FAST AND ACCURATE PARAMETER ESTIMATION OF NOISY COMPLEX EXPONENTIALS WITH USE OF PRIOR KNOWLEDGE

Leentje Vanhamme * , *Aad van den Boogaart* † , *Sabine Van Huffel* *
*ESAT Laboratory, Department of Electrical Engineering, Katholieke Universiteit Leuven, Kardinaal Mercierlaan 94, 3001 Leuven, Belgium.
Tel: +32 16 321797; fax: +32 16 321986
e-mail: leentje.vanhamme@esat.kuleuven.ac.be
† Biomedical NMR Unit, Faculty of Medicine, Katholieke Universiteit Leuven, Belgium.

ABSTRACT

In this paper we address the problem of parameter estimation of magnetic resonance spectroscopy (MRS) signals. MRS signals are modeled as complex exponentials in noise. Iterative methods based on an optimisation procedure can be used for the parameter estimation. We examine which functional we have to minimise and which nonlinear least squares algorithms we have to use in order to attain maximum efficiency and robustness. The influence of starting values and prior knowledge is examined.

1 INTRODUCTION

We address the problem of fast and accurate parameter estimation of a sum of K exponentially damped sinusoids embedded in noise. The model function we use to model the N measured data points y_n is:

$$\begin{aligned} y_n = \hat{y}_n + e_n &= \sum_{k=1}^{K} a_k e^{j\phi_k} e^{(-d_k + j2\pi f_k)t_n} + e_n \\ n &= 0, 1, \ldots, N-1 \end{aligned} \tag{1}$$

where $j = \sqrt{-1}$, a_k is the amplitude, ϕ_k the phase, d_k the damping factor and f_k the frequency of the kth sinusoid; $t_n = n\Delta t$ with Δt the sampling interval and e_n is complex white Gaussian noise. The caret on y indicates that this quantity represents the model function rather than the actual measurements.

In most applications, biochemical prior knowledge in terms of the model parameters is available. Incorporating this prior knowledge in the signal processing algorithms leads to more accurate results. For example, amplitude ratios, frequency differences or damping ratios between peaks can be represented as linear combinations between parameters in the model function.

For parameter estimation of signals modeled according to equation (1), noninteractive methods exist that are noniterative, computationally efficient and which can be fully automatic. A serious drawback however is the fact that only very limited prior knowledge can be incorporated in these algorithms. Among this class of methods are the algorithms based on Kumaresan-Tufts' linear prediction (LP) method [6] combined with Singular Value Decomposition (SVD) . Kung's state-space approach [7] combined with SVD (called HSVD here [1]) is a more efficient and more accurate alternative to the LP methods as it circumvents the steps of polynomial rooting and root selection. Rapid and more accurate variants of the state-space algorithms have been recently proposed [10], but the limitations as to the imposition of prior knowledge about model function parameters are inherent to this type of method.

On the other hand, interactive methods exist that are iterative, computationally less efficient, but that do allow inclusion of prior knowledge. The algorithms fit the model function to the data in a nonlinear least squares (NLLS) sense, leading to maximum likelihood parameter estimates in the case of white Gaussian noise. The functional we have to minimise is:

$$\begin{aligned} G(a_k, d_k, f_k, \phi_k) &= \sum_{n=0}^{N-1} (y_n - \sum_{k=1}^{K} a_k e^{j\phi_k} e^{(-d_k + j2\pi f_k)t_n})^2 \\ &= \|\mathbf{y} - \Phi\, \mathbf{l}\|^2, \quad k = 1, \ldots, K \end{aligned} \tag{2}$$

with $\mathbf{y} = [y_0, \ldots, y_{N-1}]^T$ the signal vector, $\mathbf{l} = [a_1 e^{j\phi_1}, \ldots, a_K e^{j\phi_K}]^T$ and

$$\Phi = \begin{bmatrix} e^{(-d_1 + j2\pi f_1)t_0} & \cdots & e^{(-d_K + j2\pi f_K)t_0} \\ \vdots & \ddots & \vdots \\ e^{(-d_1 + j2\pi f_1)t_{N-1}} & \cdots & e^{(-d_K + j2\pi f_K)t_{N-1}} \end{bmatrix}$$

