 24.JUL.03
Mod'd: TAP Date: 24.NOV.03
Last mod: SEE SHEET 1
Title: BEAM STEERING MIRROR ASSEMBLY ICD (PFM) MIRROR POSITION
Size A2
Dwg No: SPIRE-BSM-061-002
Scale:
MODEL: SPIRE-BSM-000-001
Rev 2
Sheet: 3 of: 4
AXIS ORTHOGONALITY 90°±0.1°
±0.6° JIGGLE ANGLE
A
2.25±0.1 MIRROR OFFSET FROM CHOP AXIS
22.87±0.1 TO JIGGLE AXIS / MIRROR SURFACE
SECTION A-A

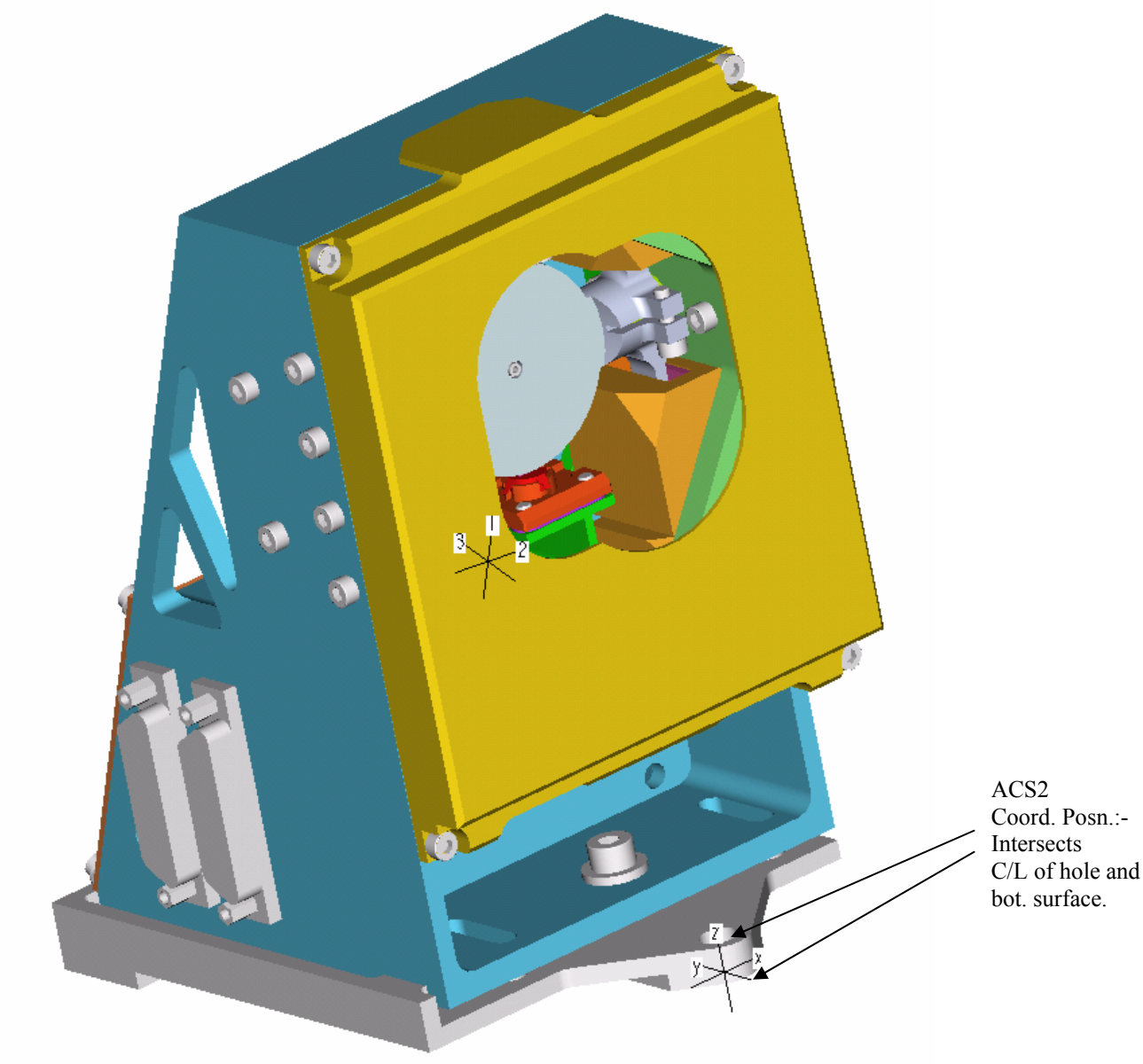

C 6 Ø5 ±.2 —
B 3 TAP TO SUIT AN M4 X 1.5D HELICOIL ±0.15 —
LOCKING INSERT ON OPPOSITE FACE.
NB. SET INSERT 1 MIN BELOW SURFACE.
A 3 REAM Ø6 TO H7 TOL. ±0.025 —
REF NO SIZE POS.TOL DATUM
THIRD ANGLE PROJECTION
DIMENSIONS IN MM
Royal Observatory Blackford Hill Edinburgh EH9 3HJ
UK Astronomy Technology Centre
Material: AL ALLOY 6082
Finish: SEE NOTE 1
CTD:-
Drawn: TAP Date: 28.JUL.03
Mod'd: Date:
Unless otherwise stated
Last mod:
Tolerances:-
Linear : ±0.1
Angular : ±0.5°
Title: BSM BASEPLATE (MODIFIED OFFSET)
Size A3
Dwg No: SPIRE-BSM-060-001-002
Rev 1
Scale:
MODEL: SPIRE-BSM-060-001-002
Sheet 1 OF 1
Remove all sharp edges
+0.5 MM MACHINING ALLOWANCE INCLUDED ON THESE PADS
+1 MM MACHINING ALLOWANCE INCLUDED ON THESE PADS
115.65 REF
Note : 1. GOLD PLATE AS PER PROCESS LIST SPIRE-ATC-PRJ-708 ID10


