J. Astrophys. Astr. (2005) **26,** 103–115

Parametric Identification of Solar Series based on an Adaptive Parallel Methodology

Juan A. Gómez Pulido[1], Miguel A. Vega Rodríguez & Juan M. Sánchez Pérez
Department of Computer Science, University of Extremadura, Campus Universitario s/n. 10071 Caceres, Spain.
[1]*e-mail: jangomez@unex.es*

Received 2004 September 4; accepted 2005 March 21

Abstract. In this work we present an adaptive parallel methodology to optimize the identification of time series through parametric models, applying it to the case of sunspot series. We employ high precision computation of system identification algorithms, and use recursive least squares processing and ARMAX (Autoregressive Moving Average Extensive) parametric modelling. This methodology could be very useful when the high precision mathematical modelling of dynamic complex systems is required. After explaining the proposed heuristics and the tuning of its parameters, we show the results we have found for several solar series using different implementations. Thus, we demonstrate how the result precision improves.

Key words. Sunspot—time series—system identification—parametric modelling—optimization, parallelism—adaptation.

1. Introduction

In many science and engineering fields, it is necessary to build mathematical models for studying the behaviour of phenomena and systems whose mathematical description is not available "*a priori*". One interesting type of these systems is the Time Series (TS). Time series are used to describe behaviours in many fields: astrophysics, meteorology, economy, etc. When dealing with TS there is only one signal available under observation; its physical structure is not known. This led us to employ the planning System Identification (SI) techniques (Söderström *et al.* 1989) in order to obtain the TS model. The model precision depends on the assigned values to certain parameters.

In this paper we have focused the effort of analysis in a kind of solar time series: the sunspot series. Then, we propose a parallel and adaptive heuristics to adjust the system identification main parameters with the aim of improving the precision of the parametric model of the sunspot series.

1.1 *Time series modelling*

We consider TS as a sampled signal with period T that is modelled with an ARMAX (Ljung 1999) parametric polynomial description (see equation 1).

$$y(k) + a_1 y(k_1) + \cdots + a_{na} y(k_{na}) = 0 \quad \text{where } na = \text{dimension}. \tag{1}$$

Basically the identification consists in determining the ARMAX model parameters a_i (θ in matricial notation) from measured samples $y(k_i)(\varphi(k)$ in matricial notation). Then it is possible to compute the estimated signal $y_e(k)$ (equation 2) and compare it with the real signal $y(k)$, computing the generated error (equation 3)

$$y_e(k) = [-a_1 y(k-1) - \cdots - a_{na} y(k-na)] = \varphi^T(k)\theta, \tag{2}$$

$$\text{error}(k) = y(k) - y_e(k). \tag{3}$$

The recursive estimation updates a_i in each time step k, thus modelling the system. The more sampled data processed, the more precision for the model, because it has more information about the system behaviour history. We consider SI performed by the well-known Recursive Least Squares (RLS) with forgetting factor (λ) algorithm (Ljung 1999). From the initial conditions, we start building $\varphi^T(k)$, and then RLS follows as it is shown in Fig. 1.

This algorithm is specified by the constant λ (forgetting factor), the initial values and the observed samples $\{y(k)\}$. There is no fixed value for λ, even if it is used as a value between 0.97 and 0.995 (Ljung 1991). The cost function F (see equation 4) is defined as the value to minimize in order to obtain the best precision.

$$F(\lambda) = \sum_{k=k_0}^{k=k_0+SN-1} |y_e(k) - y(k)| \quad (SN \text{ is the sample number}). \tag{4}$$

The recursive identification is very useful when it is a matter of predicting the following behaviour of the time series from the data observed up to the moment. For many purposes, it is necessary to make this prediction, and for predicting it is necessary to obtain information about the system. This information, acquired by means of the SI, consists in elaborating a mathematical parametric model for covering the system behaviour.

