



1. INTRODUCTION

This document presents a description of how pointed (ie. Non-mapping) photometer observations are translated from a SPOT observation description into CUS commands for the telescope and instrument. This is the simplest test case for an Observation Request's (OR, used to be called AOT) conversion into CUS, and as such represents a basic example of what needs to be done for all the other, more complex ORs. It also represents a first testbed of what is needed, so we include a series of open issues about the whole conversion process, and what structures might be needed for other, more complex, ORs. A list of assumptions which might be subject to modification, is also given. The implementation of various different Observation Requests is given in this note in a pseudocode that is related to, but different from, the full CUS language. Implementation of this code into CUS language will occur at a later stage,

1.1 Role of This Document

This document aims to provide pseudocode implementations of CUS conversion for the point source observing modes. Since these are the simplest observing modes with SPIRE, this document also acts as a tutorial in how to convert SPOT output into a pseudocode specifications for the CUS conversion. It can thus be used by SPIRE developers as a starting point for the conversion of more complex ORs into CUS.

1.2 Relevant Documents

The AOT Status Review (Clements, 2004), and the PHOT AOT Definition (Fox & Clements 2004) describe the current status of the AOTs, and how observations with the photometer are defined. The SPIRE Operating Modes document (Swinyard & Griffin, 2002) describes the basic commands and parameters for each observing mode. This latter document is in need of expanding and updating, so some assumptions are made here about certain commands. The CUS language is described in detail in The CUS Language Guide (Brunfitt & Porrett, ftp://ftp.rssd.esa.int/pub/HERSCHEL/csdt/releases/doc/cus/guide/cus_language.html) while the connection between CUS and SPOT is described in CUS and PHS – Interaction Technical note (Bonfield & Porrett, HSCDT/TN-043).

1.3 Other Assumptions and notes

Some other aspects of instrument operations are currently assumed. These are listed here:

- Calibration – calibration measurements (PCAL) need to be carried out roughly every 20 minutes (or, for this AOT, 21 minutes since this is 7 nod cycles).
- Pointing – the spacecraft can be pointed with a simple command. This in fact represents a place holder for more complex pointing commands in CUS, but is sufficient to indicate what needs to be done during conversion to CUS.

```
Point(RA, Dec)
```

- Commands for SPIRE are given as a name, and then a series of parameters and values as indicated in the operating modes document eg:

```
POF1(chop_freq, chop_direction, chop_throw, int...)
```



The meaning of relevant parameters will be made clear in the section that discusses the commands in detail if the parameter name is not sufficient. The POF1 parameters here are taken in the order they are given in the observing modes document though ordering is not significant in the actual CUS commands.

- We assume that standard values of chop frequency, chop throw, chop direction etc. have been defined.
- We use a semi-colon ';' at the start of a line to indicate comments in the pseudocode,
- We end statements with a semi-colon and bracket loops with braces '{ }' following their usage in CUS language.
- We end the translation definitions with an `end`.
- Integration time units are seconds.
- We try to include types with variable definitions as in CUS language.

2. OR CONVERSION

The aim of OR conversion is to go from SPOT output to a series of CUS commands. This may include the addition of parameters from a fixed or variable lookup table, and the inclusion of flow steps to allow repeated sub-observations to be made.

In the case of simple photometric observations, using either chop & nod, or a 7-point jiggle, such logic loops are likely to be unnecessary, but we show them here in illustrative examples.

3. THE CHOP & NOD OR

3.1 Input from SPOT

Input from SPOT will appear as a series of named parameters describing the nature of the observation. The parameters contain the user-input values that describe the observation they want performed.

For a chop & nod observation, the SPOT command issued is assumed to be of the form:

```
Mode=chopnod  
RA=  
Dec=  
Int_time=
```

Where RA and Dec describe the source position, and `int_time` is the requested integration time *on source*. This value will be less than the total time taken for the observation since overheads for pointing, calibration etc. must also be taken into account. We assume that the units for `int_time` are seconds.

3.2 Pseudocode Command to CUS

From the input SPOT command, the corresponding CUS pseudocode will be:

```
; SPOT to CUS conversion pseudocode for simple Chop & Nod observation
```



Technical Note

Single Point Photometry AOT Translation D.L. Clements

Ref: SPIRE-ICS-NOT-002141
Issue: Draft 0.2
Date: 6th Sept 2004
Page: 3 of 9

```
; Version 0.2 by Dave Clements Sept 3 2004

; assumes integration time needed is less than the 21 minute period allowed
between calibration observations

; point the telescope

Point(RA, Dec);

; calibrate

POF8();

; how long is the observation in 3 minute nod cycles?

Int Nod_cycles = Int_time mod 180s;

; do these integrations

POF1(chop_freq, chop_direction, chop_throw, nodding, nod_period,
nod_direction, nod_throw, nod_cycles);

; any left?

if (Int_time div 180s) != 0 {

nod_cycles = 1;

POF1(chop_freq, chop_direction, chop_throw, nodding, nod_period,
nod_direction, nod_throw, nod_cycles);

}

; recalibrate

POF8()

End;
```

3.2.1 Preset Parameter Values

Default parameter values are here assumed:

```
Chop_freq=2Hz
Chop_direction=parallel to y-axis
Chop_throw=126"
Nodding=on
Nod_period=3 minutes
Nod_direction=parallel to y-axis
Nod_throw=126"
```

This set of default values will have to be stored in a lookup table associated with the chopnod observation request.