Suppose the nonlinear parameters f_k, d_k are known for $k = 1, \ldots, K$. Then the matrix Φ can be computed and an estimate for the linear parameters $\mathbf{l}$ is obtained by solving a linear LS problem: $\widehat{\mathbf{l}} = \Phi^{\dagger}\mathbf{y}$, with $\Phi^{\dagger}$ the pseudo-inverse of Φ. The original functional G becomes:

$$V(d_k, f_k) = \|\mathbf{y} - \Phi\Phi^{\dagger}\mathbf{y}\|^2, \quad k = 1, \ldots, K \tag{3}$$

which is called the variable projection functional [4]. In this way the amplitudes a_k and the phases ϕ_k are eliminated. As a result we obtain a minimisation problem where the number of variables is reduced but where the functional has become more complicated. Evaluation of the functional value and the Jacobian of V is far more

complicated than in the case of G. In [5] Kaufman introduces a simplification which results in a more efficient computation of V.
The prior knowledge we impose is always a set of linear relations between parameters and as a consequence we obtain a minimisation problem with linear equality constraints which we substitute in the original functional in order to get an unconstrained NLLS minimisation problem.
As a consequence, regardless of which of the two functionals (2) or (3) we use, we always have to solve an unconstrained NLLS optimization problem. To solve it we can use dedicated routines as described in the literature [2].
The program that is currently used in the biomedical world to analyze MRS signals is called VARPRO and minimises V by a modified version of Osborne's Levenberg-Marquardt algorithm [9] with use of Kaufman's simplification.
The primary goal of this paper is to improve the existing VARPRO program in terms of efficiency and robustness. Therefore we determine which of the two functionals we have to minimise and which NLLS algorithm we have to use for the MRS application. All our results are based on a large Monte Carlo study.

2 SIMULATION PROCEDURE

For all Monte Carlo simulations we use the same approach. The simulation signal we use is derived from a typical *in-vivo* ^{31}P spectrum measured in the human brain and consists of 256 complex data points and 11 exponentials (Figure 1). From the noiseless signal we obtain 300 noisy realisations by adding white Gaussian noise whose real and imaginary components have standard deviation σ_ν. We use a low, intermediate and high noise level ($\sigma_\nu = 5, 15, 25$). A method is considered to fail if the minimisation algorithm claims not to have found the solution or if not all peaks are resolved within specific intervals (Cramér-Rao dependent) lying symmetrically around the exact frequencies. For all our results we verified that these simple criteria guarantee that all methods find the same minimum in case of a good run. As a result, root mean-squared error, bias and standard deviation of all methods as a function of the noise level are the same and will not be shown.

3 TESTING OF NEW ALGORITHMS

3.1 Compared algorithms

The minimisation routine used in VARPRO is a Levenberg-Marquardt algorithm and we will refer to it as LM from now on. In [2] a good Levenberg-Marquardt algorithm (MINPACK) and a good secant code for NLLS (NL2SOL) are recommended for solving an unconstrained NLLS problem. In this study we use the MINPACK routine lmder of netlib and dn2g, the latest NL2SOL version, from the PORT library of netlib.