SI allows finding, in sample time, a mathematical model ($\theta(k)$) from which it is possible to predict the next behaviours. As identification advances in time, the

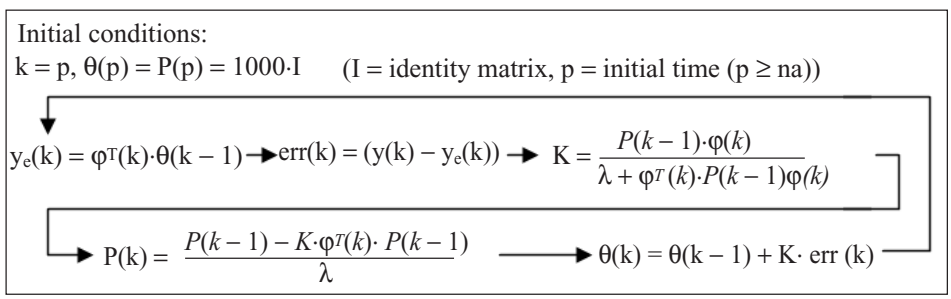

Figure 1. RLS equations.

predictions improve using more precise models. For example, we can compute in sample time the system model and then, with this model simulate the system future behaviour, forwarding real situations (Fig. 2).

For validating the SI evolutionary parallel architecture that we present, we use a time series set, corresponding to sunspot series obtained from measured observations (ROB 2004) (NOAA 2004). We have used 13 time series (Fig. 3) showing daily sunspots: 10 series (ss_00, ss_10, ss_20, ss_30, ss_40, ss_50, ss_60, ss_70, ss_80 and ss_90) each one corresponding to the sunspot measurements during ten years (for example, ss_20 compiles the sunspots from 1/1/1920 to 31/12/1929); 2 series (ss_00_40 and ss_50_90) each one covering 50 years, and 1 series (ss_00_90) covering all measurements of the 20th century.

2. The optimization problem

When SI techniques are used, the model is generated "*a posteriori*" by means of the measured data. However, we are interested in the system behaviour prediction in running time, that is, while the system is working and its data are being observed. So, it would be interesting to generate models in running time in such a way that a processor may simulate the system next behaviour.

At the same time, our first effort is to obtain a high model precision (minimal F). SI precision is due to several causes, mainly to the forgetting factor λ (Fig. 4). Frequently this value is critical for model precision. Other sources can also have lesser degree of influence (dimensions, initial values, the system...), but they are considered as problem definitions, not parameters to be optimized.

On the other hand, the precision problem may appear when a system model is generated in sample time: If the system response changes quickly, then the sample frequency must be high for avoiding the key data loss in the system behaviour description. If the system is complex and its simulation from the model to be found must be very trustworthy, then the required precision must be very high and this implies a great computational cost. Sometimes the hardware resources do not allow the computational cost in the model generation and processing to be lower than the sample period. We find a trade-off between a high sample frequency and a high precision in the algorithm computation. Adjusting the parameter λ for improving the precision can be conveniently done by means of techniques based on adaptive heuristics. So, the final goal is to design an architecture for real time SI suited for processes in which a high precision and a low sample time are required.

The method we suggest, and explain in the next section, can be used in any area. However, for the case of solar time series, the interval of sampling is not important, the real goal being to obtain the maximum accuracy of the estimation.

3. PARLS: the proposed heuristics

In order to find the optimum value of λ, we propose a parallel algorithm that is partially inspired on the concept of artificial evolution (Goldberg 1989) (Rechenberg 1973) and also in simulated annealing mechanism (Kirkpatrick *et al.* 1983). In our algorithm, named PARLS (Parallel Adaptive Recursive Least Squares), the optimization parameter λ is evolved for predicting new situations during the successive phases of the

