TECHNICAL REPORT

SPACE TELESCOPE SCIENCE INSTITUTE
Operated for NASA by AURA

Title: Correction for detector non-linearity using a Generalized Least Square algorithm	Doc #: JWST-STScI-004331, SM-12 Date: 18 February 2015 Rev: -
Authors: M. Robberto Phone: 410 – 338 -4382	Release Date: 10 July 2015

1 Abstract

This report presents a Generalized Least Square algorithm for extracting the count rate from non-linear ramps with group averaging. After reviewing two different strategies of correcting for non-linearity, I concentrate on the one capable of handling grouped frames. In order to solve the problem using the Generalized Least Square, it is necessary to add constraint equations that can be solved with the method of Lagrange multipliers. The report presents the basic IDL code implementing the algorithm and an illustration of the solutions, indicating that slope values derived by the constrained Generalized Least Square are consistent regardless of the grouping scheme adopted to sample the ramp. For a 3rd order polynomial correction, the addition of 3 unknowns Lagrange multipliers on top of the intercept and slope of the signal ramps implies that the method works only for ramps having 5 or more groups.

2 Introduction

The analysis presented in this report is built on a number of results presented in the last few years. It seems therefore appropriate to start by summarizing the relevant previous work.

The general problem of extracting the signal from an integrating detector, like e.g. the IR detectors used in JWST, can be expressed as finding the regression curve that interpolates a set of measures c_i, measured in ADUs, taken at time t_i, measured in seconds.

Assuming the signal flux is constant and the detector response is linear, one deals with a linear regression problem:

$$c_i = a + b \cdot t_i \ , \tag{1}$$

where the intercept a [ADU] represent the zero level response ("bias") of the detector and the slope b [ADU/s] represents the signal flux.

In practice, even assuming constant flux, various factors add a number of complications.

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

First, the signal samples c_i are affected by noise, both random and temporally/spatially correlated. Random, gaussian noise (readout) can be mitigated e.g. by sampling multiple times the integrating signal, while correlated noise can be mitigated by using reference pixels (pedestal jumps, 1/f noise…) or some other signal pixels (as in the case of inter-pixel coupling or cross-talk noise, respectively). In general, the corrections for all these effects are preliminary to the ramp fitting process and will not be discussed hereafter.

More directly related to the regression problem are three other effects:

1. The samples are correlated, as each one contains the integrated signal that was previously sampled. Moreover, groups of reads can be averaged together, adding a second level of "internal correlation" within each group.
2. The signal is affected by cosmic rays. They introduce spurious jumps across the ramps that need to be identified and corrected for.
3. The detector is intrinsically non-linear, as the readout circuitry implemented in the unit cells leaves the bias level variable to reduce noise.

Concerning point 1, the NICMOS and WFC3/IR pipelines have historically ignored the correlation using the standard least square procedure to deriving the signal slope (this step is performed on a ramp previously corrected for cosmic rays and non-linearity). For JWST, however, the effect of correlation has been clarified, deriving the full expression of the covariance matrix between samples (Rauscher et al 2007, Robberto 2009). Knowing the covariance matrix allows using the Generalized Least Square formalism to solve rigorously for the problem of linear regression of integrating signal (Robberto 2013a). This provides the correct solution for the ramp parameters, together with their uncertainties, correlation terms and a meaningful chi-square useful to assess the quality of the fit and provide solid rejection criteria e.g. when the presence of a cosmic ray is doubtful.

The problem of cosmic rays can be separated in two aspects: a) finding them, and b) deriving their strength. Anderson and Gordon (2011) and Martel (2012) have recently presented different algorithms for finding cosmic rays and derive their intensity to produce "cleaned" signal ramps ready for the linear fitting procedure. Finding the intensity of the cosmic ray jump, on the other hand, is not a "local" problem but requires the best knowledge of the signal ramp before and after the event. Therefore, once the presence of a cosmic ray has been identified, it is more appropriate to derive its intensity solving simultaneously also for the slope of the ramp. Robberto (2008) explored this approach, developing a minimization algorithm that calculates the cosmic ray intensity together with the slope, that was assumed to be constant before and after the event. In order to use a modified version of the basic least square fitting procedure, this early exercise neglected Poisson noise, and therefore the non-diagonal terms of the correlation matrix. Recently, Robberto (2013b) returned to this problem showing that the rigorous generalized least square procedure can handle the presence of one, or more, cosmic rays, solving consistently for the ramp and cosmic ray parameters, with their errors, correlation terms and a general chi-square. In fact, having at disposal a reliable chi-square metric allows searching for cosmic ray events across the ramp looking for a statistically acceptable solution. This is especially useful when the cosmic ray has been tentatively

identified with low counts, or the number of samples is limited.

The last problem is the intrinsic non-linear response of the detector, and generally there are two opposite approaches. If the ramp is non-linear, Eq.(1) has to be modified introducing a correction term that can be placed either on the left-hand side or on the right-hand side of the equation; if one corrects on the left side it is

$$F(c_i)\cdot c_i = a + b\cdot t_i \tag{2}$$

otherwise, on the right side:

$$c_i = a + b\cdot t_i \cdot F(b\cdot t_i)\ . \tag{3}$$

In both cases, the functional form is generally assumed to be a polynomial, typically of the 3rd degree. We have therefore, explicitly

$$c_i \cdot \left(1 + \Phi_1 \cdot c_i + \Phi_2 \cdot c_i^2 + \Phi_3 \cdot c_i^3\right) = a + b\cdot t_i \tag{4}$$

or

$$c_i = a + b\cdot t_i \cdot \left[1 + \mathrm{K}_1(b\cdot t_i) + \mathrm{K}_2(b\cdot t_i)^2 + \mathrm{K}_3(b\cdot t_i)^3\right] \tag{5}$$

where we have indicated the set of coefficients with different Greek letters: PHI if the polynomial contains powers of the measured photon flux ("Phos"= light in Greek); KAPPA if the polynomial contains powers of the measured time ("Kronos"=time).

The Left/PHI approach is the one currently adopted by the HST and JWST pipeline. Basically, once the Φ coefficients have been derived from an initial calibration process, one builds powers of the measured counts, c_i, to build a linearized ramp ready for linear regression, possibly with the generalized least square algorithm. The second approach (Robberto 2011a) instead uses the known readout times, given the different set of K coefficients, to reconstruct a linear ramp that can optimally match the observed data. Robberto 2011a adopted a simple iterative algorithm to calculate the two parameters a and b, i.e. not a generalized least square solution. Note that in this mode ramp fitting is not needed, as the linearity correction takes care of providing directly both the slope and intercept parameters.

HST and JWST are actually benefitting from this second approach, as it is used in the initial process leading to the derivation Φ coefficients for the "traditional" method. The method may have a few other advantages, described in Robberto 2011b, but the main one is that, unlike the other Phi method, this Kappa method is robust against grouped frames, as shown in the next section (see also Robberto 2012).

3 Non-linearity with grouped frames

When n frames are grouped, we deal with two different types of quantities: instead of the individual samples c_i, we have averages

$$\overline{c}_i = \frac{\sum_{j=1}^{n_i} c_j}{n_i} \tag{6}$$

and correspondingly for the time we have the averages

$$\overline{t}_i = \frac{\sum_{j=1}^{n_i} t_j}{n_i} \ . \tag{7}$$

Here the index j runs across the indexes within the n_i^{th} group, e.g. $j = 1...4$, $5...8$, etc. I will simplify the notation writing only the n_i index under the summation sign. Using the first, Phi method, one would like to solve the following equation

$$\frac{\sum_{n_i} c_j}{n_i} \cdot \left[1 + \Phi_1 \cdot \frac{\sum_{n_i} c_j}{n_i} + \Phi_2 \cdot \frac{\sum_{n_i} c_j^{\,2}}{n_i} + \Phi_3 \cdot \frac{\sum_{n_i} c_j^{\,3}}{n_i} \right] = a + b \cdot \frac{\sum_{n_i} t_j}{n_i} \tag{8}$$

that directly follows from Eq. **Error! Reference source not found.**. However, one can only deal with the equation

$$\frac{\sum_{n_i} c_j}{n_i} \cdot \left[1 + \Phi_1 \cdot \frac{\sum_{n_i} c_j}{n_i} + \Phi_2 \cdot \left(\frac{\sum_{n_i} c_j}{n_i} \right)^2 + \Phi_3 \cdot \left(\frac{\sum_{n_i} c_j}{n_i} \right)^3 \right] = a + b \cdot \frac{\sum_{n_i} t_j}{n_i} \tag{9}$$

containing the powers of the quantities that were actually measured: the squares and the cubes of the averages are different from the averages of the squares and of the cubes by an amount that increases with the departure from linearity. This means that the Φ coefficients are no longer adequate to reconstruct the ramp. One should envision a method that given the averages iteratively "guesses" the full set of original c_i values, compatible with measured averages while providing, after correction, a nice linear relation.