OPTICAL DATUM
X=52.00 Y=118.00 Z=-723.00
(FROM FIRST OPTICAL DATUM)
OPTICAL BENCH DATUM
Y (down)
X
Z
93.00
387.620
-252.908
-198.681
-193.246
-200.924
405.180
317.437
295.869
mirror surface
SPECTROMETER BAY
PHOTOMETER BAY
Y (DOWN)
FIRST OPTICAL DATUM
NOTE: 1. BSM NOMINALLY POSITIONED NORMAL TO PHOTOMETER GUT RAY. (OBTAINED FROM TABLE IN MSSL DRG NO 5264/907 ISS.5.) IE. 17.9448° OFF THE TELESCOPE AXIS.
2. ALL DIMENSIONS SHOWN ARE NOMINAL POSITIONAL DIMENSIONS.
3. FOR INFO: AN OFFSET IN RELATIONSHIP TO THE MIRROR POSITION HAS TAKEN PLACE IE. 0.7 IN THE -Z- DIRECTION FROM THE ORIGINAL NOMINAL POSITION. THIS HAS BEEN ACCOMMODATED BY MODIFYING THE BASEPLATE TO ALLOW REPOSITIONING THE BSM BY 0.665 TO THE RIGHT AND 0.216 TO THE REAR OF ORIGINAL POSITION. (LOOKING SQUARE ON TO THE MIRROR)
THIRD ANGLE PROJECTION
DIMENSIONS IN MM
Material:
Finish:
Unless otherwise stated
Tolerances:-
Linear : ±0.1
Angular : ±0.5°
Remove all sharp edges
CTD:- 03-16
Last mod: SEE SHEET 1
Title: BEAM STEERING MIRROR ASSEMBLY ICD (OPTICAL BENCH DATUM)
Size A2
Dwg No: SPIRE-BSM-061-002
Scale:
MODEL: SPIRE-BSM-060-1001
Rev 2
Sheet: 4 of: 4
UK Astronomy Technology Centre
Royal Observatory Blackford Hill Edinburgh EH9 3HJ
Drawn: TAP Date: 24.JUL.03
Mod'd: TAP Date: 24.NOV.03

4. Mass Properties

Mass properties for the assembly are shown below:

NB. ADD ON 50g FOR WIRING TO THE MASS FIGURE SHOWN BELOW




```
VOLUME =  2.4021643e+05  MM^3
SURFACE AREA =  2.3483331e+05  MM^2
AVERAGE DENSITY =  3.1293682e-06 KILOGRAM / MM^3
MASS =  7.5172564e-01 KILOGRAM

CENTER OF GRAVITY with respect to ACS2 coordinate frame:
X   Y   Z     9.3911513e-02  3.9517599e+01  5.8794322e+01  MM
```

```
INERTIA with respect to ACS2 coordinate frame:  (KILOGRAM * MM^2)

INERTIA TENSOR:
Ixx Ixy Ixz  5.3326860e+03  2.4894535e+01 -7.9300866e+01
Iyx Iyy Iyz  2.4894535e+01  4.2932193e+03 -1.4516490e+03
Izx Izy Izz -7.9300866e+01 -1.4516490e+03  2.0907542e+03

INERTIA at CENTER OF GRAVITY with respect to ACS2 coordinate frame:  (KILOGRAM *
MM^2)

INERTIA TENSOR:
Ixx Ixy Ixz  1.5602162e+03  2.7684307e+01 -7.5150240e+01
Iyx Iyy Iyz  2.7684307e+01  1.6946683e+03  2.9491818e+02
Izx Izy Izz -7.5150240e+01  2.9491818e+02  9.1682229e+02

PRINCIPAL MOMENTS OF INERTIA:  (KILOGRAM * MM^2)
I1  I2  I3   8.0911662e+02  1.5687257e+03  1.7938645e+03

ROTATION MATRIX from ACS2 orientation to PRINCIPAL AXES:
       0.10598        0.99432        0.01007
      -0.31717        0.02420        0.94806
       0.94243       -0.10367        0.31794

ROTATION ANGLES from ACS2 orientation to PRINCIPAL AXES (degrees):
angles about x  y  z -71.461          0.577        -83.916

RADII OF GYRATION with respect to PRINCIPAL AXES:
R1  R2  R3 3.2807707e+01  4.5681865e+01  4.8850063e+01  MM

---------------------------------------------
```

5. Warm Electronics Interfaces

This section outlines the electrical interface between the Beam Steering Mirror (BSM) assembly and the warm electronics.