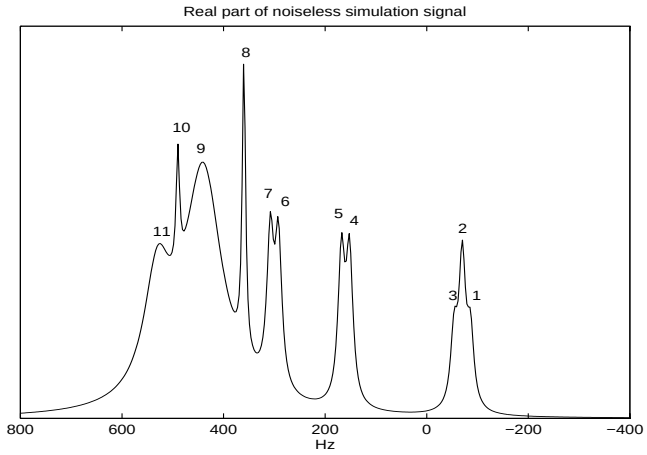

Figure 1: Typical frequency domain representation of noiseless simulation signal.

The three NLLS algorithms are used to minimise both functionals G in (2) and V in (3). As a result we have six different methods to compare.

3.2 Criteria for comparison

We compare the methods in terms of robustness and efficiency. To asses the robustness we look at the number of times the method fails. To compare the efficiency we look at the average number of functional and Jacobian evaluations and the average overall CPU time.

3.3 Signal

We impose no prior knowledge. This means that we have to estimate 4 parameters for each of the 11 peaks. In the case of minimisation of G we have a problem in 44 variables, compared to a problem in 22 variables in the case of minimisation of V. As all the methods are iterative, we have to provide starting values. This is done as follows. For each noise level we arbitrarily pick one signal out of 300. This signal is Fourier transformed to the frequency domain and displayed. We click on the top and the half height of each peak, a procedure which is known as peakpicking in the MRS world. The top value provides us with a starting value of the frequency and the value at half height is directly proportional to the damping of that peak. Starting values for the amplitudes and phases are needed for minimising G. We compute them by solving a LS problem as explained in the introduction. The obtained starting values are then used to process all the signals affected by the same noise level.

3.4 Results

The results of this Monte Carlo simulation are displayed in Table 1. The currently used VARPRO program is the LM algorithm used to minimise V. When we compare the three different algorithms for the minimisation of G,

σ_ν	crite-	*LM*		*NL2SOL*		*MINPACK*	
	rium	G	V	G	V	G	V
5	% fail	0	0	0	0	0	0
	jev	9.9	10.4	11.0	8.4	8.0	13.5
	fev	10.9	13.5	14.4	11.4	11.0	17.1
	cpu	43.7	68.9	43.2	53.0	37.4	87.3
15	% fail	0.33	12.3	0.33	0	0.33	0
	jev	10.6	11.5	13.6	11.6	10.2	10.2
	fev	11.7	15.8	18.2	14.8	13.7	13.3
	cpu	46.1	76.0	52.5	71.3	46.8	67.3
25	% fail	7.0	13.0	7.0	6.3	6.7	6.7
	jev	12.7	11.3	15.0	12.4	12.4	12.6
	fev	14.5	14.1	20.1	15.4	16.0	15.4
	cpu	55.5	73.1	58.3	76.1	55.8	83.9

Table 1: Results of Monte Carlo simulation (300 runs). LM, NL2SOL and MINPACK are used to minimise G and V. The criteria used are: % fail, the percentage of failures, jev, the average number of Jacobian evaluations, fev, the average number of functional evaluations, cpu, the average CPU time in seconds.

we see that they have the same number of failures. Concerning efficiency, NL2SOL needs the most functional and Jacobian evaluations, but since these evaluations are very "cheap" in this case, the overall CPU time of NL2SOL does not differ much from the others. When we compare the algorithms for the minimisation of V we see that the original VARPRO code has the highest failure rate of all. MINPACK and NL2SOL have the same failure rate and there is no clear winner in terms of efficiency. When we look at the difference between minimising V and G there are some interesting remarks to make. If the required number of functional and Jacobian evaluations is lower for minimisation of V than for minimisation of G, this does not necessarily mean that the overall CPU time will also be lower. This can be explained as follows. On the one hand the Jacobian of V has smaller dimensions than the one of G. This is advantageous because in each iteration step the parameter estimates are computed by solving a set of linear equations where the Jacobian enters. On the other hand, the computational complexity to compute the Jacobian of V is larger. So there is always a trade off between those two factors.

