TECHNICAL REPORT

SPACE TELESCOPE SCIENCE INSTITUTE
Operated for NASA by AURA

Title: An improved algorithm for the correction of IR detector non linearity	Doc #: JWST-STScI-002163, SM-12 Date: 13 July 2010 Rev: -
Authors: M. Robberto Phone: 410 – 338 - 4382	Release Date: 19 December 2011

1.0 Abstract

The algorithm currently used by the HST pipeline to correct for the NICMOS or WFC3 detector non-linearity approximates the effect as a polynomial function of the observed counts. However, the physical effect is actually dependent on the incident counts. In this report I propose a new algorithm that applies a polynomial fit to the corrected counts, using iteration to arrive at the best-fit value of the corrected counts. In an example where the non-linearity is exponential, the iterative method provides about one order of magnitude improvement in the accuracy of the non-linearity correction.

2.0 Introduction

Infrared detectors used by HST and JWST are intrinsically non-linear, since the detection of photons is associated to a change (increase) of the PN junction capacitance. This means that the count rate measured at the detector output has to be corrected to recover the actual rate of incoming photons. In the case of NICMOS (Thatte and Dahlen et al. 2009) the linearity correction is performed by the NLINCORR calibration step of the CALNICA pipeline. The procedure assumes 2 regimes:

1. Low and intermediate signal levels, where the detector response deviates from the incident flux in a way that is correctable using the following expression

$$F_c = F \times \left(c_1 + c_2 \times F + c_3 \times F^2\right)$$

   Where c_1, c_2 and c_3 are the correction coefficients, F are the uncorrected counts (in DN, improperly called "flux" in the NICMOS manual) and F_c are the corrected counts also in DN. In practice the coefficient c_1 is set to 1, so that no correction is needed as the signal approaches the zero level.

2. High signal levels, where saturation sets in and the response becomes highly non-linear. This regime is not correctable to a scientifically useful degree;.

Operated by the Association of Universities for Research in Astronomy, Inc., for the National Aeronautics and Space Administration under Contract NAS5-03127

The WFC3 pipeline (Kim Quijano et al. 2009) uses the same type of approach. In the case of WFC3 the polynomial expansion adds a third order term, i.e.

$$F_c = F \times \left(1 + c_2 \times F + c_3 \times F^2 + c_4 \times F^3\right) . \quad (1)$$

Two steps are therefore necessary to correct for the signal non-linearity. First, one has to determine the values of the c_i coefficients; second, one uses these coefficients into the polynomial which converts the counts delivered by the previous step of the pipeline into the true count rate.

Usually the first step is performed by taking a set of flat-field ramps with final illumination levels close to saturation, to probe to full dynamic range of the detector. The coefficients are then calculated by a polynomial minimization routine that converts the measured counts F to the estimated expected counts F_c . To determine the expected counts F_c , one typically assumes that the first part of the ramp is "close enough" to linear, deriving the true slope with a linear fit to e.g. the first half of the data. This strategy has a number of drawbacks. First, the ramp is never really linear for fundamental physical reasons so that in general the slope depends on the interval used to fit the straight line. Reducing the interval to the few sampled points closest to the beginning of the integration, using e.g. the first ¼ or 1/8 of the ramp, will produce a linear fit with slope closer to the true value at the price of higher uncertainty, due to the increased readout noise which becomes more and more relevant as one approaches the beginning of the integration. One must also consider that data taken under substantial illumination may be affected by persistence, especially the early parts of a ramp. Reset anomaly is also usually present in the first part of a ramp. In practice, quite often the first part of a ramp does not look linear at all.

Equation (1) also does not correctly captures the physical basis of the phenomenon, as it is the non-linearity that modulates the true flux, not viceversa. In the next section, I will suggest a different approach to correct for non-linearity.

3.0 Non-linearity correction

Let us look at how the NICMOS/WFC3 method works in practice. Let's start with the first step, the determination of the non-linearity coefficients.