The Kappa method, on the other hand, is immune to this problem since from Equation (5) one has

$$\frac{\sum_{n_i} c_j}{n_i} = a + b \cdot \frac{\sum_{n_i} t_j}{n_i} \cdot \left[1 + \mathrm{K}_1 \cdot \frac{\sum_{n_i} b \cdot t_j}{n_i} + \mathrm{K}_2 \cdot \frac{\sum_{n_i} \left(b \cdot t_j\right)^2}{n_i} + \mathrm{K}_3 \cdot \frac{\sum_{n_i} \left(b \cdot t_j\right)^3}{n_i} \right]$$
$$= a + b \cdot \frac{\sum_{n_i} t_j}{n_i} \cdot \left[1 + \mathrm{K}_1 \cdot b \frac{\sum_{n_i} t_j}{n_i} + \mathrm{K}_2 \cdot b^2 \frac{\sum_{n_i} t_j^{\,2}}{n_i} + \mathrm{K}_3 \cdot b^3 \frac{\sum_{n_i} t_j^{\,3}}{n_i} \right] \tag{10}$$

Here all polynomial terms are known, being powers of the known readout times; the K coefficients are the same derived for the ramp with single samples: one just has to extract the two parameters a and b.

Again, the iterative algorithm can be used to derive the linear fit parameters (Robberto 2012). The question, of course, is if it possible to derive the parameters using the generalized least square procedure. This would allow extracting the full set of statistically correct results, with the added capability of the generalized least square algorithm of consistently handling cosmic rays.

4 GLS fitting of non-linear grouped ramps

In principle, Equation (10) is treatable using the Generalized Least Square fitting. If we remove the square bracket:

$$\frac{\sum_{n_i} c_j}{n_i} = a + b \cdot \frac{\sum_{n_i} t_j}{n_i} + \mathrm{K}_1 \cdot b^2 \frac{\sum_{n_i} t_j}{n_i} \frac{\sum_{n} t_j}{n_i} + \mathrm{K}_2 \cdot b^3 \frac{\sum_{n_i} t_j}{n_i} \frac{\sum_{n_i} t_j^{\,2}}{n_i} + \mathrm{K}_3 \cdot b^4 \frac{\sum_{n_i} t_j}{n_i} \frac{\sum_{n_i} t_j^{\,3}}{n_i} \tag{11}$$

and collect the known factors:

$$\begin{aligned}
C_i &= \frac{\sum_{n_i} c_j}{n_i} \\
\mathrm{A}_i &= \frac{\sum_{n_i} t_j}{n_i} \\
\mathrm{B}_i &= \mathrm{K}_1 \frac{\sum_{n_i} t_j}{n_i} \frac{\sum_{n_i} t_j}{n_i} \\
\Gamma_i &= \mathrm{K}_2 \frac{\sum_{n_i} t_j}{n_i} \frac{\sum_{n_i} t_j^{\,2}}{n_i} \\
\Delta_i &= \mathrm{K}_3 \frac{\sum_{n_i} t_j}{n_i} \frac{\sum_{n_i} t_j^{\,3}}{n_i}
\end{aligned} \tag{12}$$

we can write

$$C_i = a + b\mathrm{A}_i + b^2\mathrm{B}_i + b^3\Gamma_i + b^4\Delta_i \ . \tag{13}$$

This equation can be solved by a Generalized Least Square estimator for the unknown parameters a, b, b^2, b^3, and b^4, except that in this framework we are actually dealing with 5 independent parameters; in other words, one is actually solving an equation of this type:

$$C_i = a + b_1\mathrm{A}_i + b_2\mathrm{B}_i + b_3\Gamma_i + b_4\Delta_i \tag{14}$$

or, if there is no grouping:

$$c_i = a + b_1 t_i + b_2 K_1 t_i^{\,2} + b_3 K_2 t_i^{\,3} + b_4 K_3 t^4 \ , \tag{15}$$

for the unknowns parameters a, b_1, b_2, b_3, and b_4, where $b_1 = b$ the ramp slope.

The fact that

$$\begin{aligned} b_2 &= b_1^{\,2} \\ b_3 &= b_1^{\,3} \\ b_4 &= b_1^{\,4} \end{aligned} \tag{16}$$

i.e. that the parameters are correlated, is ignored unless the three Equations (16) are added to the Generalized Least Square problem as constraint equations; adding these three equations reduce the number of unknowns in the fit from 5 to 2, i.e. the a and b parameters we are looking for.

To solve simultaneously for Equations (14) and (16) within the framework of the Generalized Least Square procedure, one has to use the Lagrange multipliers. As this is a refinement of the Generalized Least Square fitting, let's briefly summarize how the basic method works.

5 Solving the GLS problem

The basic ingredients of the Generalized Least Square fit, given a linear ramp of n samples described by Equation (1), are:

1. The vector of the data:

$$C = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} \tag{17}$$

2. The vector of parameters:

$$\alpha = \begin{pmatrix} a & b \end{pmatrix} \tag{18}$$

3. The matrix of known coefficients:

$$A = \begin{pmatrix} 1 & t_1 \\ 1 & t_2 \\ \vdots & \vdots \\ 1 & t_n \end{pmatrix} \tag{19}$$

Equation (1) can now be written as $C = \alpha \otimes A$, where the $\otimes$ symbol indicates matrix multiplication (it will not be used hereafter to keep the notation lean). To solve the problem, one has to introduce a fourth matrix:

4. The covariance matrix:

$$\Sigma = \begin{pmatrix} c_1 + \sigma_{ron}^2 & c_1 & c_1 & \dots & c_1 \\ c_1 & c_2 + \sigma_{ron}^2 & c_2 & \dots & c_2 \\ c_1 & c_2 & c_3 + \sigma_{ron}^2 & \dots & c_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_1 & c_2 & c_3 & \dots & c_n + \sigma_{ron}^2 \end{pmatrix} \tag{20}$$

in the case of single frames. When frames are grouped, the covariance matrix takes a more complicated form, given e.g. by Robberto 2009.

With these ingredients, the solution for the slope and intercept is given by the matrix product:

$$\alpha = \Sigma_\alpha A^T \Sigma^{-1} C \tag{21}$$

having introduced for convenience the 2×2 matrix

$$\Sigma_\alpha = \left(A^T \Sigma^{-1} A \right)^{-1} \tag{22}$$

that actually represents the covariance (errors and cross correlation term) of the slope and intercept. The chi-square of the solution, with $n-2$ degrees of freedom, is given by

$$\chi_\alpha{}^2 = (C - A\alpha)^T \, \Sigma^{-1} (C - A\alpha) \ . \tag{23}$$

Having solved the basic problem, let's now add the non-linearity terms; dealing with 5 parameters a, b_1, b_2, b_3, and b_4 , the vector α in Equation (18) becomes:

$$\alpha \equiv \begin{pmatrix} \alpha_0 & \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \end{pmatrix} = \begin{pmatrix} a & b_1 & b_2 & b_3 & b_4 \end{pmatrix} ; \tag{24}$$

the A matrix becomes (using the symbols appropriate for the grouped ramp)

$$A = \begin{pmatrix} 1 & \mathrm{A}_1 & \mathrm{B}_1 & \Gamma_1 & \Delta_1 \\ 1 & \mathrm{A}_2 & \mathrm{B}_2 & \Gamma_2 & \Delta_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \mathrm{A}_n & \mathrm{B}_n & \Gamma_n & \Delta_n \end{pmatrix} \tag{25}$$

with elements given by Equations (12). The vector of the data (also assumed to be grouped) contains the terms given in the first of Equations (12):

$$C = \begin{pmatrix} C_1 \\ C_2 \\ \vdots \\ C_n \end{pmatrix} \tag{26}$$

For the covariance matrix, we can use the expression of Robberto (2009) ignoring for the moment the effect of non-linearity in the Poisson noise (this refinement will be the subject of a future study). Equations (21)-(23) then provide the results, that we can indicate with $\bar{\alpha}$, $W_{\bar{\alpha}}$ and $\chi^2{}_{\bar{\alpha}}$. This Generalized Least Square solution with 5 parameters left unconstrained will generally returns results quite different from those derived with the basic iterative, constrained solution, i.e. the 3 extra coefficients that have been derived can be far away from being the square, the cube and the fourth power of the slope. We will see soon that both solutions, iterative and unconstrained Generalized Least Square, are necessaries to solve the constrained problem.