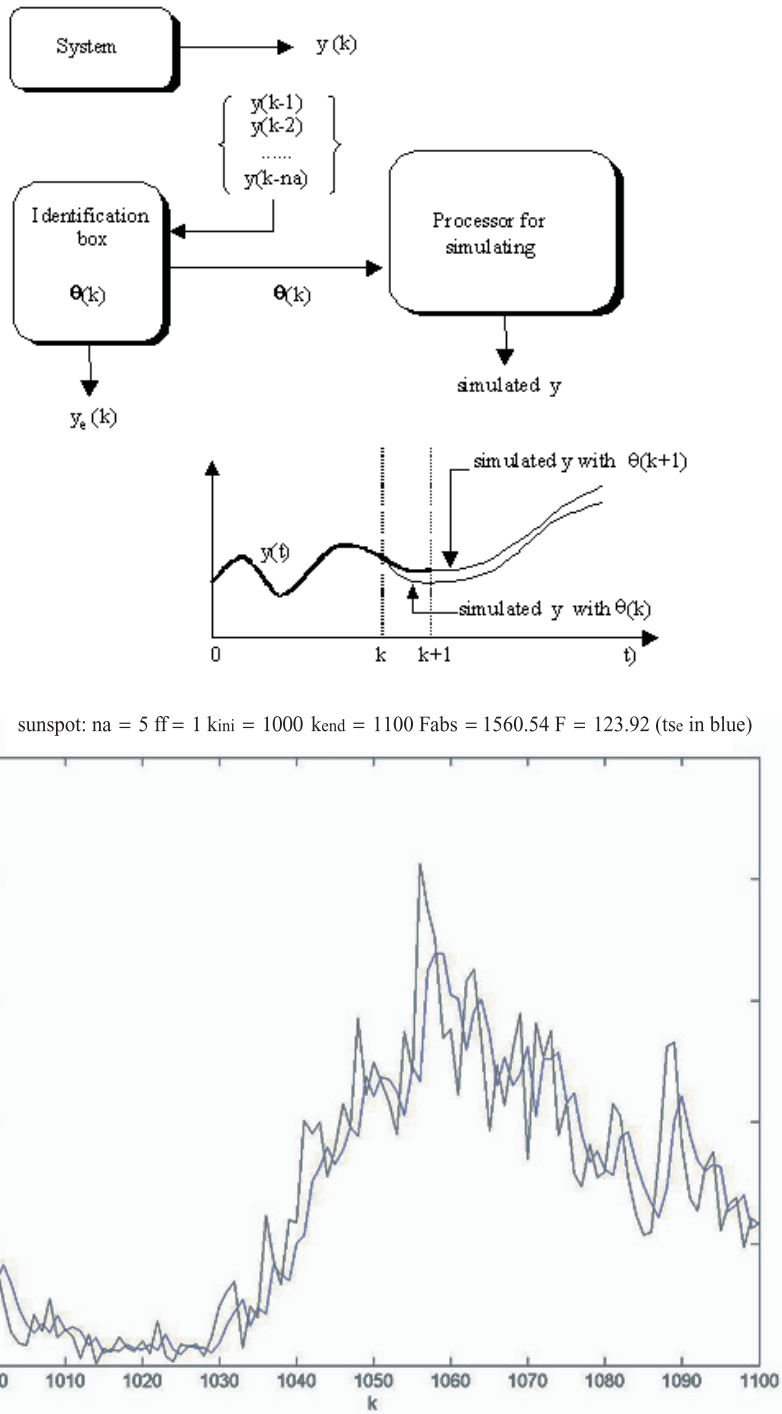

Figure 2. Recursive SI allows us to predict and simulate future system behaviour. On the right side, details of a sunspot time series used and its simulation based on the model identified.