For simplicity, I will assume a flux of 1 DN per "time unit", track it to 100,000 DNs (or time units), and add a non-linearity term using some sort of exponential law, something that cannot be trivially matched by a low-order polynomial. The IDL code to generate this ramp is:

```
b=1.                            ;flux in dn/s
f=b*findgen(100001)             ;generate the “true” ramp
fcal_obs=f*PHYSICAL_nl(f)       ;apply the non-linear departure
plot,f                          ;plot the true ramp
oplot,fcal_obs,linestyle=1      ;overplot the non-linear ramp
```

with the non-linearity function PHYSICAL_NL, containing the exponential law, given by:

```
function PHYSICAL_nl,flux
;THIS IS THE PHYSICAL NON LINEARITY
return,exp(-flux^3/5.5E15)
end
```

The original “true” ramp and the non-linear one are shown in Figure 1.

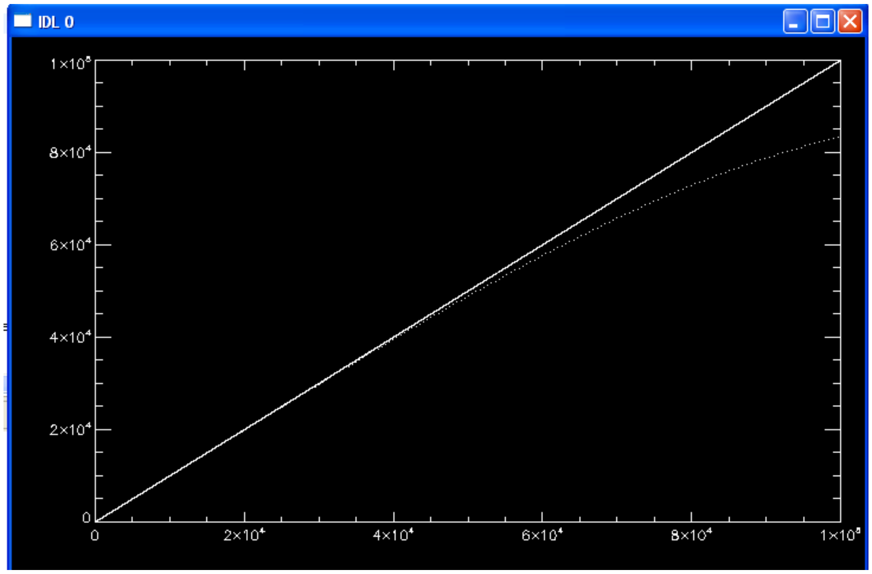

Figure 1 true flux tate (solid line) and detector flux rate (dotted line), corresponding to the true original one modulated by an exponential law

I will now derive the non linearity correction by fitting a third order polynomial to the dotted curve. I will assume, to give the highest power to this method, that the slope of the linearity corrected curve has been perfectly estimated without error. In this case the coefficients of the polynomial fit can be obtained using a simple IDL fitting procedure:

```
A =  [2D-6,0,0]               ;first guess to the set of coefficients
weights = fltarr(N_elements(Fcal_obs))+1   ;weights all = 1
yfit = CURVEFIT(Fcal_obs, f, weights, A, SIGMA, FUNCTION_NAME =
'OLD_CORRECTION_FUNC', /DOUBLE)
reconstructed=fcal_obs+A[0]*fcal_obs^2+A[1]*fcal_obs^3+A[2]*fcal_obs^4
oplot,reconstructed,color=255
```

with the fit function implementing Equation (1):

```
PRO OLD_CORRECTION_FUNC, X, A, F, pder
;x is Fc
;A is the array of coefficients
F = X + A[0]*x^2 + A[1]*x^3 + A[2]*x^4
pder = [ [x^2] , [x^3], [x^4] ]
END
```

The result of the reconstruction is shown as a red line in Figure 2:

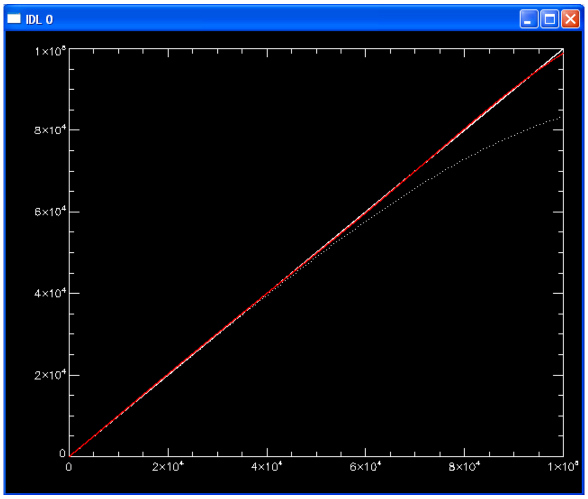

Figure 2 same as Figure (1), with the linear relation constructed from Equation (1)

We can zoom-in and look at the residuals between the reconstructed line and the original one, using the IDL commands:

```
plot,(f-f)/f,yrange=[-0.1,0.1]
oplot,(f-reconstructed)/f,color=255
```

which produce the plot presented in Figure 3.

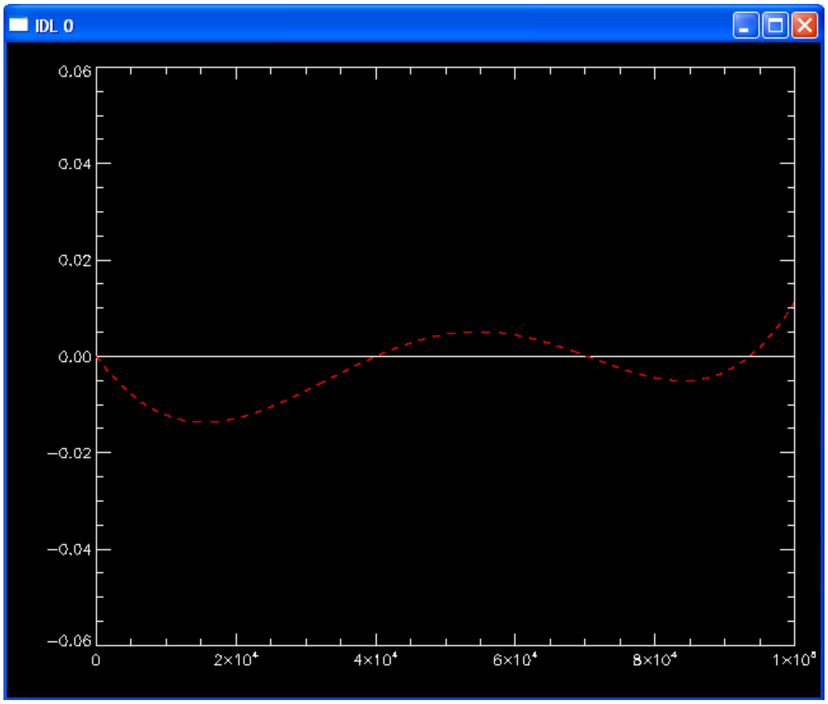

Figure 3 fractional residual of the reconstructed line shown in Figure (2) versus the original line

Figure 3 shows that, with the exception of the beginning of the ramp, the polynomial fit matches the curve within approximately 1%.

One of the problems with this approach is that, even knowing exactly the slope of the true ramp, the reconstructed ramp is not linear. This because Equation (1) assumes implicitly that the problem is with the measure of the signal, rather than with the way it is generated by an intrinsically non-linear detector.