To add the constraints we start with the set of 3 constraint equations (16), that can be written as

$$\begin{aligned} \alpha_2 - \alpha_1^{\,2} &= 0 \\ \alpha_3 - \alpha_1^{\,3} &= 0 \\ \alpha_4 - \alpha_1^{\,4} &= 0 \end{aligned} \tag{27}$$

i.e. 3 non-linear equations of the form $H(\alpha)=0$. One can expand them around a set of initial good values (i.e., those found by the iterative solution, that therefore plays here a key role) that we will indicate as $\hat{\alpha}$:

$$H(\alpha) = H(\hat{\alpha}) + \left(\frac{\partial H}{\partial \alpha}\right)_{\alpha=\hat{\alpha}} (\alpha - \hat{\alpha}) \equiv \mathrm{E} + \mathbf{D}\Delta\alpha = 0 \tag{28}$$

The E vector goes to zero, since the iterative solution, forcing the higher order terms of the polynomial to be exactly the powers of the slope parameter b, are identically satisfied: $H(\hat{\alpha})=0$. The term containing the derivatives can be immediately calculated:

$$\mathbf{D}\Delta\alpha = \begin{pmatrix} 0 & 2\alpha_1 & -1 & 0 & 0 \\ 0 & 3\alpha_1^{\,2} & 0 & -1 & 0 \\ 0 & 4\alpha_1^{\,3} & 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} \alpha_0 - \hat{\alpha}_0 \\ \alpha_1 - \hat{\alpha}_1 \\ \alpha_2 - \hat{\alpha}_2 \\ \alpha_3 - \hat{\alpha}_3 \\ \alpha_4 - \hat{\alpha}_4 \end{pmatrix}. \tag{29}$$

Finally we apply the Lagrange multipliers. The idea consists in adding a new term to the expression of the chi-square, accounting for the constraint equations, with a multiplicative factor λ that "pulls" the solution away from the unconstrained case. In other words, the constraints are applied after solving the unconstrained problem (and here is why we need to solve also for the unconstrained problem).

In the presence of constraints, the expression for the chi-square (Eq. (23) becomes:

$$\chi^2 = (C - A\alpha)^T \Sigma^{-1} (C - A\alpha) + 2\lambda^T (\mathrm{E} + \mathbf{D}\Delta\alpha) \tag{30}$$

with λ a vector of dimension 3. This expression can be rewritten as

$$\chi^2 = (C - A\bar{\alpha})^T \Sigma^{-1} (C - A\bar{\alpha}) + (\alpha - \bar{\alpha})^T \Sigma_{\bar{\alpha}}^{-1} (\alpha - \bar{\alpha}) + 2\lambda^T (\mathrm{E} + \mathbf{D}\Delta\alpha) \tag{31}$$

Here the first term on the RHS is just the chi-square for the unconstrained fit $\chi^2_{\bar{\alpha}}$, derived earlier. If we set to zero the partial derivatives of the χ^2 wrt each α parameter and λ Lagrange multiplier, we obtain the set of equations:

$$\begin{aligned} \Sigma_{\bar{\alpha}}^{-1}(\alpha-\bar{\alpha})+\mathbf{D}^T\lambda &= 0 \\ E+\mathbf{D}\alpha &= 0 \end{aligned} \tag{32}$$

that can be easily solved. For example, from the first one, we have immediately the solution for the α parameters:

$$\alpha=\bar{\alpha}-\Sigma_{\bar{\alpha}}\mathbf{D}^T\lambda \tag{33}$$

while for the Lagrange multipliers it is:

$$\begin{aligned} &\alpha-\bar{\alpha}=\Sigma_{\bar{\alpha}}\mathbf{D}^T\lambda \Rightarrow \\ &\mathbf{D}\alpha-\mathbf{D}\bar{\alpha}=\mathbf{D}\Sigma_{\bar{\alpha}}\mathbf{D}^T\lambda \Rightarrow \\ &\cancel{\mathbf{E}+\mathbf{D}\alpha}-(\mathbf{E}+\mathbf{D}\bar{\alpha})=\mathbf{D}\Sigma_{\bar{\alpha}}\mathbf{D}^T\lambda \end{aligned} \tag{34}$$

providing

$$\lambda=\left(\mathbf{D}\Sigma_{\bar{\alpha}}\mathbf{D}^T\right)^{-1}(E+\mathbf{D}\bar{\alpha})\ . \tag{35}$$

Similar calculations can be performed for the covariance and χ^2 terms, obtaining eventually the set of final results:

$$\begin{aligned} &\alpha=\bar{\alpha}-\Sigma_{\bar{\alpha}}\mathbf{D}^T\lambda \\ &\Sigma_\alpha=\Sigma_{\bar{\alpha}}-\Sigma_{\bar{\alpha}}\mathbf{D}^T\Sigma_\lambda\mathbf{D}\Sigma_{\bar{\alpha}} \\ &\chi^2=\lambda^T\Sigma_{\mathbf{D}}^{-1}\lambda=(E+\mathbf{D}\bar{\alpha})^T\,\Sigma_{\mathbf{D}}(E+\mathbf{D}\bar{\alpha})=\lambda^T(E+\mathbf{D}\bar{\alpha}) \end{aligned} \tag{36}$$

with

$$\begin{aligned} &\lambda=\Sigma_D(E+\mathbf{D}\bar{\alpha}) \\ &\Sigma_D=\Sigma_\lambda=\left(\mathbf{D}\Sigma_{\bar{\alpha}}\mathbf{D}^T\right)^{-1} \end{aligned} \tag{37}$$

Note that the auxiliary matrix Σ_D is also the covariance matrix of the Lagrange multipliers.

In summary, adding the constraints requires specifying the $\mathbf{D}$ and $\mathbf{E}$ matrices, but $\mathbf{E}=0$ in our case, so only the 3×3 auxiliary matrix Σ_D has to be calculated.

6 Implementation in IDL

Before describing the results provided by this algorithm, it may be useful to illustrate the key steps of the algorithm, that can be implemented in an extremely compact form using matrix operators. I will show here its implementation in plain IDL code, removing some non-essential line added to make the code robust against bad pixels.

In group mode, the vectors of data and time can be built from a basic ramp taken in RAPID mode with code of this type:

```
ramp_gr = FLTARR(Nframes)
time_gr = FLTARR(Nframes)
for i = 0,Nframes-1 do begin
  ramp_gr[i] = MEAN(ramp(FIX([(i*(Ngrouped+skip)),(i*(Ngrouped+skip))+(Ngrouped-1)])))
  time_gr[i] = MEAN(time(FIX([(i*(Ngrouped+skip)),(i*(Ngrouped+skip))+(Ngrouped-1)])))
endfor
```

where the FOR loop implements the first two Equations in (12). Of course, the construction of ramp_gr is omitted when the data come already averaged by the readout system.