The BSM assembly comprises two controllable axes with position sensors and torque motors. It operates at 4 deg. K, remotely (approx. 5m) from the warm electronics, which operates at ~300 deg K.

The BSM electronics has 3 types of analogue sub-circuits per axis :

a) Position sensor current source
b) Position sensor output instrumentation amplifier
c) Motor power amplifier.

The PCAL mechanism and thermometry wiring are also attached to the BSM structure and harness.

All BSM sub-circuits and associated wiring are duplicated to provide full redundancy. In the event of a detected error, the faulty component will be switched out and the backup component used.

5.1 BSM Assembly to Warm Electronics

Each position sensor has 5 connections to a magnetoresistive element designed to form half of a Wheatstone bridge. Two connections supply current and three sense the bridge voltage. The warm electronics has a constant current source and a set of differential receivers for each axis sensor. It also contains voltage-to-current power amplifiers for each axis motor. This ensures that the motor is driven by the correct current independent of its resistance.
To enable back-EMF sensing for the estimation of rate if a position sensor fails, 4-wire connections are used for the motors.
The BSM wiring diagram is shown in Figure 1, and Table 1 lists the wiring requirements for screening and pairing of signals.

Note that the prime and redundant wiring is identical and has separate connectors, so Figure 1 shows the wiring for either the Prime or Redundant circuits. Therefore, the complete BSM wiring will comprise twice the wiring shown in Figure 1.
Dashed lines around an assembly indicate a common assembly to both Prime and Redundant circuits.
The instrument wiring harness wiring for the motor is defined to have a maximum resistance of 20 ohms.

5.2 BSM to Warm Electronics Interface

The position sensor and power amplifier connect to the warm electronics power amplifiers, differential amplifiers and A-D and D-A converters.

5.3 Position Sensor Interface

5.3.1 Current Source

Each position sensor requires a constant current from a suitable precision source, typically Analog Devices AD584.

Value	1.0 mA per sensor
Tolerance on value	+/- 5%
Variation over 4 hours	< +/- 80 ppm (assumes temperature variation of 1 deg/hr in electronics)
Noise	< 1.0 μA rms, 0 to 25 Hz (nominal)
Load Voltage capability	> 1V

5.3.2 Position sensor amplifier

The position sensors require a high-impedance differential amplifier, typically Analog Devices AD524.

Noise	< 1 μV rms (input)	

5.3.3 Motor Interface

Each axis motor is driven by a voltage-to-current amplifier.

Transfer function gain	5 mA/V +/- 5%	
Current sensing resistor	10Ω	
Load impedance	< 500 Ω	@20 deg.C
	< 1 Ω	@ 4 deg.K
Load Current	50 mA peak	
-3 dB bandwidTH	>= 5 kHz	

5.4 Electronics to SMEC Processor Interface

5.4.1 A-D and D-A Signals

Sample Rate

The sample rate for chop and jiggle position sensor A-D and motor drive D-A interfaces is 360 μS.

5.4.2 Chop and Jiggle A-D

Resolution	16 bits minimum
Full Scale i/p	+/- 10V
Conversion time	To suit loop sample time

5.4.3 Chop and Jiggle D-A

Resolution	8 bits minimum
Full Scale o/p	+/- 10V
Conversion time	To suit loop sample time

5.5 PCAL TO WARM ELECTRONICS INTERFACE

Refer to the Design Description Document,Annex B for the PCAL INTERFACE.

To minimise noise coupling, the PCAL wiring is kept physically separate from the BSM wiring, and the connector spare pins are allocated between these groups also.

To minimise BSM internal wire bundle sizes, most of the harness screens terminate (open-circuit) at the 37-way connector, with the exception of the sensor screens.

5.6 ISOLATION

The BSM electronics and wiring will have a minimum resistance to chassis of 10 Mohm at 100V DC.

