About Transients

Technical Report

Version 1.1

C. Delattre
CEA, DSM/DAPNIA, CE-Saclay, F–91191 Gif–sur–Yvette, France
delattre@ariane.saclay.cea.fr

April 1996

1 Introduction

The aim of this technical report is to write down and explain thoroughly everything which has been done over the thema TRANSIENTS FITTING CORRECTION within the ISOCAM team. An overview of the numerical methods used in practice, as well as compared and discussed, is given.

Working Set of Data We used the raster 4×4 and 10×10 of M51 and the raster 4×2 of the Antenna.

Nota All results, plots, and images have been performed from real ISO data.

2 Transient Fitting with SAp Model

2.1 Why a new model: the SAp model

This model which is not far different from the IAS and IPAC ones takes into account the very low transition (in time) of a pixel after a strong exposure (like the companion of M51). Real data show that this kind of downward transition are badly fitted by the above models. Why? Because the curve can be splitted into two parts:

– a first part where data are quickly decreasing and they can be fitted by an downward exponential;

– a second part where data are decreasing very slowly but which cannot be seen as the tail of the decreasing exponential of the first part.

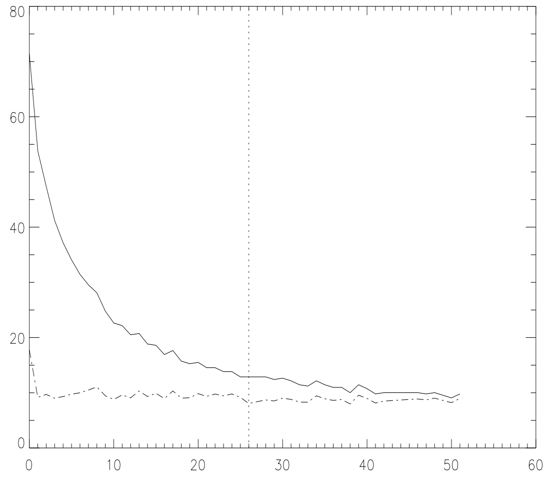


FIG. 1 -

That's why we introduce a second decreasing exponential with a different relaxation time constant as in the decreasing model of the first part of the transition. The SAp model is given by the following functions:

- **Downward Transition**

$$J_d(t) \;=\; J_\infty + (J_0 - J_\infty).\beta.exp(-\frac{t}{\tau}) + \beta'.exp(-\frac{t}{\tau'})$$

  where $\tau' >> \tau$;

- **Upward Transition**

$$J_u(t) \;=\; J_\infty - (J_\infty - J_0).\beta.exp(-\frac{t}{\tau})$$

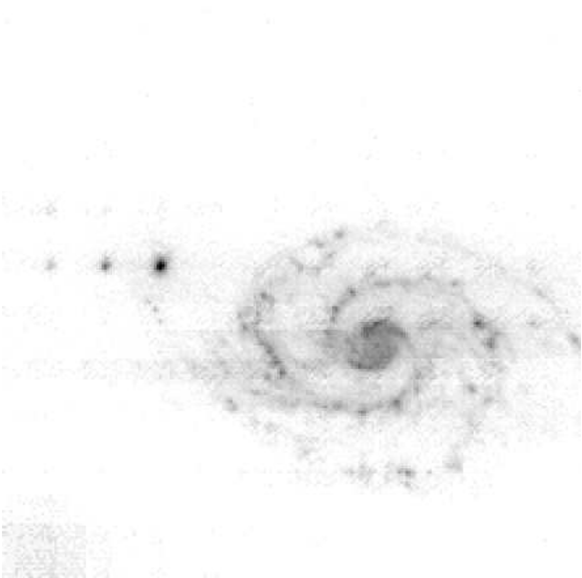
the low decrease of transient is modeled by a second larger time constant τ'. The SAp model has 6 parameters: $J_0,\ J_\infty,\ \beta,\ \tau,\ \beta',\ \tau'$.