The first step is to linearize the ramp following the Kappa method, using a basic iterative procedure to derive an approximate solution.

```
;start with an initial estimate of the linearized counts
A1t = (ramp_gr[2]-ramp_gr[1])/(time_gr[2]-time_gr[1])*time_gr[First_frame:*]
;..do the iteration solve (doing 10 iterations is an overkill, and lazy programming…)
    FOR ix=0,10 do BEGIN
        DEN = (1+K3[0]*A1T+K3[1]*A1T^2+K3[2]*A1T^3)
        A0 = (LINFIT(time_gr[FIRST_frame:LAST_frame],(ramp_gr[FIRST_frame:LAST_frame]-
A1T*DEN)))[0]
        A1T = (ramp_gr[FIRST_frame:LAST_frame]-A0)/DEN
        A1 = MEAN(A1T/time_gr[FIRST_frame:last_frame])
    ENDFOR
```

The next step is to specify the other terms of Equations (12). The summation terms in the B_i , Γ_i and Δ_i equations are given by

```
T1_gr=time_gr
T2_gr=DBLARR(Npoints_gr)
T3_gr=DBLARR(Npoints_gr)
for i = 0,Last_frame_gr do begin
 T2_gr[i] = mean((time(FIX([(i*(Ngrouped+skip)),(i*(Ngrouped+skip))+(Ngrouped-1)])))^2)
 T3_gr[i] = mean((time(FIX([(i*(Ngrouped+skip)),(i*(Ngrouped+skip))+(Ngrouped-1)])))^3)
endfor
```

The iterative solution of Equation (13), written in the form of the analog Equation (5), is therefore found

```
bt_gr = (ramp_gr[2]-ramp_gr[1])/(time_gr[2]-time_gr[1])*time_gr[First_frame:*]; 1st cut
to linear ramp
```

```
b_gr = MEAN(bt_gr[FIRST_frame:last_frame_GR-1]/time_gr[FIRST_frame:last_frame_GR-1]) ;
b2t2_gr = b_gr^2*T2_gr
b2t3_gr = b_gr^3*T3_gr
FOR ix=0,10 do BEGIN  ;10 iterations, normally 3 or four are enough to converge
  DEN_gr = (1 + K3[0]*bt_gr + K3[1]*b2t2_gr + K3[2]*b3t3_gr)
  a0_gr = (LINFIT(time_gr[FIRST_frame:LAST_frame_GR],(ramp_gr[FIRST_frame:LAST_frame_GR]-
bt_gr*DEN_gr)))[0] ; intercept
  bt_gr = (ramp_gr[FIRST_frame:LAST_frame_GR]-a0_GR)/DEN_gr
  b_gr = MEAN(BT_gr/(time_gr[FIRST_frame:last_frame_GR]))
  b2t2_gr = b_gr^2*T2_gr
  b3t3_gr = b_gr^3*T3_gr
ENDFOR
alpha_hat=fltarr(5)
alpha_hat[0]=a0_gr  ;fit parameters in output
alpha_hat[1]=b_gr
alpha_hat[2]=b_gr^2
alpha_hat[3]=b_gr^3
alpha_hat[4]=b_gr^4
```

Having calculated an approximate value for the intercept (ab[0]) and slope (ab[1]), with polynomial terms given by the powers of the slope, we can now find the solution for the Generalized Least Square case, limited to the unconstrained case.

The matrix A of parameters, defined in Eq. (25), is given by

```
;T MATRIX
AA = fltarr(5,N) ;AA=fltarr(2,N)
AA[0,*]=1.
AA[1,*]=time_gr
AA[2,*]=Coeffs[0]*time_gr^2
AA[3,*]=Coeffs[1]*time_gr*t2_gr
AA[4,*]=Coeffs[2]*time_gr*t3_gr
```

The covariance matrix, Equation (20), is given by

```
;size the COVARIANCE MATRIX
SIGMA = DBLarr(N,N)
;
b = ab[1] ; adopt the preliminary estimate of signal rate
;Calculate the first row/col of the covariance matrix out of the loop, just for
convenience
;(M=nr. Frames/group; ft=readout time, G=gain… see papers on ramp covariance)
SIGMA[0,0] =  b*tf*(M+1)/(2.*M) + b*tf*1/(3.*M)*(M-1)*(M-2) + RON^2/M +
(G*M/SQRT(12))^2
SIGMA[0,1:*] = (M+1)/2.*b*tf
SIGMA[1:*,0] = (M+1)/2.*b*tf
```

```
;
;and the other terms of the matrix...
for i=1,N-1 do begin
  SIGMA[i,i] = b*i*tg + b*tf*(M+1)/(2.*M) + b*tf*1/(3.*M)*(M-1)*(M-2) + RON^2/M +
(G*M/SQRT(12))^2
  IF I LT N-1 THEN BEGIN & $;diagonal terms
    SIGMA[I,I+1:N-1] = REPLICATE(b*i*tg+b*(m+1)/2.*tf,N-(i+1))
     SIGMA[I+1:N-1,I] = REPLICATE(b*i*tg+b*(m+1)/2.*tf,N-(i+1))
  ENDIF
endfor
```

with its inverse:

```
;INVERSE COVARIANCE (AKA WEIGHTS if DIAGONAL)
W_Y_1 = INVERT(SIGMA)
```

The covariance matrix of the ramp parameters, Equation (22), is therefore

and the vector of parameters, Equation(21), is

Here are the main results derived from the unconstrained GLS fit:

plus three more parameters, index 2-4, that we have left "unleashed" to provide the best possible quartic fit to the data ramp. Again, in general they will not be the square, the cube and the fourth power of the slope. For this we need to add the constraint equation. Just notice that, unlike the two alpha_hat parameters derived using the iterative solution, the five alpha_bar parameters carry a full covariance matrix giving us, for example, the errors on the intercept and the slope:

```
 ;W_A MATRIX
W_alpha_bar=INVERT(TRANSPOSE(AA)##W_Y_1##AA)
alpha_bar=W_Abar##TRANSPOSE(AA)##W_Y_1##C_data
intercept = alpha_bar[0]
slope = alpha_bar[1];
delta_intercept=SQRT(W_alpha_bar[0,0])
delta_slope=SQRT(W_alpha_bar[1,1])
b = alpha_hat[1]  ;we start from the iterative, "almost good" hat solution.
D = dblarr(5,3)
D[0,*] = 0.  ;first column, intercept, all zero
D[1,*] = [-2*b , -3*b^2, -4*b^3] ;second column, slope
D[2,0] = 1   ;diagonal terms…
```

We must now add the constraints; first, we define the matrix $\mathbf{D}$ and the vector $\Delta\alpha$, according to Equation (29) :

```
D[3,1] = 1
D[4,2] = 1
;
```

```
delta_alpha = fltarr(5)
delta_alpha = alpha_bar - alpha_hat
delta_alpha = TRANSPOSE(alpha_bar - alpha_hat)
```

Then we calculate the auxiliary matrix Σ_D defined in Equation (37)

```
V_D = INVERT(D##W_alpha_bar##TRANSPOSE(D)) ;the covariance matrix of the lagrange
multipliers
```

And finally we solve the problem, Equations (36) and (37), finding the Lagrange multipliers,

```
lambda=V_D##(D##delta_alpha)
```

the GLS fit parameters, with costraints:

```
alpha=alpha_bar-W_alpha_bar##TRANSPOSE(D)##lambda
```

the covariance of the GLS fit parameters:

```
W_alpha = W_alpha_bar - W_alpha_bar##(TRANSPOSE(D)##V_D##D)##W_alpha_bar
```

and the additional χ^2 term, relative to the Lagrange multipliers.:

```
chi2 = TRANSPOSE(lambda)##(D##delta_alpha)
```

The covariance of the Lagrange multipliers was also given by the auxiliary matrix Σ_D.

7 Results

To test the algorithm and give a first illustration of the method, we use a single ramp obtained with an early NIRCam detector (SCA1) during the FM1 tests at Lockheed Martin. The ramp, originally taken for linearity test, contains 80 frames taken in RAPID mode. The non-linearity coefficients, both Kappa and Phi, have been previously estimated by averaging ramps of the same family. The ramp has not been corrected for dark current, assuming the effect is negligible for ramps approaching saturation. We randomly take pixel (100,100) for this analysis.

Figure 1 shows the data ramp with the results obtained applying the iterative method (Equation (5)). The curves labeled with t, t^2, t^3 and t^4 represent the functions

$$\begin{aligned}
\text{t:}\quad & c_i = a + b\cdot t_i \\
\text{t^2:}\quad & c_i = a + b\cdot t_i \cdot \left[1 + \mathrm{K}_1\left(b\cdot t_i\right)\right] \\
\text{t^3:}\quad & c_i = a + b\cdot t_i \cdot \left[1 + \mathrm{K}_1\left(b\cdot t_i\right) + \mathrm{K}_2\left(b\cdot t_i\right)^2\right] \\
\text{t^4:}\quad & c_i = a + b\cdot t_i \cdot \left[1 + \mathrm{K}_1\left(b\cdot t_i\right) + \mathrm{K}_2\left(b\cdot t_i\right)^2 + \mathrm{K}_3\left(b\cdot t_i\right)^3\right]
\end{aligned} \tag{38}$$

The first function is the linearized signal, the others show how the intermediate terms contribute to reproduce the data values. Of course, having used the iterative algorithm, we do not have a reliable estimate of the errors.