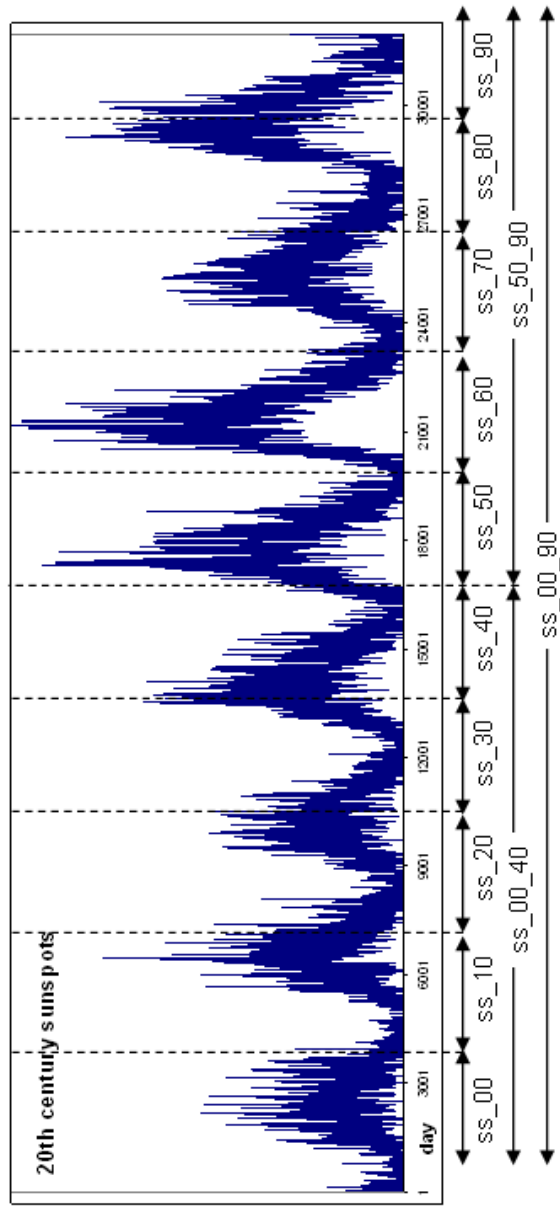

Figure 3. The sunspot time series used in this work.

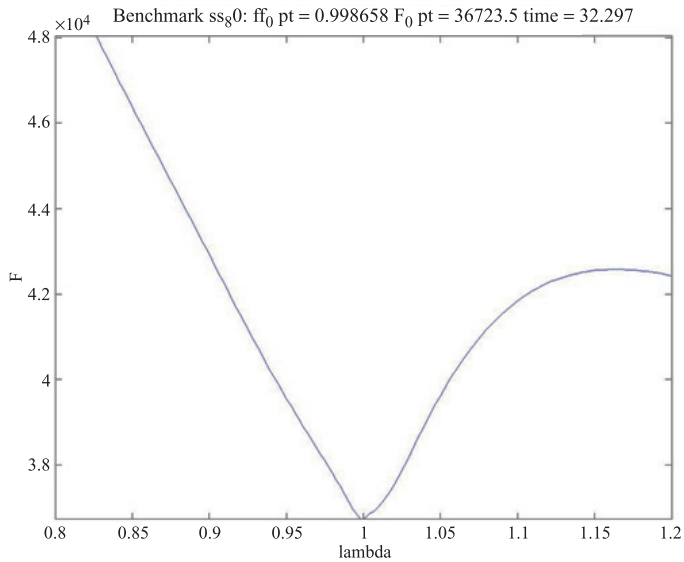

Figure 4. Cost function evaluated for several λ values using RLS identification for ss_80 benchmark when $na = 5$. We can see how the best performance is reached when $\lambda = 0.998658$.

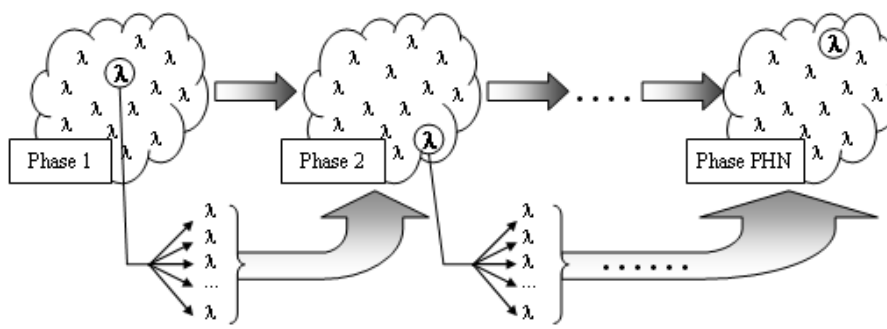

Figure 5. Evolution mechanism. In each phase, a set of λ values performs RLS identification during a certain number of samples. Then, the λ value whose corresponding F is the minimum of all computed F is the optimum, and from it a new set of λ values is generated. They are used in the next PARLS phase to perform new identification during the following PHS samples.

process (iterations of the algorithm). In other words, λ evolves at the same time that improves the cost function performance (Fig. 5).