4 INFLUENCE OF STARTING VALUES

4.1 Starting values

In this section we investigate the effect of different starting values on minimising V and G. Again we compared all six methods, but the trends observed were independent of the algorithm used and therefore we only show the results for MINPACK. We use three different sets of starting values. The first set is taken, for every noisy signal, from the results of a fully automatic HSVD parameter estimation on that signal. As a second set we take the peakpick values "peakpick1" of subsection 3.3. Using the same procedure as in 3.3, we determine a third set of starting values "peakpick2". The "peakpick2" set was determined more carefully than the "peakpick1" set.

4.2 Results

σ_ν	crite-	*HSVD*		*peakpick1*		*peakpick2*	
	rium	G	V	G	V	G	V
5	% fail	0	0	0	0	0	0
	jev	3.8	3.6	8.0	13.5	7.7	6.7
	fev	5.0	4.6	11.0	17.1	9.9	7.8
	cpu	18.8	24.6	37.4	87.3	35.4	43.8
15	% fail	0.33	0.33	0.33	0	0	0.33
	jev	8.26	5.7	10.2	10.2	9.6	8.8
	fev	10.9	6.7	13.7	13.3	13.0	10.6
	cpu	37.9	37.1	46.8	67.3	43.9	57.3
25	% fail	54.3	52.7	6.7	6.7	6.3	7.3
	jev	21.3	12.9	12.4	12.6	11.3	10.7
	fev	27.2	15.3	16.0	15.4	15.1	12.8
	cpu	94.3	82.4	55.8	83.9	51.1	69.0

Table 2: Results of Monte Carlo simulation (300 runs). MINPACK is used to minimise G and V, three different sets of starting values are used.

For low and intermediate noise levels we see that the starting values provided by HSVD are better than those provided by manual peakpicking. For the high noise level however HSVD starts to fail and the starting values are no longer good. Hence the combination noninteractive-interactive methods is only useful for low and intermediate noise levels. We see that in the case of the "peakpick2" starting values the number of functional and Jacobian evaluations needed for V is always lower than in the case of G, but the overall CPU time is still higher for V. Only in the case of the "HSVD" starting values for $\sigma_\nu = 15$ or 25 we see that the overall CPU time is shorter for V.

5 INFLUENCE OF PRIOR KNOWLEDGE

5.1 Prior knowledge

Until now all signals were processed using no prior knowledge at all. In Table 3 all available knowledge about the simulation signal is given. In the first comparison we impose as prior knowledge that all peaks have a phase of 135° ("prior1"). This reduces the number of linear variables to 11. In the second we impose all the prior knowledge of Table 3 ("prior2"). The number of nonlinear parameters drops to 12 and the number of linear parameters to 5.

5.2 Results

The results are presented in Table 4. Again we only show the results for MINPACK, since the observed trends are independent of the algorithm we used. We see

peak k	f_k (Hz)	d_k	a_k	ϕ_k (°)
1	f_1	d_1	a_1	135
2	$f_1 + 16$	d_1	$2 \times a_1$	135
3	$f_1 + 32$	d_1	a_1	135
4	f_2	d_1	$2 \times a_1$	135
5	$f_2 + 16$	d_1	$2 \times a_1$	135
6	f_3	d_1	$2 \times a_1$	135
7	$f_3 + 16$	d_1	$2 \times a_1$	135
8	f_4	d_2	a_2	135
9	f_5	d_3	a_3	135
10	f_6	d_4	a_4	135
11	f_7	d_5	a_5	135

Table 3: Imposed relations between parameters of simulation signal ("prior2").

that the number of functional and Jacobian evaluations needed for minimisation of V and G are almost equal and the overall CPU times are also very comparable.