Technical Note

Single Point Photometry AOT Translation D.L. Clements

Ref: SPIRE-ICS-NOT-002141
Issue: Draft 0.2
Date: 6th Sept 2004
Page: 4 of 9

3.2.2 Notes

It is unclear at this point from the observing mode document what parameters POF8, PCAL calibration, requires.

Each Nod Cycle is 3 minutes. It is assumed that this is the time for a full cycle (ie. ON and OFF) and not just for one phase. In line with the operating modes document, we also assume that the ground-based standard ON-OFF-OFF-ON is not being used.

The actual integration time will have to be quantized into 3 minute units since this is the length of the standard nod cycle. This must be reflected in the integration time reported to the user by SPOT, the S/N that will be achieved, and the value fed to the chop & nod command.

4. A MORE ADVANCED CHOP & NOD

The confusion limit on Herschel is likely to be about 20mJy across the SPIRE bands. Since this can be reached to 3σ in 21 minutes or less in each of the bands it is unlikely that anyone will want a longer integration which, if we assume that calibration is needed every 21 minutes, would require a more complex implementation.

However, for the purposes of illustration of how a more complicated AOT might be implemented, let us consider what would be needed for this.

4.1 Chop & Nod with interleaved Calibration

Once again we have the SPOT output as the starting point:

```
Mode=chopnod  
RA=  
Dec=  
Int_time=
```

From this we need to evaluate the number of 21 minute (ie. 1260 seconds) integration blocks needed, and what remains after that.

```
; SPOT to CUS conversion pseudocode for long duration Chop & Nod observation  
  
; Version 0.2 by Dave Clements Sept 3 2004  
  
; assumes integration time needed is greater than the 21 minute period  
allowed between calibration observations  
  
; calculate the number of full 21 minute observing blocks needed  
  
int Nblock = int_time mod 1260 seconds;  
  
; then calculate the time remaining after that
```



Technical Note

Single Point Photometry AOT Translation D.L. Clements

Ref: SPIRE-ICS-NOT-002141
Issue: Draft 0.2
Date: 6th Sept 2004
Page: 5 of 9

```
int Tremain = int_time div 1260 seconds;

; now point the telescope

Point(RA, Dec);

; now do the Nblock sets of 21 minute (ie. 7 nod cycle) integrations

For (int I = 1 .. Nblock) {

    Int nod_cycles = 7;

; calibrate

    POF8();

; observe for 21 minutes ie. 7 nod cycles

    POF1(chop_freq, chop_direction, chop_throw, nodding, nod_period,
        nod_direction, nod_throw, nod_cycles);

; calibrate again

    POF8();

}

; are there any remaining observations to do?

If (tremain > 0) {

; now do the remaining observations, calculating how many 3 minute nod
cycles are needed

; always do 1 more full cycle than is needed so as to get the required
integration time or slightly more

nod_cycles = (Tremain div 180 seconds) + 1;

; do the observations

POF1(chop_freq, chop_direction, chop_throw, nodding, nod_period,
nod_direction, nod_throw, nod_cycles);

; now do the final calibration

POF8();

}

end;
```

4.1.1 Preset Parameter Values

Default parameter values are here assumed:



Technical Note

Single Point Photometry AOT Translation D.L. Clements

Ref: SPIRE-ICS-NOT-002141
Issue: Draft 0.2
Date: 6th Sept 2004
Page: 6 of 9

```
Chop_freq=2Hz
Chop_direction=parallel to y-axis
Chop_throw=126"
Nodding=on
Nod_period= 180 seconds
Nod_direction=parallel to y-axis
Nod_throw=126"
```

This set of default values will have to be stored in a lookup table associated with the chopnod observing request.

5. A 7-POINT JIGGLE FOR A POINT SOURCE

Observing for a 7-point jiggle map is not dissimilar from a chop & nod staring observation. The differences arise in the fact that the beam is not stationary on the target during the observation, but makes a jiggle of 7 positions on and around the target location. There are thus extra parameters needed to specify the jiggle and the timings relative to any nods are different.

The operating modes document describes a standard 7-point as taking 70 seconds, and then the default nodding set up for this will have 1 7-point for each nod position. A complete nod cycle thus takes 140 seconds.