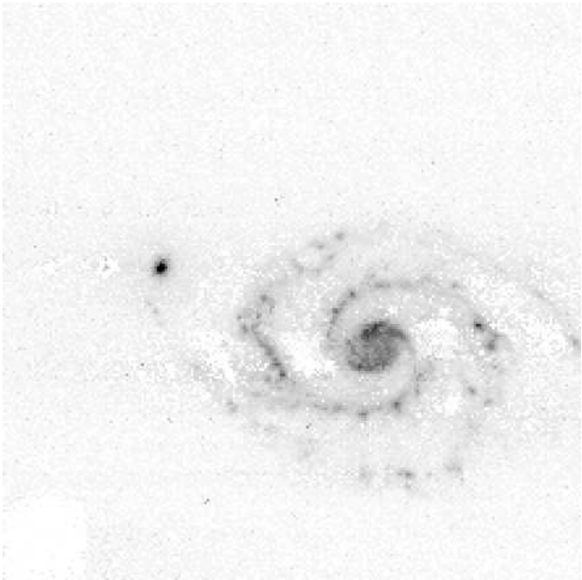
We have to keep in mind that three goals are to be achieved when using a model for fitting data. First the model should lead to a complete fit of the data, whatever is the number of points you are using. Secondly it should give the real stabilized value J_∞ when you have "enough" points. And it should allow us to work only with the first points to guess the real stabilized value J_∞. As far as the first and second points are concerned, the SAp model turns out to be very good, but if you have few points whose value is quite far from the stabilized value, the fit wouldn't give the expected value for the J_∞ parameter when you have a strong downward transient ; however the SAp model will yield to a best estimate of the J_∞ parameter than IAS or IPAC models (they only have one time–constant).

2.2 First Result

Here is the first comparison between the SAp model and the IAS model for downward transients. The effect of low decrease can be seen in the raster 10×10 of M51: there are four "ghosts" due to the *Companion* of M51(see figure 2.2).

Here are the results of transition corrections with the SAp model ; it was tried on the raster 10×10 of M51. On the left you have the raster before correcting transients ; on the right you have the raster after correction. As one can see, the SAp model leads to quite good results(no ghosts in the final image whereas there were four ghosts before treating M51).





3 Parameters Estimation

3.1 Introduction

Suppose you have a finite set of observations $Y = (y_1, \ldots, y_n)$ which represents values of a deterministic model m at the known time sequence $X = (x_1, \ldots, x_n)$, plus an additive noise $R = (r_1, \ldots, r_n)$. We shall assume that the deterministic model depends on a set of parameters represented by the unknown vector $\vec{\theta}$:

$$y_i \;=\; m(x_i; \vec{\theta}) + r_i, \qquad 1 \leq i \leq n$$

If you want to estimate the parameters vector $\vec{\theta}$, it is usual in Signal Processing to use a **L.M.S.**(which stands for Least Mean Square) algorithm, that is to minimize the sum of the square residuals. This is the famous *Least Square Criterion*:

$$\min_{\vec{\theta} \in \Theta} \|Y - m(X; \vec{\theta})\|_2 \;=\; \min_{\vec{\theta} \in \Theta} \|R(\vec{\theta})\|_2 \;=\; \min_{\vec{\theta} \in \Theta} \sqrt{\sum_{i=1}^{n} r_i^2(\vec{\theta})}$$

BUT there are also many other criteria. For instance there is the $\mathcal{L}^1$ method which consists of minimizing the absolute deviations of the residuals: this is the L.M.A.D.(which stands for Least Mean

Absolute Deviate) criterion,

$$\min_{\vec{\theta}\in\Theta} \sum_{i=1}^{n} |r_i(\vec{\theta})| \ .$$

If L.M.S. algorithms are so famous, it's not because there are the "best" but because there are the simpliest and the quickest ones. There are surely not the more robust and this is the point I would like to stress on.

3.2 Merit–function Criterion and Noise Distribution

In order to understand what *robustness* means and why L.M.S. algorithms are said to be not *robusts*, we shall show up the implicit link between the **minimizing criterion** (least square criterion for example) and the **noise distribution** when interpreting measured data as predictable data (depending on a model) plus noise :

$$y_i \ = \ m(x_i; \theta_1, \ \ldots, \ \theta_M) \ + \ r_i$$

where $\vec{\theta} = (\theta_1, \ \ldots, \ \theta_M)$ are the parameters of the model, which are defined by the functionnal relationship $x \mapsto m(x;\vec{\theta})$.