PARLS considers a λ value as a state. Starting on an initial λ value (λc) and an initial R value (the interval of generation where λc is in the middle), a set of λ values is generated covering the entire interval R uniformly. The λ values generated are equal to the number of parallel processing units (PUN). Each phase of PARLS process is an identification loop that considers a given number of sample times (PHS) and the corresponding λ value. In this work, we use the nomenclature shown in Table 1.

In each phase, R is reduced dividing itself by the RED factor (the interval limits are moved so that the center of the interval corresponds with the optimal λ value found in the previous phase), in such a way that the generated set of λ will be more and

Table 1. PARLS nomenclature.

R	Generation interval
λc	λ central in R
PHS	Phase samples
PHN	Number of phases
PUN	Number of parallel processing units
TSN	Total number of samples
RED	Reduction factor of R

more near to the previous optimum found. The new set of generated λ values always covers uniformly the new interval R. In each processing unit, during each phase, the cost function F is computed (F is defined as the accumulated error of the samples that constitute each phase). From equation (4), we have:

$$F(\lambda_{\mathrm{PUx}}) = \sum_{k=k_0}^{k=k_0+PHS-1} |y_e(k) - y(k)|. \tag{5}$$

At the end of each phase, the best λ is chosen. This is the corresponding value to the lower F. From this λ, new values are generated in a more reduced (new R) interval (Figs. 6, 7). The goal is that the identifications performed by the processing units will converge to optimum λ parameters when a given stop criteria is achieved. So the identification will be of high precision.

Therefore, PARLS could be considered as population-based metaheuristics rather than parallel metaheuristics, because each processing unit is able to operate in isolation, as well as the tackled problem itself as only a single real-valued parameter (λ) is optimized.

4. Experimental results

We consider several criteria for evaluating PARLS performances. All these criteria have been fully checked and tested (Gómez *et al.* 2001) in order to get a set of better values for parameters and strategies. For example, we have studied strategies as the optimum λ criteria (the λ value that produces a minimum F), the stop criteria (indicating when a processing unit must stop the work), the model generation criteria (how to consider the initial model in the next phase), the optimum F definition (to consider the optimum F as the lowest in all phases or the lowest computed in the present phase), etc.

A question of interest is about the optimal model size. There is a greater computational cost when na is higher. Therefore, the na value should be selected in relation to the required computational cost, and this dimension is considered as part of the definition of the problem. We have computed several problems with different values for na, using SISO (Single Input Single Output) systems instead of time series and with processing units implemented with neural network (Gomez *et al*. 2002), and our conclusion is that an accurate tuning for general purposes could be from $na = 5$, although this value can be increased a lot for systems with wide sampling periods. Because a bigger na increases the model precision, for the sunspot series an adequate value of