5.7 CONNECTOR TYPE

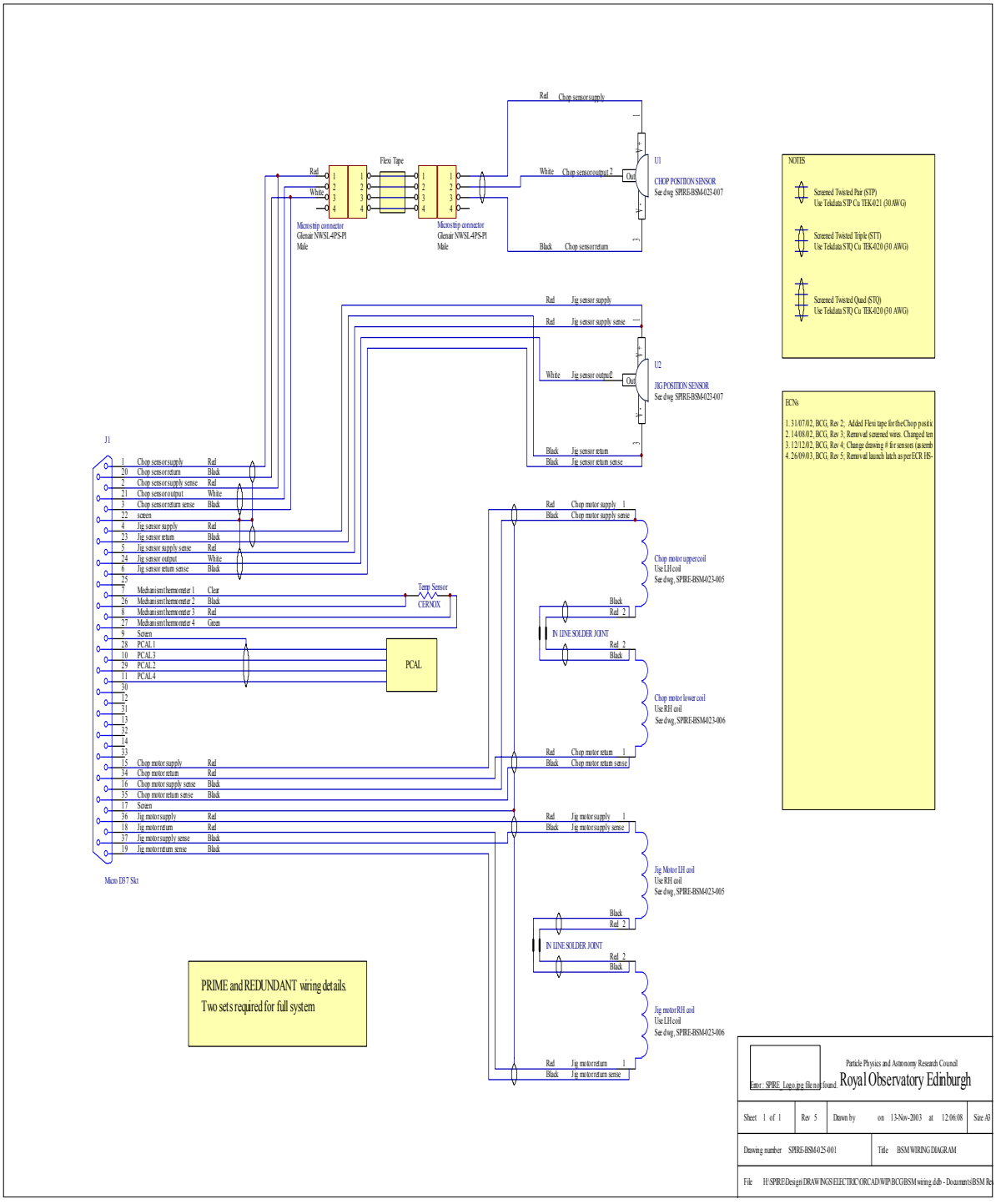
The Connectors used will be 37-way MDM micro-D types with sockets, one connector for prime wiring and one for redundant wiring.

TABLE 1 BSM Prime and Redundant Wiring

STP = Screened Twisted Pair
STT = Screened Twisted Triple
STQ = Screened Twisted Quad

Pin	Title	Max Voltage	Max Current	Wire Type /Comment
15	Chop motor supply	+/- 15 V	40 mA	STP(15,34)
34	Chop motor return	0V	40 mA	STP(15,34)
16	Chop motor supply sense	+/- 15 V	10 µA	STP(16,35)
35	Chop motor return sense	0V	10 µA	STP(16,35)
36	Jiggle motor supply	+/- 15 V	40 mA	STP(36,18)
18	Jiggle motor return	0V	40 mA	STP(36,18)
37	Jiggle motor supply sense	+/- 15 V	10 µA	STP(36,19)
19	Jiggle motor return sense	0V	10 µA	STP(37,19)
17	Motor Screens			screens (commoned)
1	Chop sensor supply	0.4 V	1 mA	STP(1,20)
20	Chop sensor return	0V	1 mA	STP(1,20)
2	Chop sensor supply sense	0.4 V	10 µA	STT(2,3,21)
21	Chop sensor o/p	0V	10 µA	STT(2,3,21)
3	Chop sensor return sense	0.4V	10 µA	STT(2,3,21)
4	Jiggle sensor supply	0.4 V	1 mA	STP(4,23)
23	Jiggle sensor return	0V	1 mA	STP(4,23)
5	Jiggle sensor supply sense	0.4 V	10 µA	STT(5,6,24)
24	Jiggle sensor o/p	0V	10 µA	STT(5,6,24)
6	Jiggle sensor return sense	0.4V	10 µA	STT(5,6,24)
22	Sensor screens			screens (commoned)
7	Mechanism Thermometer 1	0V	2.5 nA	Single
26	Mechanism Thermometer 2	0V	2.5 nA	Single
8	Mechanism Thermometer 3	0V	2.5 nA	Single
27	Mechanism Thermometer 4	0V	2.5 nA	Single
25	Spare			
13	Spare			
32	Spare			
14	Spare			
33	Spare			
12	Spare			
30	Spare			
31	Spare			
28	PCAL 1	TBD	TBD	STQ(28,29,10,11)
29	PCAL 2	TBD	TBD	STQ(28,29,10,11)
10	PCAL 3	TBD	TBD	STQ(28,29,10,11)
11	PCAL 4	TBD	TBD	STQ(28,29,10,11)
9	PCAL Screen			STQ screen

FIGURE 1 The BSM Prime (and Redundant) Wiring.