What is important is that the *Least Square Estimator* is the *Maximum Likelihood Estimator* when the noise r_i has a **Gaussian distribution**. Since the *Maximum Likelihood Estimator* is in some way the best estimator we can build from noisy measures, it is natural to find out the *Maximum Likelihood Estimator* of parameters $\vec{\theta} = (\theta_1, \ \ldots, \ \theta_M)$. And if the noise r_i is a Gaussian noise, it is equivalent to find out the *Maximum Likelihood Estimator* of parameters $\vec{\theta} = (\theta_1, \ \ldots, \ \theta_M)$ than minimize the Least Square criterion, because **in the Gaussian case, the M.L.E. is equal to the L.S.E.**: it is solution of the problem,

$$\min_{\vec{\theta}\in\Theta} \ \sum_{i=1}^{n} \|y_i - m(x_i;\vec{\theta})\|^2$$

The proof is given just beneath.

The M.L.E. is solution of $\max_{\vec{\theta}\in\Theta} p(y_1,\ldots,y_n;\vec{\theta})$. Since Y_i are supposed to be independant random variables and all have the density function ρ except for the mean (but they have the same standard deviation), $p(y_1,\ldots,y_n;\vec{\theta}) \ = \prod_{i=1}^{n} \rho(y_i - m(x_i;\vec{\theta}))$. If Y_i have a Gaussian law, $\rho(y) = \frac{1}{\sqrt{2\pi\sigma^2}} exp(-\frac{y^2}{2\sigma^2})$ and the M.L.E. is solution of:

$$\max_{\vec{\theta}\in\Theta} \ -\sum_{i=1}^{n} \|y_i - m(x_i;\vec{\theta})\|^2 \ .$$

So what I want to highlight is that if the noise is no longer gaussian, the M.L.E. won't be the L.S.E. Then finding out the Least Square Estimator won't give you the best estimator for parameters $\vec{\theta}$. Here is the list of three well-known probability distribution (which are likely to model the measurement noise) and their three associated merit-function to minimize when the M.L.E. is looked for. Each distribution is plotted on figure 2.

- Gaussian distribution(LMS criterion)

$$p(u) \ = \ \frac{1}{\sqrt{2\pi}\sigma} \, . \exp(-\frac{u^2}{2.\sigma^2})$$

$$\updownarrow$$

$$\text{minimize over } \vec{a} \quad \sum_{i=1}^{N} \|y_i - m(x_i;\vec{a})\|^2$$

- Double Exponential distribution(LMAD criterion)

$$p(u) \;=\; \frac{1}{2\sigma} \,.\, \exp(-\frac{|u|}{\sigma})$$
$$\updownarrow$$
$$\text{minimize over } \vec{a} \quad \sum_{i=1}^{N} |y_i - m(x_i;\vec{a})|$$

- Cauchy distribution(LMLSD criterion)

$$p(u) \;=\; \frac{1}{\pi\sigma} \cdot \frac{1}{1+\frac{u^2}{\sigma^2}}$$
$$\updownarrow$$
$$\text{minimize over } \vec{a} \quad \sum_{i=1}^{N} \ln\left\{1 + \|y_i - m(x_i;\vec{a})\|^2\right\}$$

