

Publication Year	2001	
Acceptance in OA@INAF	2023-03-09T13:54:46Z	
Title	þÿQPATCH An IDL Module to Simulate Signal Quan	tiza
Authors	MARIS, Michele	
Handle	http://hdl.handle.net/20.500.12386/34018	
Number	PL-LFI-OAT-TN-020	





TITLE: QPATCH – An IDL Module to Simulate Signal Quantization on Sky Patches

DOC. TYPE:	OAT-TN	
PROJECT REF.:	PL-LFI-OAT-TN-020	PAGE: I of IV, 00
ISSUE/REV.:	0	DATE: November 2001

Issued by	M. Maris	Date:	November 2001
	LFI DPC Development Team	Signature:	
Agreed by	F. PASIAN	Date:	December 12, 2001
	LFI Program Manager	Signature:	
Approved by	R.C. BUTLER	Date:	
	LFI Program Manager	Signature:	
Approved by	N. MANDOLESI	Date:	
	LFI Principal Investigator	Signature:	



Document No.: Issue/Rev. No.: Date: Page: PL-LFI-OAT-TN-020 0 November 2001 ii

DISTRIBUTION LIST

Recipient	Company / Institute	E-mail address	Sent



Document No.: Issue/Rev. No.: Date: Page: PL-LFI-OAT-TN-020 0 November 2001 iii

CHANGE RECORD

Issue	Date	Sheet	Description of Change	Release



Document No.: PL-LFI-Issue/Rev. No.: Date: No Page:

PL-LFI-OAT-TN-020 0 November 2001 iv

TABLE OF CONTENTS

2.1 APPLICABLE DOCUMENTS	-2
2.2 REFERENCE DOCUMENTSA	-2
2.3 ACRONYMS LIST	-2
APPENDIX A OPATCH	-5
APPENDIX B DRIVE OPATCH	-7



Document No.:PL-LFI-OAT-TN-020Issue/Rev. No.:0Date:November 2001Page:1

1 SCOPE

This document describes QPATCH a simulation tool to simulate signal quantization over a map or patch of sky without to perform a full simulation of the scan of the sky, i.e. avoinding the generation of a TOD followed by map making.

The main reason to develop this tool is to give a simple way to evaluate the quantization impact on component separation.

1.1 LIMITS OF APPLICABILITY

Basic hypotheses of the present release of QPATCH are:

- 1. Just one quantization step is performed
- 2. Noise is stationary
- 3. Quantization properties are stationary

This introduces limitations:

- 1. 1/f noise and the destriping process can not be introduced easily
- 2. The first quantization effect as the onboard processing is not fully accounted for.
- 3. Variations in the calibration of the quantization step q are not accounted for.
- 4. The number of quantized samples entering a pixel is exactly integer .



2 APPLICABLE/REFERENCE DOCUMENTS

2.1 APPLICABLE DOCUMENTS

- [1] Quantization Errors on Simulated LFI Signals, M. Maris, D. Maino, M. Bersanelli, C. Burigana, D. Mennella, F. Pasian 2000-07-12, PL-LFI-OAT-TN-011
 [5] Guidania Guidania Construction C
- [2] Statistics of a σ/q ~ 2 Data Stream from Planck/LFI
 M. Maris
 2001-06-12, PL-LFI-OAT-TN-18

2.2 REFERENCE DOCUMENTS

- [3] On Board Processing, Compression and Telemetry Rate Bersanelli et al., (2000), Issue #3, April 21, 2000
- [4] Kollár I., (1994), IEEE Trans. Instr. Meas., 43, 733
- [5] Data Streams from the Low Frequency Instrument On-Board the Planck Satellite: Statistical Analysis and Compression Efficiency, Maris M., Maino D., Burigana C., Pasian F., (2000), astro-ph/0005101, accepted for publication in A&ASS

2.3 ACRONYMS LIST

ADU	Analog Digital Unit



Document No.:PL-LFI-OAT-TN-020Issue/Rev. No.:0Date:November 2001Page:3

3 CODE DESCRIPTION

In the present release the code is composed of a driver drive_ qpatch.pro and a single routine qpatch.pro which accepts in input a map *tmap*, generates noise, then the quantized and the reconstructed map.

The quantized map is generated as specified in the following metacode:

For each pixel in the *tmap* N samples of white noise are generated For each noise sample The given noise sample is added to the given map pixel The map pixel is quantized and stored The arithmetic mean of the N quantized samples is computed

This process reproduces the two basic features for

The quantization process is performed according to the formula:

qt = floor[(t - tmin)/q]

where floor[x] is the floor operator which rounds *x* to the bigger integer *I* for which $I \le x$. While *tmin* is the minimum value accepted for the input signal, *q* is the quantization step.

Further constraints are:

if (t < tmin) then qt = 0if (t > tmax) then qt = floor(tmax/q)

where *tmin* and *tmax* are the minimum and maximum signal accepted by the quantization process.