A better approach is possible, which starts from the recognition that Equation (1) does not describe correctly the underlying physical phenomenon. Physically, one can formulate the situation in this way: it is the F_c signal which, being intrinsically linear, gets modulated by a function that depends on the signal F_c itself, producing the observed counts F. I.e.

$$F_c \times P(F_c) = F \tag{2}$$

Now the non-linearity term is on the left side of the equation, not on the right side as in Equation (1). We can express the modulation function, i.e. the non-linearity effect, with a polynomial expansion similar to the one used by CALNICA or CALWFC3. I will actually use the same third order expansion, having:

$$F_c \times \left(1 + c_2 \times F_c + c_3 \times F_c^{\;2} + c_4 \times F_c^{\;3}\right) = F \quad . \tag{3}$$

I will also calculate the c_i coefficients together with the unknown counts (or, better, the count rate b given by $F=bt$) only on the basis of the observed counts F_c and time. The combination of F_c and c_i unknown coefficients makes Equation (3) non linear, but still easily solvable, given the known values of t_i and F_i, using e.g. the IDL procedure CURVEFIT.pro for the general fit to a generic curve:

```
t=dindgen(100001)                ;x-coordinates
A =  [1D-7,-1D-12,-1D-16,1.]  ;first guess to the set of coefficients
weights = indgen(N_elements(Fcal_obs))   ;weights
yfit = CURVEFIT(t, Fcal_obs, weights, A, SIGMA,
FUNCTION_NAME='gfunct',/DOUBLE)
print,A                          ;print the coefficients
;  2.7702732e-007 -7.6269588e-012 -1.1773109e-016      0.99724070
```

having defined the function

```
PRO GFUNCT,X, A, F, pder
;F is the function to fit
F = A[3]*x + A[0]*A[3]^2*x^2 + A[1]*A[3]^3*x^3 + A[2]*A[3]^4*x^4
;pder is the array of 4 derivatives with respect to the parameters
;A[0], A[1], A[2] and A[3]
pder = [ [A[3]^2*x^2] , [A[3]^3*x^3] , [A[3]^4*x^4], [ X +
A[0]*2*A[3]*x^2 + A[1]*3*A[3]^2*x^3 + A[2]*4*A[3]^3*x^4] ]
END
```

Here the A[3] coefficient represents the slope b of the true ramp. In our case, it turns out b=0.9972, which is correct within 0.3%.

This procedure in principle allows to correct “internally” for the non linearity of any ramp. However, in practical cases the number of points at disposal, or the noise level, may make more recommendable to assume fixed values for the c_i coefficients well measured in the calibration program and calculate only the slope *b* using Equation (3). Also in this case the algorithm is slightly less trivial than the old one, since the count rates F_c one wants to calculate now appears on the wrong (left) side of Equation (3). Still, I can then rewrite Equation (3) as:

$$F_c = \frac{F}{\left(1 + c_2 \times F_c + c_3 \times F_c^{\ 2} + c_4 \times F_c^{\ 3}\right)} \tag{4}$$

which can be solved by applying an iterative method. The basic IDL code is:

```
Final=fcal_obs
for i=0,7 do begin & $
  Final = fcal_obs/nl(Final) & $
  oplot,(f-final)/f & $
endfor
```

In Figure 4 I have marked with the red dashed line the original reconstruction of Figure 1 and overplotted the results obtained with the new method as solid curves. The red solid line, in particular, shows the new reconstruction at i=4. It is clear, again, that the error in this case is much smaller than with the old method.