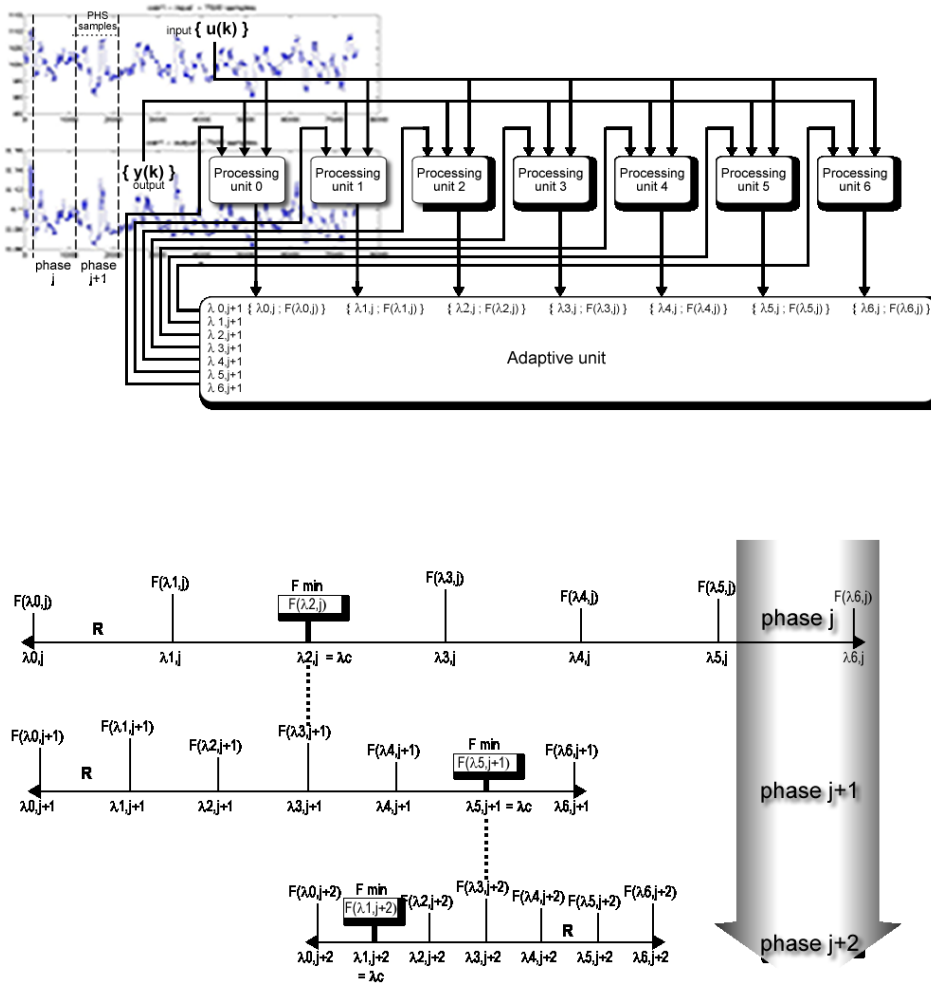

Figure 6. The SI uses different λ values in each phase performed by each processing unit. All the λ values in the same phase running in the processing units are generated in the R interval from the previous phase optimum λ found, corresponding with the smallest computed F.

na can be 20, since the computational cost is not big when they are series with a high interval of sampling (one day).

Another important question is how to establish the initial range of λ values in PARLS search. We have performed experiments in order to determine the approximated optimal λ for several series using a lot of RLS computations. In Fig. 8, some of these experiments are shown. We can see that there is a distinct optimal λ for each series, but in all the cases there is a smooth V-curve that is very useful to the initial PARLS search. We have thus selected as initial searching parameters tuned values $\lambda c = 1$ and $R = 0.05$. In view of the results, it could be said that the optimum λ value is 1, but it is dangerous to state this if we have not reduced the range more. Thus, reducing the interval of search, we will be able to find a near, but different optimum value to 1, as we can see in Fig. 9.

```
λc   /* λ  value in the middle of the set of λ values in the interval */
R    /* Length of that interval */
do   {
     Generating PUN λ's from λc, covering all R interval
     do  {                  /* phase = Identification of PHS samples */
         Computing accumulated error (F) in
         each PU during all samples of the phase
         }
         while (processing PHS samples);
         F(λ) = Accumulated error for each PU
         Determining λ optimum of all PUs for which F(λ) is minimum
         λc = λopt;
         R = R/RED;     /* R is reduced in a RED factor */
         }
         while (!stop criterion)
```

Figure 7. PARLS pseudo-code.

PARLS offers a great variability for its parameters. According to the results we have obtained, we can conclude that there are no common policies for tuning the parameters in such a way that the best results will always be found. But results indicate that there is a set of values for which the results are good. We can thus establish fixed values for PARLS parameters (Table 2) in order to define a unique algorithm applicable to any series. This has the advantage of a quick application for a series without the previous task of tuning parameters.