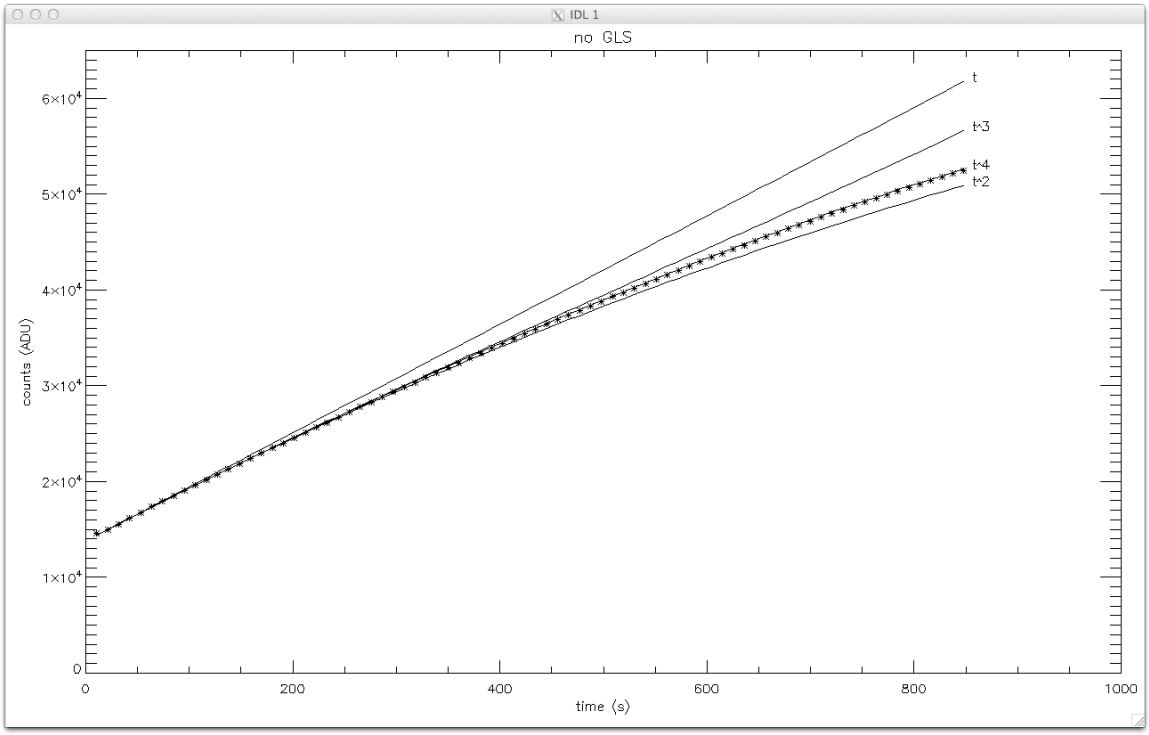

Figure 1 Example of non-linearity correction for a ramp taken in RAPID mode (80 frames) using the iterative algorithm of Robberto 2011. The asterisks mark the original raw data, the curves the different terms in Equation (5).

The same data can be treated using the Generalized Least Square method, leaving for the moment the parameters unconstrained. In the plot shown in Figure 2 the curves represent the contribution of the various terms of Equation (15):

$$\begin{aligned}
\text{t:} \quad & c_i = a + b_1 t_i \\
\text{t^2:} \quad & c_i = a + b_1 t_i + b_2 K_1 t_i^2 \\
\text{t^3:} \quad & c_i = a + b_1 t_i + b_2 K_1 t_i^2 + b_3 K_2 t_i^3 \\
\text{t^4:} \quad & c_i = a + b_1 t_i + b_2 K_1 t_i^2 + b_3 K_2 t_i^3 + b_4 K_3 t^4
\end{aligned} \tag{39}$$

Notice the differences between Figure 2 and Figure 1 e.g. in the location of the t^2 curve; different combinations of coefficients can provide a good reconstruction (assessed "by eye", at the moment) of the signal. The difference here is that the parameters are not bounded to be powers of the slope; the Generalized Least Square method returns a full 5×5 covariance matrix for the 5 coefficients a, b_1, b_2, b_3, and b_4.

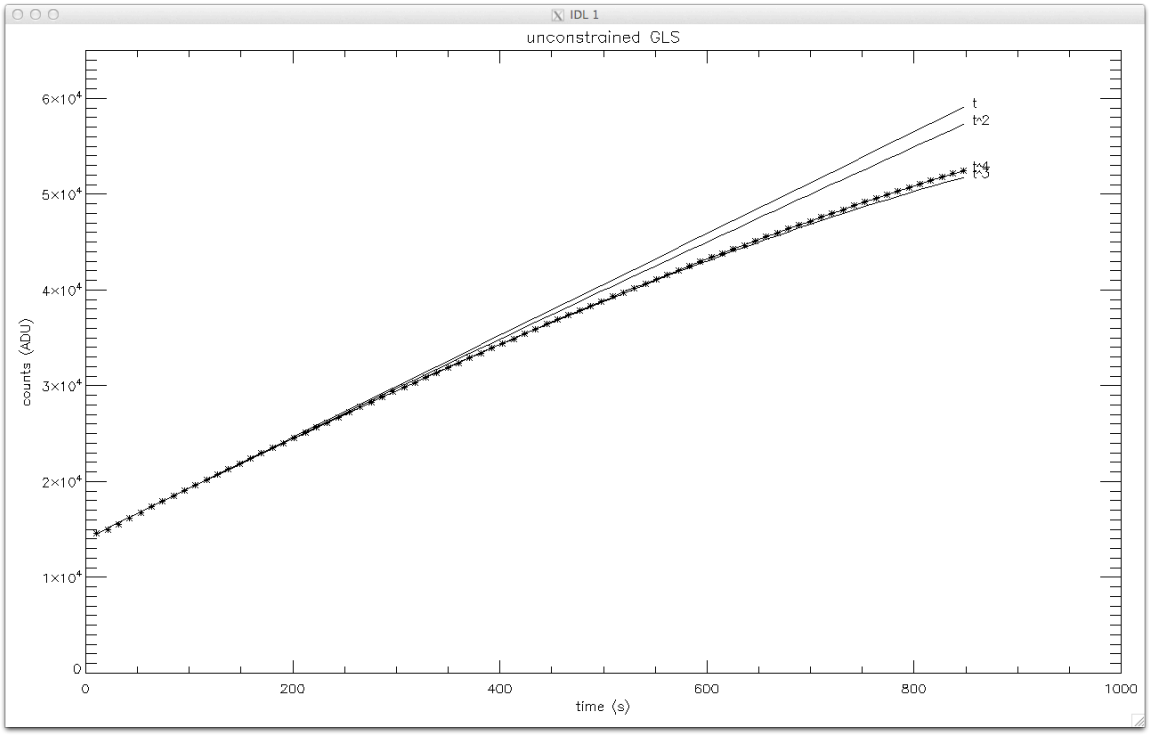

Figure 2 Same as Figure 1, for the reconstruction obtained using the unconstrained GLS method.

The last step is to solve Equation (5), with constrained parameters. We obtain in this the results shown in Figure 3. It is almost identical to Figure 1, not surprising since we expanded Equation (28) in the vicinity of that solution. This time, however, we have also the covariance terms and the chi-2; the covariance, for example, is given by the following Table 1.