6. SOFTWARE REQUIREMENTS

CHANGE RECORD (SOFTWARE)

1.0	First Issue
1.1	Remove 'Control Structure Command Requirements' section – include necessary flags, etc in control parameters table. Correct observer coefficients used in code for variable 'cha_h'. Add note to indicate that any suitable code for sinusoidal shaping would be OK (maybe some easier assembler method ?)
1.2	Fix some variable name errors. Correct integral gain term in expression for 'ch_pe'. Include 'Kt' in control parameter list.
1.3	Add the Jiggle code – copy of Chop code with name changes
1.4	Incorporated into the BSM Design Description annex G
2.0	Revise control code to use PID with Feed-forward, replace sinusoidal trajectory with slew-rate limit, add sensor linearisation, change sample time.
2.1	Change sample time. Revise Jiggle cross-coupling to include Chop acceleration.
2.2	Revise cross-coupling

6.1 INTRODUCTION

This document describes the BSM software operation in 'pseudo-code' – however, the automatically produced code by the Matlab/Simulink Real-Time Workshop build process that links to the hardware-in-the-loop dSPACE development system used with the BSM should be used in preference, as no manual translation is involved.

The BSM is controlled via software running on a DSP. The BSM software controls the position of the two BSM axes in response to external commands from the host software. Each axis can move independently.
The movement with respect to time is profiled via stored parameters to give a minimum energy, minimum noise position change, particularly for step commands.
In general the movements are repetitions of the same position/time profile.

In addition, in the event of measured behaviour resulting in a fault diagnosis, some system backup procedures are available.
Diagnosis of excessive position errors and analysis of recorded transient behaviour during operation can result in modifications to the control system by uploading different parameters into electrically-eraseable memory.

NOTE : As each BSM has specific parameters, the indicated parameters in this document are only nominal values, and the specific values for each BSM are given in the calibration document delivered with that model. These parameters are marked "(CAL)".

6.2 WAVEFORM COMMAND REQUIREMENTS

The BSM is slaved to the input demands at all times, so to perform a repetitive chop pattern the host processor has to issue a succession of position demands at the relevant times.

For example, to perform a 2 Hz chop between BSM positions p1 and p2, the following chop axis demand sequence and timing is required.

The command update time (Tu) must always be larger than the loop sample time (Ts). The resultant mirror movements will be limited by the bandwidth of the servo loops.

Time	Demand
0.0 → (0.5-Tu)	p1
0.5 → (1.0-Tu)	p2
1.0 → (1.5-Tu)	p1
1.5 → (2.0-Tu)	p2
2.0 → (2.5-Tu)	p1
etc.	

A step command waveform is assumed. Other waveforms, such as triangular, can be approximated by a succession of incremental step demands, however the resolution will always be dependant on the update rate.

Table 1 : Waveform Commands

No.	Parameter	Value or Range
1	Chop Position	+/- 2.53 degrees (BSM axes)
2	Jiggle Position	+/- 0.573 degrees (BSM axes)

6.3 CONTROL SYSTEM DIAGNOSTIC DATA

Some control system data may be used by the host system to determine detected failure correction activities.

Table 2 : Control System Diagnostic Data

No.	Parameter	Range
1	Chop position	+/- 10V = +/-2.6deg (CAL)
2	Chop motor current	+/- 10V = +/-50mA
3	Jiggle position	+/- 10V = +/-0.6 deg (CAL)
4	Jiggle motor current	+/- 10V = +/-50mA
5	Chop position error	+/- 10V = +/-2.6deg (CAL)
6	Jiggle position error	+/- 10V = +/-0.6deg (CAL)

6.4 CONTROL ALGORITHMS

Both axes have position control loops, running at a fixed sample time of 360 μS, a sample rate of 2.778 kHz.

6.4.1 Description

The algorithm comprises three main parts – the control loop itself, an input slew rate limiter and a sensor linearisation look-up table. On reception of a new position, the transition will be slew-rate limited, and the rate-limited demand applied to the control loop.
The control loop processes the position demand using a digital filter to produce a torque demand to the mirror motor. The algorithm implements a PID servo with position feed-forward.
The slew-rate limiter reduces the rate of change of the Chop demand to a fixed value to reduce power dissipation during step changes.

6.5 Chop Control

6.5.1 Chop Slew-rate Limiter

Chop Constant List

Chop demand slew rate limit:
c_slimit = 0.36 degrees per sample (= 1000 deg/sec) (CAL)

Chop Variable List

c_in	Chop position command from host
c_lim	Slew-limited command to chop position loop
c_lim_p	Previous slew-limited command to chop position loop

CHOP SLEW-RATE LIMITER PSEUDO-CODE

Code is expected to be running at a sample time of Ts = 360 μS.
The symbol % indicates a comment following the symbol.

```
%       ( accept chop position demand 'c_in' from host )
%
IF ABS(c_in – c_lim_p) > c_slimit
THEN
c_lim = c_lim_p + c_slimit * SGN(c_in – c_lim_p)        %  SGN(x) = 1 if x > 0, -1 if x < 0
ELSE
c_lim = c_in
END IF
%
c_lim_p = c_lim                 %  store variables for next cycle
```