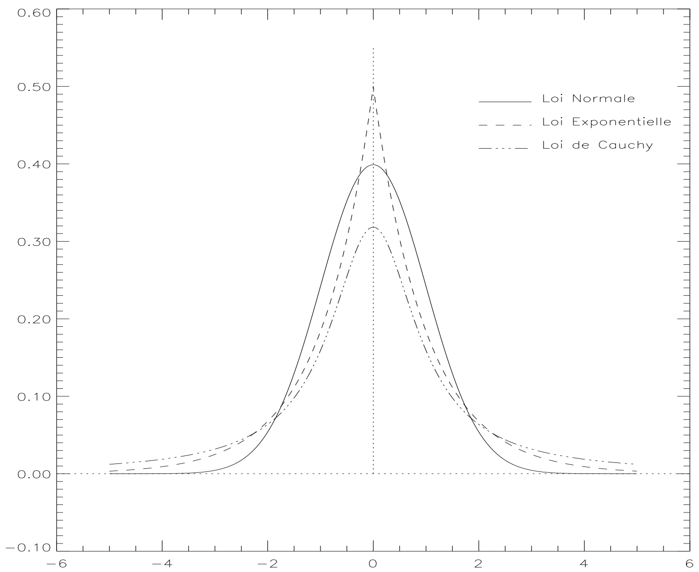


Fig. 2 - Probability Distribution

3.3 Robustness

3.3.1 *Robustness*: definition

In this paragraph, we will see what *robustness* means, *how robust* are algorithms, and how interpreting *robustness*.

As far as *Statistics* is concerned, I will take over the definition of "robust" in [1]: *insensitive to small departures from the idealized assumptions for which the estimator is optimized*, that is when you 've got large departures for a small number of data points. Roughly speaking *robust* means that estimators shouldn't take into account outliers! Of course, it is impossible to remove such points, but a robust estimator shouldn't give them too much importance when updating parameters of the model.

3.3.2 A short example

Before going forward, we are going to study a very simple example which show that the $\mathcal{L}^2$ method is less robust than the $\mathcal{L}^1$.

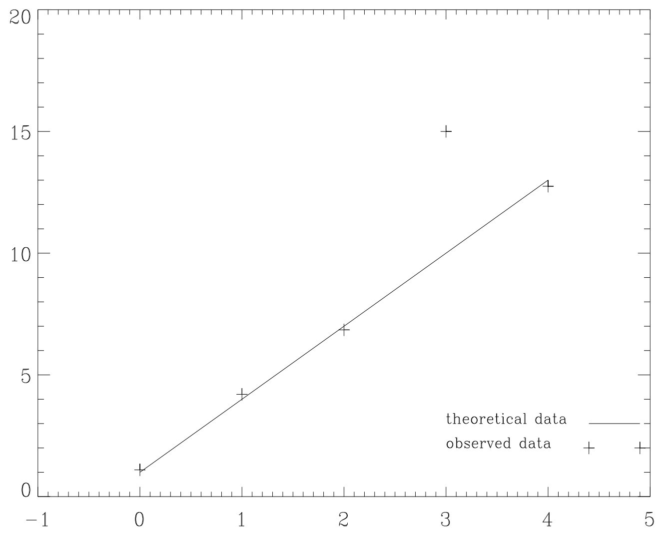


FIG. 3 - Example of a noisy line

Suppose you observe five values at five different time which follows the linear model $x \mapsto 3x + 1$. Because of the measurement noise, observations are not the theoretical expected values(see figure 3). In this case the vector of residuals is $R = (0.1, 0.2, -0.15, 5.0, -0.25)$. The 5.0 is most likely discordant and as a consequence, it will badly influence the estimation of parameters vector $\vec{\theta} = (a, b)$ when fitting the observed data with a linear model $m: \ x \mapsto ax + b$. The following table compare the results given by the LMS algorithm and the LMAD algorithm and figure 4 show the "best line" resulting from both algorithms in comparison with the theoretical line.

	a	b
theoretical values	3.0	1.0
LMS algorithm	3.41	1.16
LMAD algorithm	2.9125	1.10

The results are convincing: the LMS algorithm leads to a very bad fit because of one outlier ; it is clearly not robust if we refer to the above definiton of *robustness*, whereas the LMAD algorithm leads to a much better result. Futhermore LMAD algorithm is insensitive to the outlier since the "LMAD line" is under the theoretical line, that means it behaves as if the discordant point didn't exist. The difference between these two algorithms is puzzling! This is the difference between a robust and a non robust algorithms and that's why I'd like to warn you about the LMS algorithm. LMS algorithm can lead to a very bad estimation of the model parameters even for simple cases.