Table 1 Covariance matrix for the 5 parameters derived using the constrained GLS method

682.15486	-1.8985704	-214.79007	-18224.807	-1374545.7
-1.8985704	0.13459220	15.226756	1291.9810	97443.387
-214.79007	15.226756	1722.6414	146165.08	11024017.
-18224.807	1291.9810	146165.08	12402019.	9.3538114e+08
-1374545.7	97443.387	11024017.	9.3538114e+08	7.0548021e+10

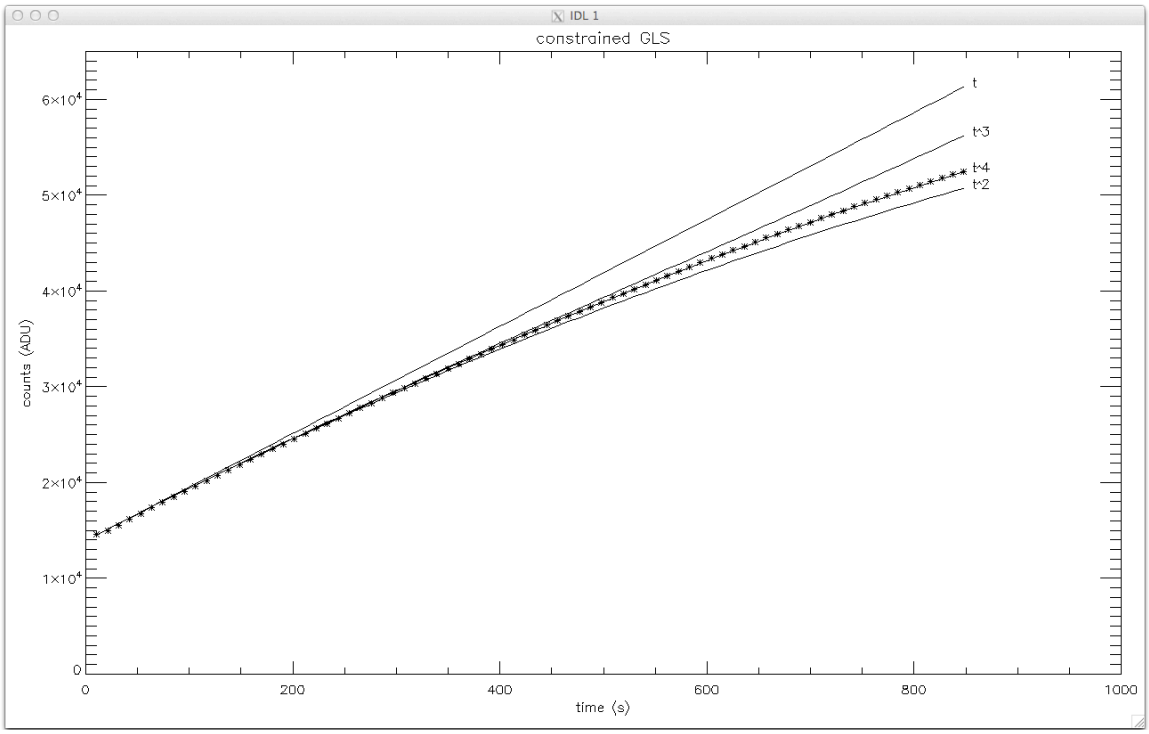

Figure 3 Same as Figure 1 and 2, for the reconstruction obtained with the constrained GLS method.

Table 2 lists the coefficients in the 3 cases; the similarity between the first iterative solution and the Generalized Least Square with constraints is evident.

Table 2 Summary of the parameters obtained with the three methods, for the data shown in Figures 1 - 3.

	iterative	GLS unconstrained	GLS + constraint
intercept	13781.8	13998.7	13950.5
Slope	56.56	53.02	55.83
b_1	3199.8	309.02	3116.5
b_2	180997	-220467	173930
b_3	10238350	-3956197	9705332

A direct check can reveal that the three *b* parameters for the iterative method are *exactly* the second, third and fourth power of the slope; for the unconstrained Generalize Least

Square method they have nothing to do with slope; for the constrained Generalized Least Square they are *approximately*, but not exactly, the second, third and fourth power of the slope, indicating that the Lagrange multipliers have done their job of constraining the results without collapsing to the exact values, due to the presence of uncertainties. If we take the square root of the first two diagonal terms of the Covariance Matrix, we have

$$a = 13950 \pm 26$$
$$b = 55.83 \pm 0.37$$

for the full ramp sampled in RAPID mode. If we now group the data, for example using a MEDIUM8 scheme (groups of 8 frames, skip 2), we have the three plots shown in Figure 4-6 , analog to Figure 1-3:

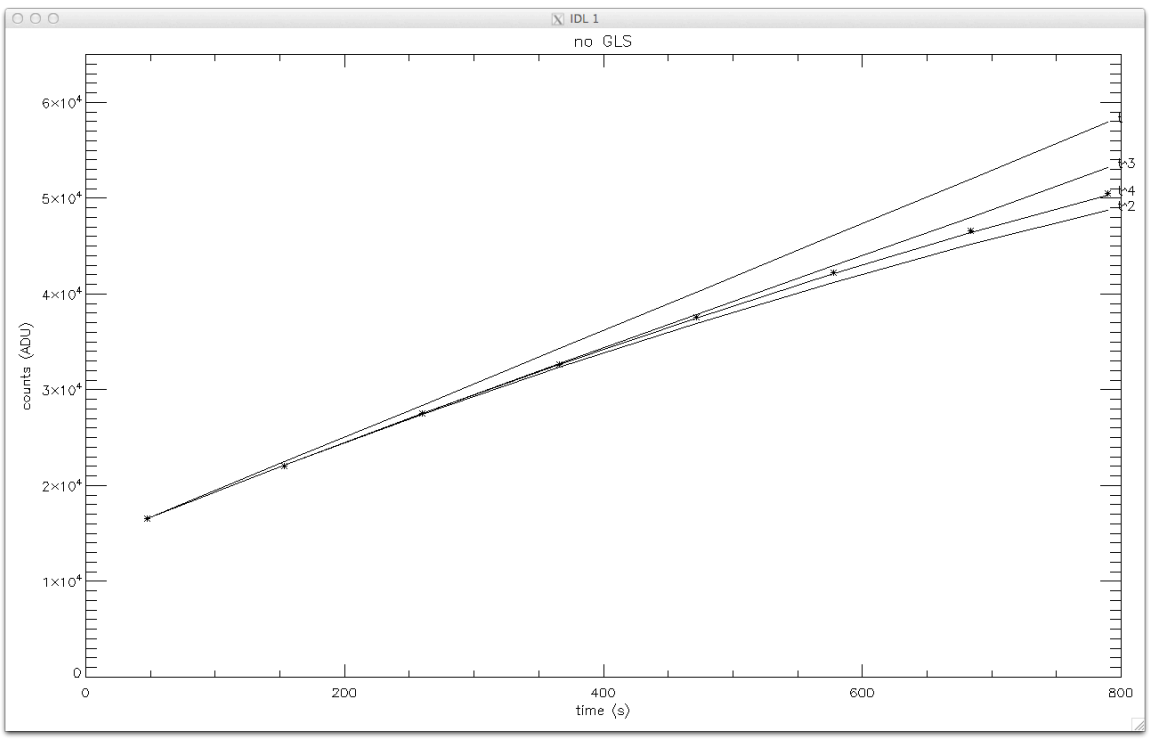

Figure 4 Same as Figure 1, iterative method, after converting the data to MEDIUM 8 mode.