6.5.2 Chop Position Sensor Linearisation

Chop Linearisation parameters

Lookup table parameter number (n) (CAL)	cin(deg)	cout(V)
1	(CAL)	(CAL)
2	(CAL)	(CAL)
3	(CAL)	(CAL)
4	(CAL)	(CAL)
5	(CAL)	(CAL)
6	(CAL)	(CAL)
7	(CAL)	(CAL)
8	(CAL)	(CAL)
9	(CAL)	(CAL)
10	(CAL)	(CAL)
11	(CAL)	(CAL)
12	(CAL)	(CAL)
13	(CAL)	(CAL)
14	(CAL)	(CAL)
15	(CAL)	(CAL)
16	(CAL)	(CAL)
17	(CAL)	(CAL)
18	(CAL)	(CAL)
19	(CAL)	(CAL)
20	(CAL)	(CAL)
21	(CAL)	(CAL)

Chop Linearisation Pseudo-code

```
%       ( Read the non-linear Position Sensor output 'pc_nl' )
%       Linearise based on the lookup table values
%       and add alignment offset
%
%       nclut = number of chop lookup table parameter pairs

FOR n = 2 TO nclut
IF (pc_nl < cin(n))AND(pc_nl > cin(n-1))
THEN
pc = cout(n-1) + (cout(n)-cout(n-1))*(pc_nl-cin(n-1))/(cin(n)-cin(n-1))
END IF
IF pc_nl = cin(n)  THEN pc = cout(n)
IF pc_nl = cin(n-1) THEN pc = cout(n-1)
NEXT n
```

6.5.3 Chop Control Code

Description

This pseudo-code description can be used as an alternative to the code automatically produced by dSPACE, which is the hardware-in-the-loop simulation system used in BSM development.

The Control loop is based on a PID scheme, with feed-forward of demand and additional rate feedback. The feed-forward of step demand allows a lower PID loop bandwidth than would otherwise be necessary.

Two backup control schemes are used – for the case of a broken flex, the control code is the same, but the control constants are different. For the case of no position sensor being available, the axis rates are estimated from the motor voltage.
No useful operation is likely with complete flex joint failure AND no position sensor data – however, 'parking' to a useful fixed position may be possible.

Chop Control Parameters

For nominal Chop sensor scale factor of 2.11 V/deg after electronic gain and subtraction, and output scale factor of 5mA/V.

Parameter	Description	Value
prime	prime control scheme	0
broken_flex	control scheme for broken flex joint	1
no_sensor	control scheme for no position sensor	2
chop_control	Select control scheme	prime, broken_flex, no_sensor

Parameter	Description			
		Control Scheme Value		
		Prime (CAL)	**Redundant** (CAL)	**No Pos. Sensor** (CAL)
c_pid_p	PID proportional gain	0.5	0.5	not used
c_pid_i	PID integral gain	55.0	55.0	not used
c_pid_ith	integrator threshold	1.0	1.0	not used
c_pid_ilim	integrator limit	0.04	0.04	not used
c_ff_g	feed-forward gain	0.238	0.238	not used
c_ff_d	feed-forward diff gain	0.0017	0.0017	not used
c_rfbkg	rate feedback gain	0.0045	0.0045	not used
dfilter_t1*	differential filter t.c. 1	0.666666	0.666666	0.92307
dfilter_t2*	differential filter t.c. 2	833.333	833.333	192.307
c_kt	motor BEMF const	not_used	not_used	0.1
mot_r *	motor resistance	not_used	not_used	1.0
mot_l *	motor inductance	not_used	not_used	0.2
c_nps_rsf	rate scale factor	not_used	not_used	1.75
c_nps_psf	position scale factor	not_used	not_used	0.215
nps_t1 *	BEMF rate filter t.c. 1	not_used	not_used	0.03846
nps_t2 *	BEMF rate filter t.c. 2	not_used	not_used	0.92308