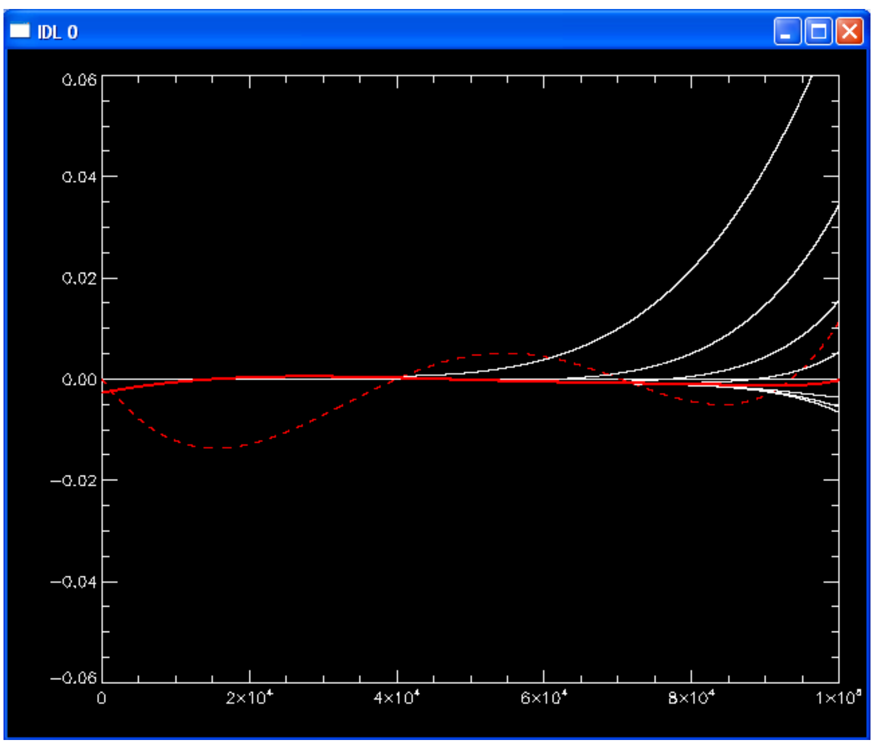

Figure 4 same as Figure (3), with fractional residual of the reconstructed line according to the new proposed method shown as a red solid line.

Up to now I have reconstructed the same ramp used to derive the c_i coefficients. To validate the algorithm, let's try to reconstruct a different ramp. I will assume a ramp with a smaller slope, say 0.8adu/unit, sampled 20 times between 0 and 100000 units. The IDL code to generate these data points is

```
rate=0.8
Ti=INDGEN(21)*5000.
Fi = ti*rate
Fi_nl = Fi*PHYSICAL_nl(Fi)  ;apply the non-linearity effect
plot,ti,Fi
oplot,ti,Fi_nl,psym=1
```

Of course, I have used the same detector exponential non-linearity used to produce Figure (1). The non-linear data points are now shown in Figure 5.

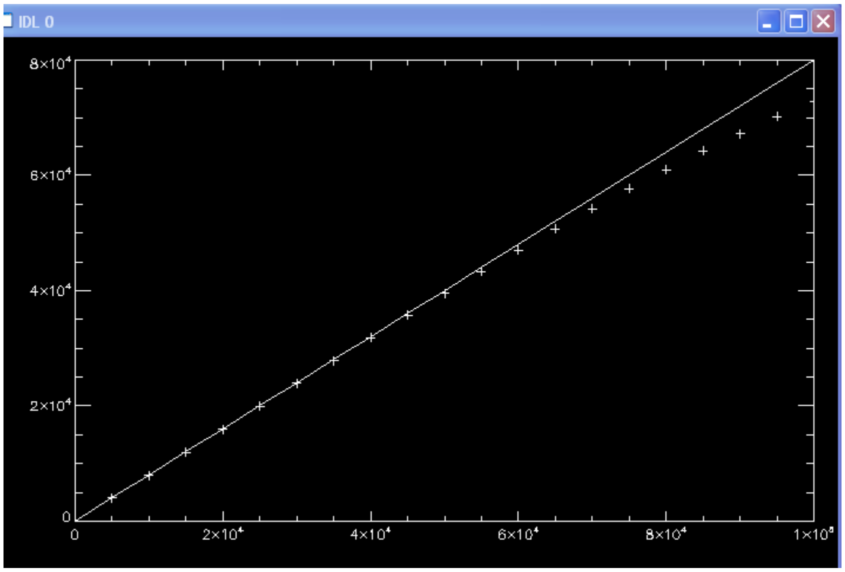

Figure 5 true flux rate (solid line) and detector counts measured as the output of a non linear detector for a 0.8 count/unit ramp. The same exponential law used to create Figure 1 has been applied.