The output of the quantization process is expressed in ADU (Analog/Digital Units), 1 ADU = 1*q. The nominal q for Planck, according to ESA requirements, is $\sigma/q \sim 2$. Tested values have been: $\sigma/q \sim 4$, 2, 1, 1/2. So the quantization step has to be tuned accordingly to the noise r.m.s. expected for the given radiometer and the σ/q of choice.

Note how in this scheme the computation of the average is not followed by any truncation to reduce the real number so far generated to an integer. The reason being the fact that such average is not performed on board but on the ground segment.

In a mission like Planck, the number of averages per pixel in the map is not a constant. In the simplest case the number of samples varies as the sine of the colatitude of the pixel. The present version of the code accepts two rules to fix the number of samples per pixel:



- 1. constant number,
- 2. Number specified pixel by pixel through an external map.

A typical number of averages for Planck is N = 60. But other *N* shall be considered.

The reconstruction process transforms quantized quantities in physical quantities given *q*, *tmin*, it is simply described by the formula:

 $rqt = (qt + \frac{1}{2}) * q + tmin,$

note the $\frac{1}{2}$ offset due to the use of the floor[x] quantization function, as noted in [1] if round[x] is used instead the offset has to be set to zero.

Note how the reconstruction process is performed *after* the averaging instead of *before*. In the hypothesis that averaging generates real numbers instead of integers the two schemes are equivalent but time is saved.



4 PERFORMANCES AND COMMENTS

On a Ultra/SPARK work station a map of ... samples is quantized for N = 60 in ... sec.

The use of IDL make the program quite slow, however since the limited scopes of this code the velocity is acceptable. If required, a future release of qpatch.pro in F90 may be considered.

APPENDIX A QPATCH

This section contains the listing of qpatch.pro

```
pro qpatch,rqmap,qmap,tmap,n,q,sigma,tmin,tmax
```

```
IDL routine to generate quantized maps
 performs just the second quantization plus reconstruction
 assumes the use of floor
 rgmap = reconstructed map after quantization (float)
 qmap = quantized map (float)
 tmap = unquantized input map without noise (float)
       = number of repetitions of the quantization process (integer)
 n
          if N is a number the same number of repetitions is applied to all the
samples of the map
         if N is a matrix with size(N) = size(tmap) each element of N specifies
the number of sampling
         operations applied to the correspondign element of tmap
;
 q
       = quantization step (same units of tmap)
;
 sigma = white noise sigma (same units of tmap)
 tmin = minimum value for the quantized scale (same units of tmap)
 tmax = maximum value for the quantized scale (same units of tmap)
 nbits = number of bits for the quantization
 M. Maris - 0.0 - 21 Nov 2001
 change this offset if you change the quantization mode
 qoffset = 1./2.
 maps initialization
 qmap = tmap
 rqmap = tmap
 if (n_elements(n) eq 0) then
    print, 'Error: n shall contain 1 element or as many elements as tmap'
    return
```



```
endif
```

```
if (n_elements(n) ne n_elements(tmap)) then
     print, 'Error: n shall contain 1 element or as many elements as tmap'
     return
 endif
 quantized, averaged map
;
  if (n_elements(n) eq 1) then
 for r=1,n_elements(tmap) do
      if (n_elements(n) eq 1) then
         wn = random(n, 0, sigma)
      endif else
         wn = random(n(r), 0, sigma)
      endelse
      x = (tmap(r) - tmin + wn)
      x(where(x lt 0)) = 0
      x(where(x gt (tmax-tmin))) = tmax-tmin
      qx = floor(x/q)
      qmap(r) = float(sum(qx))/float(n)
 endfor
;
 reconstructed, quantized, averaged map
;
```

```
rqmap = (qmap + qoffset) * q + tmin
```

end



APPENDIX B DRIVE_QPATCH

This section contains the listing of drive_qpatch.pro, which may be considered an example of use for qpatch.pro.

```
drive_qpatch.pro 0.0 - 2001 Nov 11 - M. Maris
  an example of use of qpatch
  gets the various components
;
readfits,'component1.fits',c1,h
readfits, 'component2.fits', c2, h
  combines them to obtain tmap
;
;
tmap = c1+c2
 quantized maps for 30 and 60 repetitions
 sigma = 1 mK
;
      = 0.5 mK
; q
; tmin = q^{2^{15}}
; tmin = -q^{2^{15}}
qmap,rqmap30,qmap30,tmap,n=30,q=0.5,sigma=1.,tmin=-0.5*2^16,tmax=0.5*2^15
qmap,rqmap60,qmap60,tmap,n=60,q=0.5,sigma=1.,tmin=-0.5*2^16,tmax=0.5*2^15
ï
 writes the result
;
;
writefits,'rqmap30.fits',rqmap60,h
writefits,'rqmap60.fits',rqmap60,h
```