* same for Chop and Jiggle

6.5.4 Chop Control Pseudo-Code

```
%       ( Accept position demand 'c_lim' from Slew Rate Limiter )
%
IF (chop_control = prime) OR (chop_control = broken_flex)
THEN
%       PID and Feed-forward controller :
% ------------------------------------------------------------------------------------------------------------
%       ( Accept the linearised Position Sensor output 'pc' )
%
```

```
%       Calculate filtered differential of position demand and position sensor output
%       The sensor differential voltage is also used for the Jiggle axis cross-coupling
%
c_out = c_lim     %   change name to use in this section, else 'stored' version 'c_lim_p' will have
          %   same name as previous section
c_out_diff = dfilter_t1 * c_out_diff_p + dfilter_t2 * (c_out – c_out_p)
c_out_diff_p = c_out_diff
c_out_p = c_out                         %       store variables for next cycle
pc_diff = dfilter_t1 * pc_diff_p + dfilter_t2 * (pc – pc_p)
pc_diff_p = pc_diff
pc_p = pc                               %       store variables for next cycle
%
%       Feedforward components :
%
c_ff_out = c_ff_g * c_out + c_ff_d * c_out_diff
%
%       Rate feedback :
%
c_rfbk =  c_rfbkg * pc_diff
%
%       PI components :
%
c_err = c_out – pc
c_pid_pout = c_pid_p * c_err            % PI proportional term
%
IF ABS(c_err) > c_pid_ith               % zero i/p to integrator if > limit
THEN
c_pid_iin = 0
ELSE
c_pid_iin = c_err
END IF
%
%       PI integral term with anti-windup :
%
c_pid_iout = 0.5*Ts*(c_pid_iin + c_pid_iin_p) + c_pid_ioutl_p       % integrate
IF ABS(c_pid_iout) > c_pid_ilim                                     % anti-windup :
THEN c_pid_ioutl = c_pid_ilim * SGN(c_pid_iout)
ELSE c_pid_ioutl = c_pid_iout
END IF
%
c_pid_ioutl_p = c_pid_ioutl             %       store variables for next cycle
c_pid_iin_p = c_pid_iin                 %       store variables for next cycle
%
%       Controller output :
%
c_con_out = c_ff_out + c_rfbk + c_pid_out + c_pid_ioutl
END IF



IF (chop_control = no_sensor)
THEN
%       Controller for  no sensor :
% ------------------------------------------------------------------------------------------------------------------
%       ( read motor voltage ‘c_mvolts’ and motor current ‘c_mcur’ )
%
%       calculate filtered differential of current
```

```
%
c_mcurd= dfilter_t1*c_mcurd_p + dfilter_t2*(c_mcur – c_mcur_p)
%
%       now estimate rate :
%
c_rate_ns = c_mvolts –  mot_r*c_mcur – mot_l*c_mcurd
c_mcur_p = c_mcur               %       store variables for next cycle
c_mcurd_p = c_mcurd             %       store variables for next cycle
%
%       Filter result :
%
c_rate_nsf = nps_t1*(c_rate_ns + c_rate_ns_p) + nps_t2*c_rate_nsf_p
c_rate_nsf_p = c_rate_nsf       %       store variables for next cycle
c_rate_ns_p = c_rate_ns         %       store variables for next cycle
%
c_con_out = c_nps_psf*c_lim - c_nps_rsf*c_rate_nsf
%
END IF
```

See File SPIRE-ATC-MDL for the Matlab/Simulink model representing the control loop and mechanism.

6.6 JIGGLE CONTROL

6.6.1 Description

The algorithm is essentially similar to the Chop axis, apart from a cross-coupling term and some parameter values to suit the different inertia, motor torque and flex joint characteristics.

6.6.2 Jiggle Slew-rate Limiter

The slew-rate limiter reduces the rate of change of the Jiggle demand to a fixed value to reduce power dissipation during stwep changes.

JIGGLE CONSTANT LIST

Jiggle demand slew rate limit:
j_slimit = 0.03 degrees per 0.4msec (CAL)

JIGGLE VARIABLE LIST

j_in	Jiggle position command from host
j_out	Slew_limited command to Jiggle position loop
j_out_p	Previous slew-limited command to Jiggle position loop

JIGGLE SLEW-RATE LIMITER PSEUDO-CODE

Code is expected to be running at a sample time of Ts = 360 μS.
The symbol % indicates a comment following the symbol.

```
%       ( accept chop position demand 'j_in' from host )
%
IF ABS(j_in – j_lim_p) > j_slimit
THEN
j_lim = j_lim_p + j_slimit * SGN(j_in – j_lim_p)       %  SGN(x) = 1 if x > 0, -1 if x < 0
```

```
ELSE
j_lim = j_in
END IF
%
j_lim_p = j_lim                    %  store variables for next cycle
```

6.6.3 Jiggle Sensor Linearisation

Jiggle Position Sensor Linearisation

Lookup table parameter number (n) (CAL)	jin(deg)	jout(V)
1	(CAL)	(CAL)
2	(CAL)	(CAL)
3	(CAL)	(CAL)
4	(CAL)	(CAL)
5	(CAL)	(CAL)
6	(CAL)	(CAL)
7	(CAL)	(CAL)
8	(CAL)	(CAL)
9	(CAL)	(CAL)
10	(CAL)	(CAL)
11	(CAL)	(CAL)
12	(CAL)	(CAL)
13	(CAL)	(CAL)
14	(CAL)	(CAL)
15	(CAL)	(CAL)
16	(CAL)	(CAL)
17	(CAL)	(CAL)
18	(CAL)	(CAL)
19	(CAL)	(CAL)
20	(CAL)	(CAL)
21	(CAL)	(CAL)

Jiggle Sensor Linearisation Pseudo-code

```
%       ( Read the non-linear Position Sensor output ‘pj_nl' )
%       Linearise based on the lookup table values
%       and add alignment offset
%
%       njlut = number of jiggle look-up table parameter pairs
FOR n = 2 TO 21
IF pj_nl (< jin(n))AND(> jin(n-1))
THEN
pj = jout(n-1) + pj_nl*((jout(n)-jout(n)) / (jin(n)-jin(n-1))
END IF
IF pj_nl = jin(n)  THEN  pj = jout(n)
IF pj_nl = jin(n-1)  THEN  pj = jout(n-1)
NEXT n
```

6.6.4 Jiggle Control Code

Description

Jiggle uses the same algorithm as Chop, except for a cross-coupling factor from the Chop axis to prevent Chop forces from affecting the Jiggle axis.