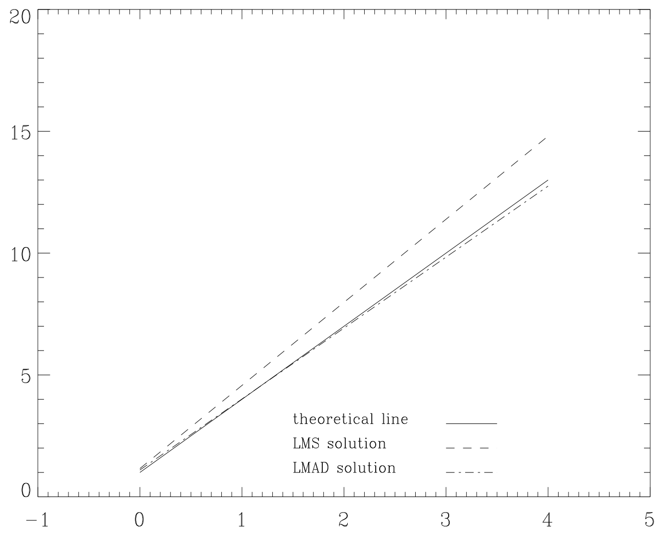


FIG. 4 - line fitting with LMS and LMAD algorithms

3.3.3 *Robustness*: outliers' weight

To understand how outlier influence more or less the algorithm and as a consequence how robust algorithms are, let's consider the merit–function minimizing criterion. The LMS and LMAD merit–function can be put in the following form:

$$\|R\|_1 = \sum_{i=1}^{n} 1.0 \times |r_i| \tag{1}$$

$$\|R\|_2^2 = \sum_{i=1}^{n} |r_i| \times |r_i| \tag{2}$$

Thus we can interpret LMS and LMAD algorithms as special cases of a more general algorithm which is to minimize the following weighted merit–function

$$\min_{\vec{\theta} \in \Theta} \sum_{i=1}^{n} w_i \times |r_i|,$$

where $W = (w_1, \ldots, w_n)$ are the weighted coefficients. In the case of an LMAD algorithm, each residual has the same weight whereas in the case of an LMS algorithm the weight of each residual is proportionnal to its own value. That means outliers have a greater weight than every other points when using a LMS algorithm(in the LMAD case, outliers have exactly the same weight than every other points). Therefore outliers have a much greater influence for minimizing the merit–function in the LMS case than in the LMAD case. The following table show the importance(in percent) of the outlier within LMS and LMAD merit–function as far as the above example is concerned. We also notice that the LMLSD algorithm is worse than the LMAD one as far as the previous example is concerned.

algorithm	importance of the outlier(%)
LMS	0.877
LMAD	0.200
LMLSD	0.487

3.3.4 ***Robustness*: probability that an outlier occurs**

An other way to answer to the question *where does this robustness come from?*, is to compare the tail of the density function of the gaussian curve($\mathcal{L}^2$ criterion) and the symetrical exponential curve($\mathcal{L}^1$ criterion).

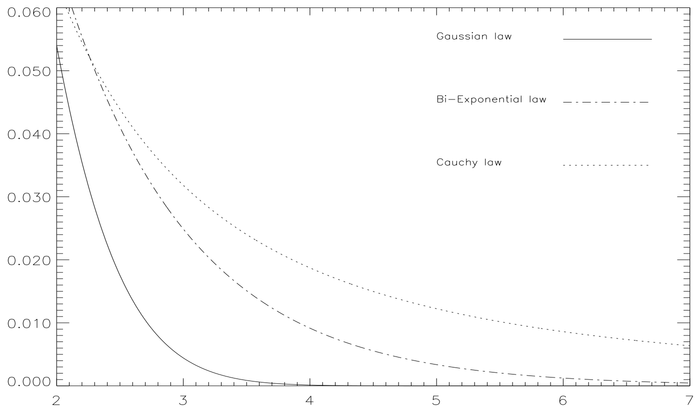


FIG. 5 - tail of density function for gaussian, bi–exponential and cauchy laws

These curves demonstrate that $\mathcal{L}^1$ criterion is more robust than $\mathcal{L}^2$ criterion: since the bi–exponential density function goes to zero far more slowly than the gaussian density function(see the figure 5), large residuals that would be extremely improbable with a gaussian noise, ie the LMS criterion, can altogether occur with a non negligeable probability. To be more precise we can refer to the table 3.3.4 which gives the probability of encountering a residual equal to or greater than a certain value, $\mathcal{P}(r \geq k.\sigma)$, for the gaussian , bi–exponential and cauchy laws.