I now correct for non-linearity using the two methods: the old one based on the polynomial expansion of the observed counts and the new iterative one. The correction coefficients are the same estimated before on the calibration ramp.

```
;old way
old_Way= Fi_nl+result[0]*Fi_nl^2+result[1]*Fi_nl^3+result[2]*Fi_nl^4

;new way
final = Fi_nl
for i=0,30 do begin
Final = Fi_nl/nl(Final)
```

```
endfor
```

To compare the results, I directly overplot the residuals to the original true line:

```
plot,ti,(Fi-Fi)/Fi,yrange=[-0.05,0.05],ystyle=1
oplot,ti,(old_way-FI)/Fi,psym=1,color=255
oplot,ti,(final-Fi)/Fi,color=255,psym=2
```

obtaining the plot shown in Figure 6.

Also in this case, the new iterative method gives extremely good results, surpassing by about a factor of 10 the accuracy of the old method.

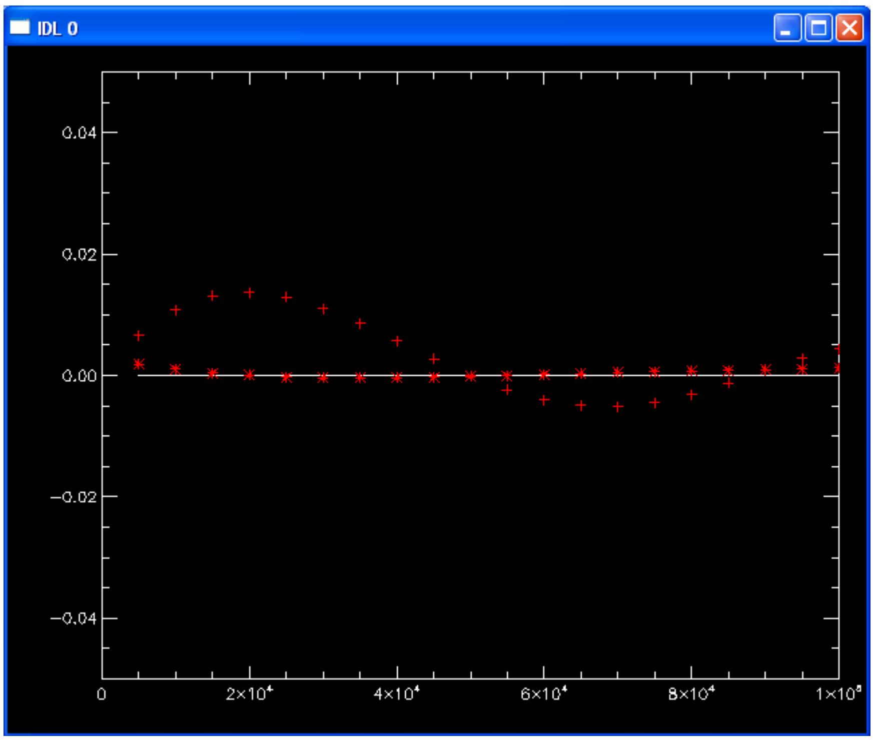

Figure 6 same as Figure 4, with fractional residual of the reconstructed line according to the old (crosses) and the new (asterisks) methods for a ramp different from the one used to derive the correction coefficients.

4.0 Conclusions

There is potential to improve the algorithm currently used to correct for detector non-linearity for NICMOS and WFC3. In this report, I have proposed an alternative algorithm that iteratively derives the corrected counts from the observed counts, but applying a polynomial to the corrected counts. With the same order polynomial, the new algorithm provides a better correction if the non-linearity effect is exponential. The improvement in accuracy is about one order of magnitude in the example illustrated here.

5.0 References

Kim Quijano, J., et al. 2009, "WFC3 Mini-Data Handbook", Version 1.0, (Baltimore: STScI)

Thatte, D. and Dahlen, T. et al. 2009, "NICMOS Data Handbook", version 8.0, (Baltimore, STScI)