σ_ν	crite-	*prior1*		*prior2*	
	rium	G	V	G	V
5	% fail	0	0	0	0
	jev	7.2	6.4	5.0	5.0
	fev	8.7	7.4	6.0	6.0
	cpu	19.6	18.3	6.6	7.1
15	% fail	1.3	8.6	0	0
	jev	9.8	9.2	5.1	5.1
	fev	12.2	10.8	6.1	6.1
	cpu	26.1	25.6	6.5	7.1
25	% fail	3.6	11.0	1	0.6
	jev	11.6	11.5	7.9	7.7
	fev	13.8	13.5	9.2	9.0
	cpu	30.3	31.5	8.8	9.6

Table 4: Results of Monte Carlo simulation (300 runs). MINPACK is used to minimise G and V, two different sets of prior knowledge are used.

6 CONCLUSIONS

If we choose to minimise the variable projection functional V, the LM algorithm implemented in VARPRO is better replaced by either MINPACK or NL2SOL. In general we found that minimising the general functional G leads to more robust and efficient algorithms. LM, MINPACK and NL2SOL perform well on the minimisation of G. If the noise level is low or moderate, starting values provided by HSVD give the best results and fully automatic, fast and accurate parameter estimation is possible.

7 ACKNOWLEDGMENTS

The first author is a Ph.D. student funded by the IWT (Flemish Institute for Support of Scientific-Technological Research in Industry). The third author is a Research Associate with the N.F.W.O. (Belgian National Fund for Scientific Research). This work is supported by the Belgian Programme on Interuniversity Poles of Attraction (IUAP-nr.17), initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture, by the EU Programme "Human Capital and Mobility, Networks", project CHRX-CT94–432, and by a Concerted Research Action (GOA) project of the Flemish Community, entitled "Model-based Information Processing Systems".

References

[1] H. Barkhuysen, R. de Beer, D. van Ormondt, *Improved algorithm for noniterative time-domain model function to exponentially damped Magnetic Resonance Signals*, Journal of Magnetic Resonance 73, pp. 553-557, 1987.

[2] J.E. Dennis, R.B. Schnabel, *Numerical methods for unconstrained optimisation and nonlinear equations*, Prentice Hall, Englewood Cliffs, 1983.

[3] J.E. Dennis, D.M. Gay, R.E. Welsch, *Algorithm 573. NL2SOL - An adaptive nonlinear least-squares algorithm*, ACM Transactions on Mathematical Software, Vol. 7, No. 3, September 1981, pp. 169-383.

[4] G.H. Golub and V. Pereyra, *The differentiation of pseudo-inverses and nonlinear least squares problems whose variables seperate*, SIAM J. Numer. Anal., Vol. 10, No. 2, pp.413-432, April 1973.

[5] L. Kaufman, *A variable projecton method for solving seperable nonlinear least squares problems*, BIT 15, pp. 49-57, 1975.

[6] R. Kumaresan, D.W. Tufts, *Estimating the parameters of exponentially damped sinusoids and pole-zero modeling in noise*, IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP 30, No. 6, pp. 833-840, December 1982.

[7] S.Y. Kung, K.S. Arun, D.V. Bhaskar Rao, *State-space and singular-value decomposition-based approximation methods for the harmonic retrieval problem*, J. Opt. Soc. Am., Vol. 73, No. 12, pp. 1799-1811, December 1983.

[8] J.J. Moré, B.S. Garbow, K.E. Hillstrom, *User Guide for Minpack-1*, Argonne National Laboratory Report ANL-80-74, Argonne, IL, 1980.

[9] M.R. Osborne, *Some aspects of non-linear least squares calculations*, Numerical methods for nonlinear optimization, ed. Lootsma, Academic Press, London, 1972.

[10] S. Van Huffel, *Enhanced Resolution Based on Minimum Variance Estimation and Exponential Data Modeling*. Signal Processing 33, no.3, pp. 333-355, 1993.