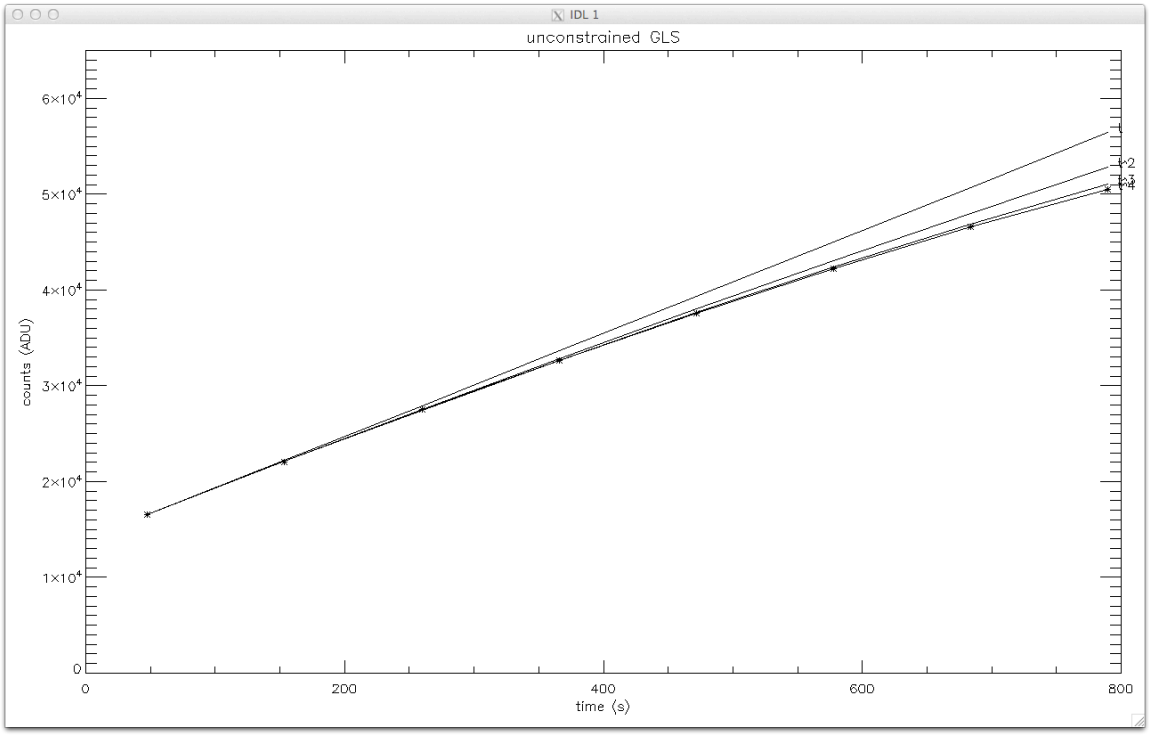

Figure 5 Same as Figure 2, unconstrained GLS, after converting the data to MEDIUM 8 mode.

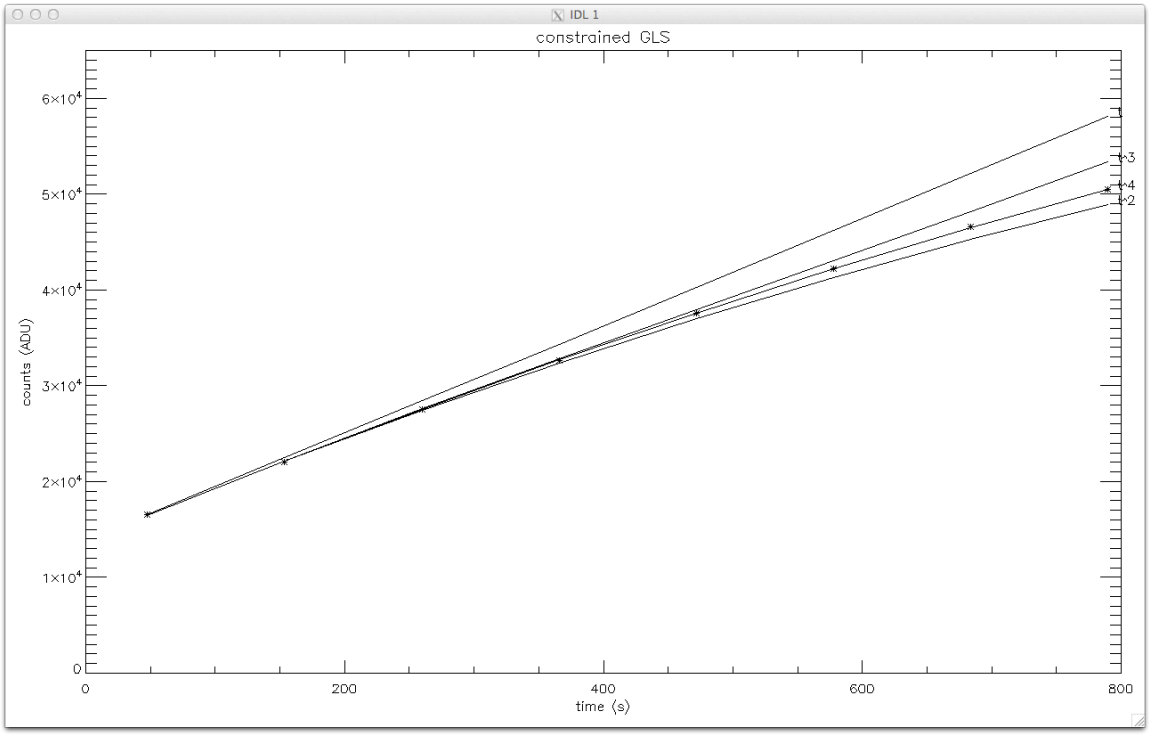

Figure 6 Same as Figure 3, constrained GLS, after converting the data to MEDIUM 8 mode.

and the results shown in Table 3

Table 3 Same as Table 2 for the ramp converted in MEDIUM8 Mode

	iterative	GLS unconstrained	GLS + constraint
intercept	13923.1	13976.8	13863.7
Slope	55.68	53.74	56.04
b_1	3100.2	1233.4	3140.4
b_2	172622	-69769	175972
b_3	9611579	1805736	9860344

Again, one can notice the large discrepancies between the parameters derived in the case of unconstrained fit (central column) and the two other cases.

If we repeat this exercise sampling the ramp according to the 9 modes of NIRCam, we have the following table summarizing the results for the slope, also illustrated in Figure 7.

Table 4 Summary of the slope parameter determined with the 3 methods, sampling the same 80 samples ramp with the 9 different NIRCam modes. See next Section for an explanation of the last 2 "pseudo" Deep modes.

mode	Rapid Ng=1 Ns=0	Bright1 Ng=1 Ns=1	Bright2 Ng=2 Ns=0	Shallow2 Ng=2 Ns=3	Shallow4 Ng=4 Ns=1	Medium2 Ng=2 Ns=8	Medium8 Ng=8 Ns=2	Pseudo Deep2 Ng=2 Ns=16	Pseudo Deep8 Ng=8 Ns=10
Iterative	56.56	56.55	56.40	55.97	55.93	55.77	55.68	54.96	55.14
GLS unconstrained	53.02	53.05	54.69	54.67	54.44	54.87	53.74	55.13	54.42
GLS + constraint	55.83	55.82	56.05	56.03	56.02	56.06	56.04	56.04	56.01

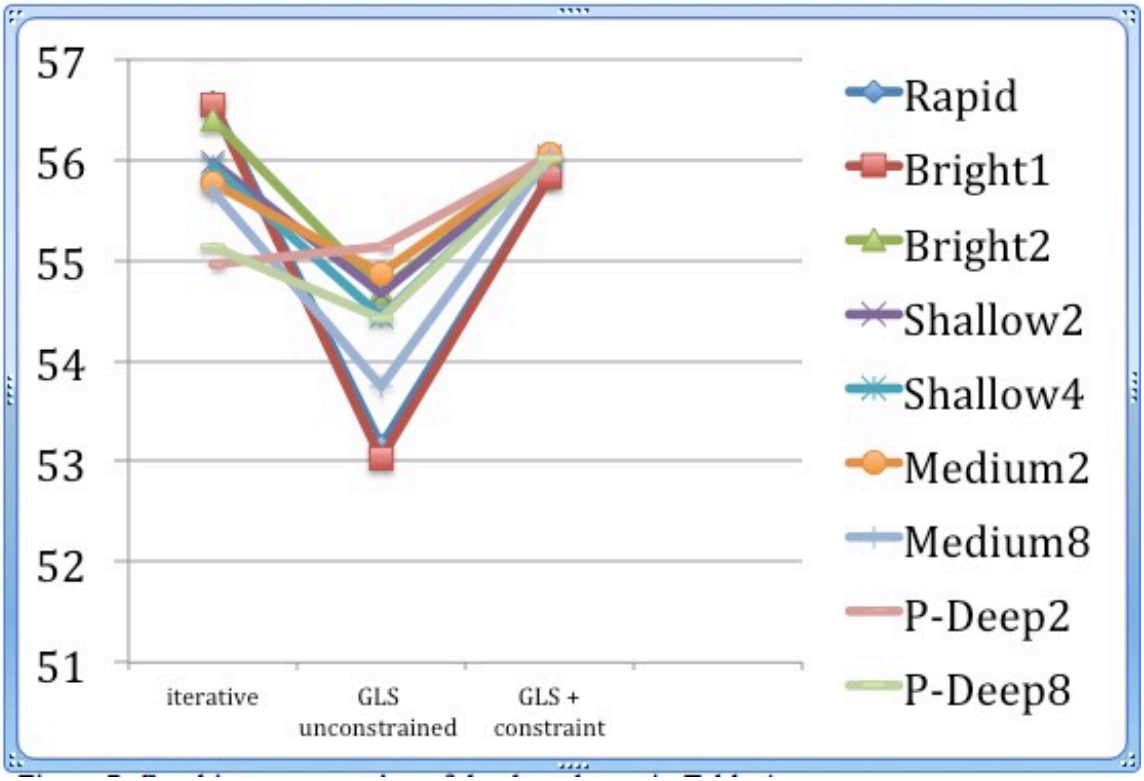

Figure 7 Graphic representation of the data shown in Table 4.