5. Conclusions

In Table 3, we show the comparison of found results between RLS search and PARLS heuristics for several sunspot series. In all the cases the same tuned parameters have been used ($na = 20$, PUN $= 11$, $\lambda c = 1$, $R = 0.05$, RED $= 2$, PHN $= 4$). In (a) the results of 11 RLS identifications with their corresponding 11 equidistant in R values of λ are shown, and in (b) the PARLS results are displayed too. The computational effort of 11 RLS identifications is almost equal to PARLS cost with 11 processing units, so both results can be compared in order to establish the conclusions. With these tuned parameters, PARLS always finds better results than RLS. This fact contrasts with the results obtained for series of other areas (Gomez *et al*. 2002), for which PARLS finds better results in most of them, but for a few series the difference of F oscillates between 2% and 5% RLS better than PARLS. Therefore, if for the series studied in Gomez *et al*. (2002) we said that PARLS improves or holds the results found with RLS with the same computational effort, now, for the sunspot solar series, we can say PARLS always gives us better results.

As starting point, we can say that the parallel adaptive heuristic PARLS offers a good performance, and this encourages us to follow this research. Also, nowadays, we are trying to accelerate the computation of some parts of PARLS using reconfigurable hardware to obtain a synthesis on specialized coprocessors capable of improving the global efficiency (Gomez *et al*. 2002) for series of other areas where the sampling interval is very short.

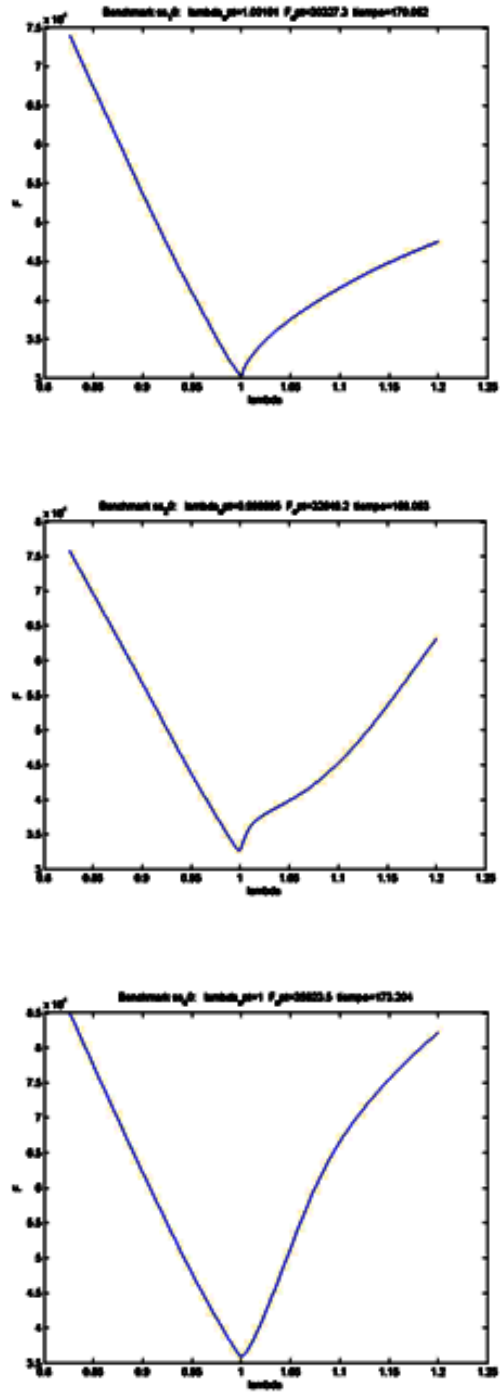

Figure 8. The cost function of the sunspot series ss_10, ss_20 and ss_40, calculated by RLS for 200λ values in the same range ($\lambda c = 1$, $R = 0.4$), using the same model dimension ($na = 20$). We can see a smooth V-curve in all the cases.