5.1 CUS conversion for 7-point Jiggle

This is broadly similar to the chop&nod with interleaved calibration given above.

SPOT input:

```
Mode=7-point jiggle
RA=
Dec=
Int_time=
```

Assuming that calibration is needed about every 20 minutes, we again find that the closest whole number of nod cycles to this is 21 minutes, ie. 9 nod cycles, or 1260s of integration.

```
; SPOT to CUS conversion pseudocode for long duration 7-point jiggle
observation
```

```
; Version 0.1 by Dave Clements Sept 3 2004
```

```
; assumes integration time needed is greater than the 21 minute period
allowed between calibration observations
```

```
; calculate the number of full 21 minute observing blocks needed
```

```
int Nblock = int_time mod 1260 seconds;
```

```
; then calculate the time remaining after that
```



Technical Note

Single Point Photometry AOT Translation D.L. Clements

Ref: SPIRE-ICS-NOT-002141
Issue: Draft 0.2
Date: 6th Sept 2004
Page: 7 of 9

```
int Tremain = int_time div 1260 seconds;

; now point the telescope

Point(RA, Dec);

; now do the Nblock sets of 21 minute (ie. 7 nod cycle) integrations

For (int I = 1 .. Nblock) {

    Int nod_cycles = 9;

; calibrate

    POF8();

; observe for 21 minutes ie. 7 nod cycles

    POF2(chop_freq, chop_direction, chop_throw, jiggle_pattern,
        chops_per_jiggle, jiggles_per_nod, nodding, nod_period, nod_direction,
        nod_throw, nod_cycles);

; since nodding is being used, we omit the 'total integration time'
parameter given in the operating mode document

; calibrate again

    POF8();

}

; are there any remaining observations to do?

If (tremain > 0) {

; now do the remaining observations, calculating how many 140 second nod
cycles are needed

; always do 1 more full cycle than is needed so as to get the required
integration time or slightly more

    nod_cycles = (Tremain div 140 seconds) + 1;

; do the observations

    POF2(chop_freq, chop_direction, chop_throw, jiggle_pattern,
        chops_per_jiggle, jiggles_per_nod, nodding, nod_period, nod_direction,
        nod_throw, nod_cycles);

; since nodding is being used, we omit the 'total integration time'
parameter given in the operating mode document

; now do the final calibration

    POF8();

}
```



Technical Note

Single Point Photometry AOT Translation D.L. Clements

Ref: SPIRE-ICS-NOT-002141
Issue: Draft 0.2
Date: 6th Sept 2004
Page: 8 of 9

end;

5.1.1 Preset Parameter Values

Default parameter values are here assumed:

```
Chop_freq=2Hz
Chop_direction=parallel to y-axis
Chop_throw=126"
Jiggle_pattern=7-point theta = 6"
Chops_per_jiggle = 20
Jiggles_per_nod = 1
Nodding=on
Nod_period= 140 seconds
Nod_direction=parallel to y-axis
Nod_throw=126"
```

This set of default values will have to be stored in a lookup table associated with the 7-point-jiggle observing request.

6. OPEN ISSUES

Modulo arithmetic is very useful in determining how many observing segments of a given time length are needed. Support for this within CUS is currently unclear. If it is absent, it should be added.

How are scheduling constraints (eg. Roll angle values) to be indicated by SPOT? They don't naturally fit into the simple observation command, with parameters, paradigm assumed here.

How are positions of moving sources (ie. Sources with proper motion or solar system bodies) to be handled? The assumption here is that something in the scheduler provides the RA and Dec for the time of observations to the command that will be processed by the CUS. The RA and Dec received from SPOT is thus the correct value for the actual observation to be made. Alternate equinox values could and should be handled in the same way.

In all the Ors described I have assumed that nodding will happen. This would seem to be the best default, but do we wish to allow the possibility of non-nodded observations?

7. CONVERSION TO CUS LANGUAGE

As far as possible we have used CUS language elements in this pseudocode, but it is not strictly CUS language as yet. Various elements of CUS, including building blocks, the details of instrument commands, and statements such as `data_rate` and `noise_level` have been ignored. These will be left to the CUS experts who will implement this pseudocode.

8. CONCLUSION

The current document achieves two goals. Firstly it acts as an introduction to the specification of OR conversion from the outputs of SPOT into the CUS language to be used by the telescope. Secondly it presents pseudocode implementations of SPOT conversion into CUS for two observing modes –



Technical Note

Single Point Photometry AOT Translation D.L. Clements

Ref: SPIRE-ICS-NOT-002141
Issue: Draft 0.2
Date: 6th Sept 2004
Page: 9 of 9

chop&nod and 7-point jiggle maps. These are the two modes that will be used to observe point sources.