Jiggle Control Parameters

For Jiggle sensor scale factor of 4.67 V/deg after electronic gain and subtraction, and output scale factor of 5mA/V.

Parameter	Description	Value
prime	prime control scheme	0
broken_flex	control scheme for broken flex joint	1
no_sensor	control scheme for no position sensor	2
jig_control	Select control scheme	prime, broken_flex, no_sensor

Parameter	Description			
		Control Scheme Value		
		Prime (CAL)	**Redundant** (CAL)	**No Pos. Sensor** (CAL)
j_pid_p	PID proportional gain	0.5	0.5	not used
j_pid_i	PID integral gain	100.0	100.0	not used
j_pid_ith	integrator threshold	0.25	0.25	not used
j_pid_ilim	integrator limit	0.04	0.04	not used
j_ff_g	feed-forward gain	0.4	0.4	not used
j_ff_d	feed-forward diff gain	0.0	0.0	not used
j_rfbkg	rate feedback gain	0.0125	0.0125	not used
dfilter_t1 *	differential filter t.c. 1	0.666666	0.666666	0.92307
dfilter_t2 *	differential filter t.c. 2	833.333	833.333	192.307
j_cc1	cross-coupling factor 1	0.0	0.0	not used
j_cc2	cross-coupling factor 2	0.0	0.0	not used
j_kt	motor BEMF const	not_used	not_used	0.12
mot_r *	motor resistance	not_used	not_used	1.0
mot_l *	motor inductance	not_used	not_used	0.2
j_nps_rsf	rate scale factor	not_used	not_used	1.75
j_nps_psf	position scale factor	not_used	not_used	0.516
nps_t1 *	BEMF rate filter t.c. 1	not_used	not_used	0.03846
nps_t2 *	BEMF rate filter t.c. 2	not_used	not_used	0.92308

* same for Chop and Jiggle

Jiggle Control Pseudo-Code

```
%       ( Accept position demand 'j_lim' from Slew Rate Limiter )
%
IF (jiggle_control = prime) OR (jiggle_control = broken_flex)
THEN
%       PID and Feed-forward controller :
% ------------------------------------------------------------------------------------------------------
%       ( Accept the linearised Position Sensor output 'pj' )
%
%       Calculate filtered differential of position demand and position sensor output
```

```
%
j_out = j_lim       %   change name to use in this section, else 'stored' version 'j_lim_d' will have
            %   same name as previous section
j_out_diff = dfilter_t1 * j_out_diff_p + dfilter_t2 * (j_out – j_out_p)
j_out_p = j_out                                 %       store variables for next cycle
pj_diff = dfilter_t1 * pj_diff_p + dfilter_t2 * (pj – pj_p)
pj_p = pj                                       %       store variables for next cycle
%
%           Feedforward  :
%
j_ff = j_ff_g * j_out + j_ff_d * j_out_diff
%
%
%           Rate feedback :
%
j_rfbk =  j_rfbkg * pj_diff
%
%           PI components :
%
j_err = j_out – pj
j_pid_pout = j_pid_p * j_err                    % PI proportional term
%
IF ABS(j_err) > j_pid_ith                       % zero i/p to integrator if > limit
THEN
j_pid_iin = 0
ELSE
j_pid_iin = j_err
END IF
%
%           PI integral term with anti-windup :
%
j_pid_iout = 0.5*Ts*(j_pid_iin + j_pid_iin_p) + j_pid_ioutl_p % integrate
IF ABS(j_pid_iout) > j_pid_ilim                                         % anti-windup :
THEN j_pid_ioutl = j_pid_ilim * SGN(j_pid_iout)
ELSE j_pid_ioutl = j_pid_iout
END IF
%
j_pid_ioutl_p = j_pid_ioutl                     %       store variables for next cycle
j_pid_iin_p = j_pid_iin                         %       store variables for next cycle

%
%           Controller output (including cross-coupling terms from Chop axis) :
%
j_con_out = j_ff + j_rfbk + j_pid_out + j_pid_ioutl  + j_cc1*c_lim + j_cc2*pc_diff
END IF



IF (jiggle_control = no_sensor)
THEN
%           Controller for  no sensor :
% ------------------------------------------------------------------------------------------------------------------------
%           ( read motor voltage ‘j_mvolts’ and motor current ‘j_mcur’ )
%
%           calculate filtered differential of current
%
j_mcurd = dfilter_t1*j_mcurd_p + dfilter_t2*(j_mcur – j_mcur_p)
```

```
%
%       now estimate rate :
%
j_rate_ns = j_mvolts –  mot_r*j_mcur – mot_l*j_mcurd
j_mcur_p = j_mcur                   %       store variables for next cycle
j_mcurd_p = j_mcurd                 %       store variables for next cycle
%
%       Filter result :
%
j_rate_nsf = nps_t1*(j_rate_ns + j_rate_ns_p) + nps_t2*j_rate_nsf_p
j_rate_nsf_p = j_rate_nsf           %       store variables for next cycle
j_rate_ns_p = j_rate_ns             %       store variables for next cycle
%

j_con_out = j_nps_psf*j_lim - j_nps_rsf*j_rate_nsf
%
END IF
```

See File SPIRE-ATC-MDL for the Matlab/Simulink model representing the control loop and mechanism.