$k.\sigma$	Gaussian	Bi–exponential	Cauchy
1	0.159	0.184	0.25
2	0.023	0.0677	0.148
3	$1.35\,10^{-5}$	0.025	0.102
4	$3.2\,10^{-3}$	$9.16\,10^{-3}$	0.078
5	$2.86\,10^{-7}$	$3.37\,10^{-3}$	0.063
10	$7.6\,10^{-24}$	$2.27\,10^{-5}$	0.032
20	$2.76\,10^{-89}$	$1.031\,10^{-9}$	0.016

We also notice than the cauchy density function goes to zero far more slowly than the bi–exponential density function, which means that the LMLSD criterion is more robust than the $\mathcal{L}^1$ criterion and far more robust than the $\mathcal{L}^2$ criterion.

4 Minimization Method

We have to minimize one of the three criteria given in the previous section, so we need a minimization algorithm to determine the best estimator $\hat{\vec{\theta}}$ of the parameters vector $\vec{\theta}$. Several algorithms exist and most of them can be splitted into two groups:

- $$\frac{\partial F(\vec{\theta})}{\partial \vec{\theta}} = 0 \tag{3}$$

  which means derive the criterion over the parameter $\vec{\theta}$ and find a solution of a non–linear set of m equations.

- $$\min_{\vec{\theta} \in \Theta} F(\vec{\theta}) \tag{4}$$

  which means find directly the minimum of the single merit–function F.

As far as I am concerned, I would prefer the second method for the following reasons:

- We know that the condition $\frac{\partial F(\vec{\theta})}{\partial \vec{\theta}}(\vec{\theta}_0) = 0$ is not sufficient to decide whether $\vec{\theta}_0$ is a minimum or not and further more whether or not it is a global minimum of function F. And if we refer to the form of the merit–function F and its number of parameters, we imagine it may have several local minima. That's why I would prefer the second group of minimization methods.

- The first method makes strong assumptions about continuity of the merit–function. For instance, the LMAD criterion leads to a merit–function F which is discountinous(because of the absolute deviate) and usually such discontinuities make equation solvers failed, especially when minima are close to the discontinuities. Of course you can say that you only use LMS criterion, but it is not robust. As for LMLSD criterion, the derivate of the merit–function has a bad form because of its peaks when the denominator is close to zero and equation solvers can fail as well. Notice this is also bad news for most of usual minimization routines.

For all these above reasons, I used the **downhill simplex minimization algorithm** which makes no assumptions about continuity of the merit–function and works for any kind of functions: it is the generalization of the dichotomy algorithm when considering real function of one–dimension–parameter. Besides such a method will surely give a global minimum. The seamy side of this good algorithm is that it time consuming in comparison to the well known gradient or quasi–Newton algorithms. But with hyper SPARC or SPARC 20 Unix machines, the routine runs very fast et takes only a few minutes for big raster and less than one minute for a small raster. So the downhill simplex minimization algorithm is worth being used.

5 In Practice

To fit a model(SAp model for instance) we choose the *Cauchy criterion*, I mean the **LMLSD criterion**, or the **LMAD criterion** which are quite robust. And for the numerical calculation of the esimate, we used the **downhill simplex minimization algorithm**. I'd prefer a robust technique for the estimation of the parameters of the model because on one hand we frequently have some outliers(think of "glitches") and on the other hand the noise is not always gaussian.

Références

[1] Press & al
Numerical Recipes in C,
The Art of Scientific Computing, 2e edition, 1992

[2] R. L. Branham
Scientific Data Analysis,
An introduction to overdetermined Systems,
Springer–Verlag, 1990