Figure 7 provides a clear illustration of the superiority of the constrained Generalized Least Square fitting vs. the two other methods: the calculated slope, regardless of how the ramp has been sampled, is basically the same, whereas the iterative and unconstrained Generalized Least Square method return values that have a spread of a few percent.

The intercept (Figure 8) shows similar trends, with the exception of the two values for the RAPID and BRIGHT1 mode that do not match the others. A careful look at Table 1 shows that the slopes in these two modes are slightly higher; it is intuitive that higher slopes correlate with lower intercept, for the same ramp. The reason for this discrepancy is unclear; it could be attributed to the relatively larger contribution of systematic noise at the beginning of the ramps, well sampled in these two readout modes.

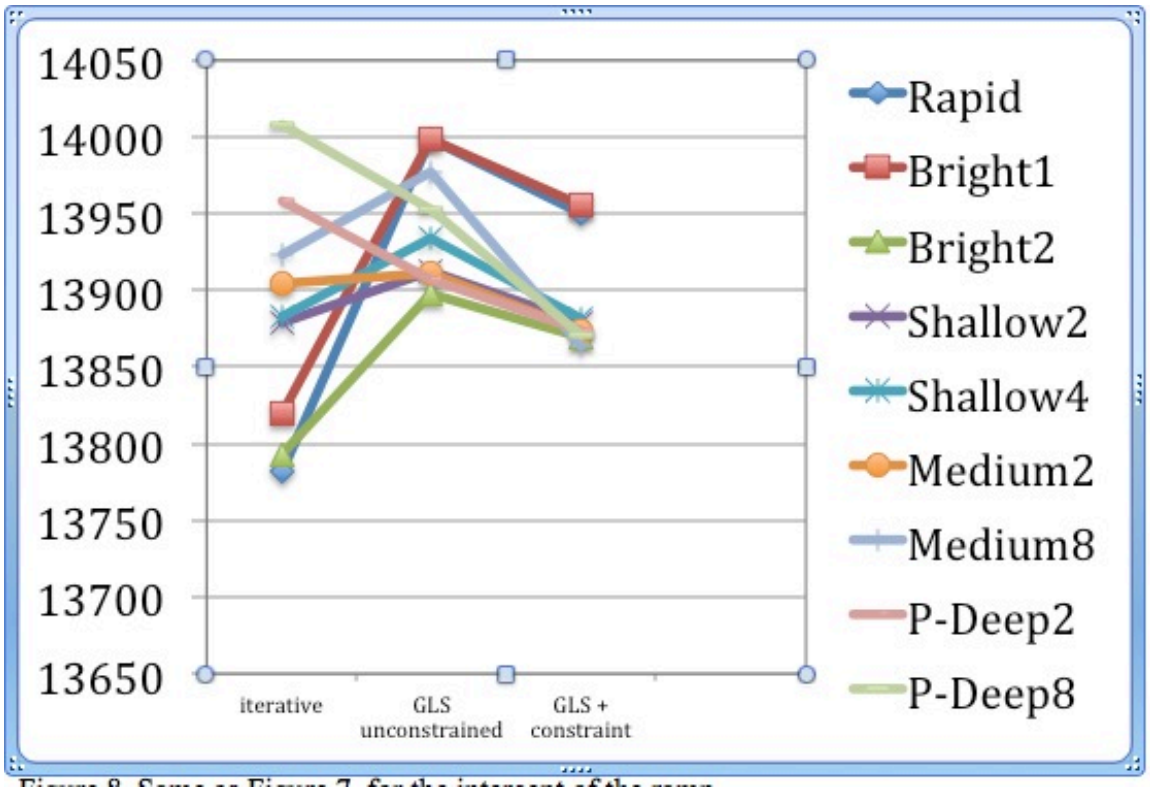

Figure 8 Same as Figure 7, for the intercept of the ramp.

8 Limitations

The main limitation of the constrained Generalized Least Square method is that it needs to solve first the unconstrained Generalized Least Square problem, which means to calculate a number of parameters equal to the order of the correction polynomial plus one; in our case, with a fourth degree polynomial we search for 5 parameters. This means that at least 5 data points are needed for the method to work. The "pseudo" DEEP ramps in Table 3 were in fact obtained by reducing the number of skipped samples, that in true DEEP modes are 18 and 12, to 16 and 10, respectively; this in order to create 5 groups from the original ramp of 80 samples.

If one has less than five samples, one can either:

a) use a lower order polynomial; different coefficients have to be derived for this case.
b) use the traditional KAPPA method for correcting for non-linearity, and then the basic GLS method on linearized data. Unfortunately, in the case of grouped frames, this method provides wrong results.
c) use the iterative method; it works for both ungrouped and grouped data and provides quite accurate results, but one has no covariance or chi-2.

Other approaches may be possible. The problem of ramp fitting (and cosmic ray search) in the case of 5 or less frames will be treated more in detail in a future study.

9 Conclusions

The Generalized Least Square method can be extended to solve for non-linear ramps that have been sampled with grouped frames. In this report we have illustrated the solution, based on the use of Lagrange multipliers, and provided the key elements of the IDL code implementing the algorithm. An example of the solutions shows that the slopes derived by the method are very consistent regardless of the grouping scheme adopted. A minimum number of frames related to the order of the correction polynomial is needed for the method to work.

10 References

Anderson, R.E., & Gordon, K.D. 2011, Cosmic-Ray Detection for Nondestructive Read Ramps, PASP, 123, 1237

Martel, A. R. 2012, "The Detection of Outliers in JWST Nondestructive Ramps with the Extreme Studentized Deviate (ESD) Test", JWST-STScI-002982

Rauscher, Bernard J.; Fox, Ori; Ferruit, Pierre; Hill, Robert J.; Waczynski, Augustyn; Wen, Yiting; Xia-Serafino, Wei; Mott, Brent; Alexander, David; Brambora, Clifford K.; Derro, Rebecca; Engler, Chuck; Garrison, Matthew B.; Johnson, Thomas; Manthripragada, Sridhar S.; Marsh, James M.; Marshall, Cheryl; Martineau, Robert J.; Shakoorzadeh, Kamdin B.; Wilson, Donna; Roher, Wayne D.; Smith, Miles; Cabelli, Craig; Garnett, James; Loose, Markus; Wong-Anglin, Selmer; Zandian, Majid; Cheng, Edward; Ellis, Timothy; Howe, Bryan; Jurado, Miriam; Lee, Ginn; Nieznanski, John; Wallis, Peter; York, James; Regan, Michael W.; Hall, Donald N. B.; Hodapp, Klaus W.; Böker, Torsten; De Marchi, Guido; Jakobsen, Peter; Strada, Paolo, 2007 PASP 119, 768

Robberto, M. 2008, "On the Optimal Strategy to fit MULTIACCUM ramps in the presence of cosmic rays", JWST-STScI-001490

Robberto, M. 2009, "Derivation of the correct noise equation for General MULTIACCUM readout", JWST-STScI-001853

Robberto, M. 2011, "An improved algorithm for the correction of IR detector non linearity", JWST-STScI-002163

Robberto, M. 2011b " Implementation of a new algorithm for the correction of non-linearity in JWST detectors", JWST-STScI-002344

Robberto, M. 2012 "On the Linearity Correction of IR Ramps in the Case of Grouped Frames", JWST-STScI-002588

Robberto, M. 2013a, "A Generalized Least Square algorithm to correctly perform linear ramp fitting", JWST-STScI-003193

Robberto, M. 2013b, "A generalized least square algorithm to process infrared data taken in non-destructive readout mode", Proc. SPIE 9143, 9143Z-1