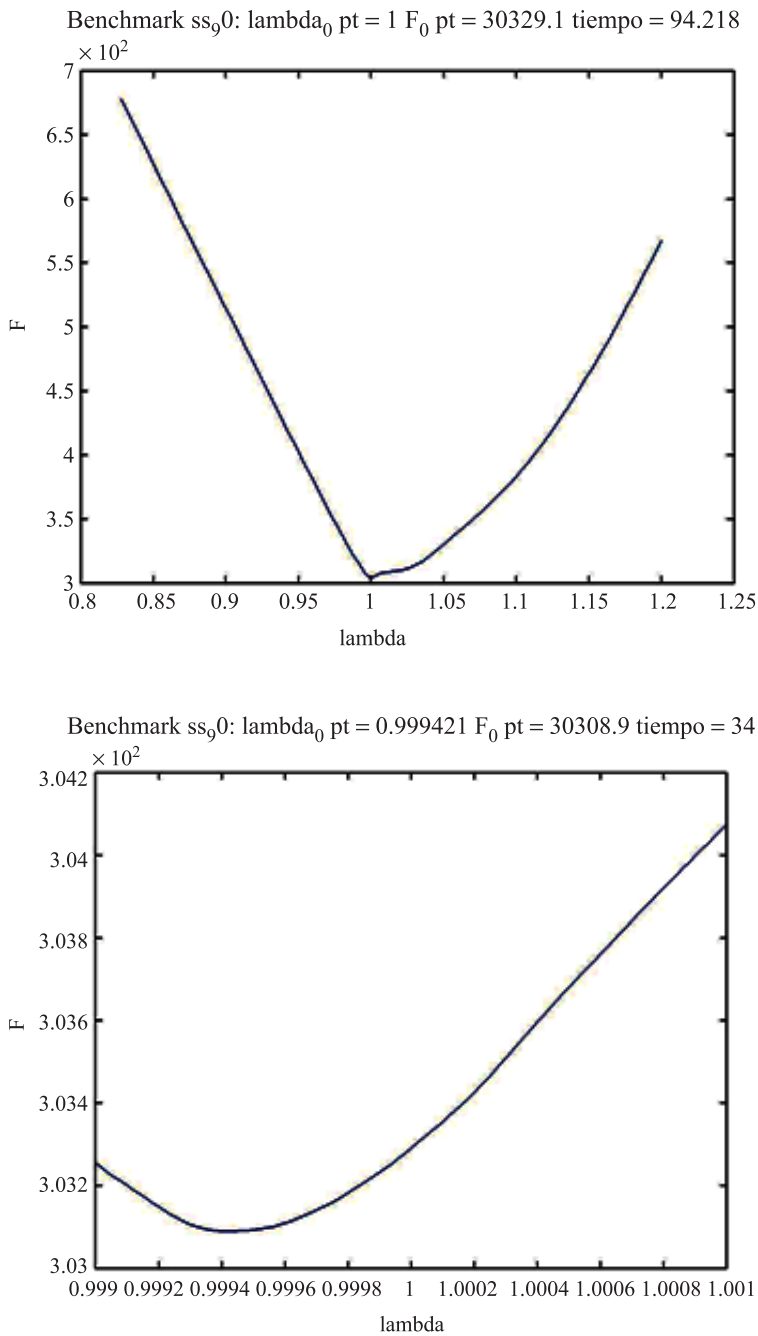

Figure 9. The cost function of ss_90 series calculated by RLS for 100λ values in two different ranges ($R = 0.4$ and $R = 0.002$), both centered in $\lambda c = 1$ and using the same model dimension ($na = 20$). We can see how the optimal λ value is not 1, and how it can be found doing the search range smaller.

Table 2. Main PARLS parameters tuned.

Parameter	Tuned value
na	20
λc	1
R	0.05
RED	2
PUN	11
PHN	4

Table 3. A summary of some results. This table shows a comparison of results between RLS search (RLS 11) and PARLS heuristics for several series with the same tuned parameters. The better algorithm is PARLS for all the series. See the conclusions section for more details.

Series	TSN	PHS	(a) RLS 11 F	(b) PARLS F	Better algorithm	PARLS vs. RLS 11
ss_00_90	36524	9131	$3.125670e+5$	$2.792250e+5$	PARLS	12%
ss_00_40	18262	4565	$1.561340e+5$	$1.344990e+5$	PARLS	16%
ss_50_90	18262	4565	$1.550950e+5$	$1.465290e+5$	PARLS	6%
ss_00	3652	913	$3.025500e+4$	$1.238500e+4$	PARLS	144%
ss_10	3652	913	$3.020200e+4$	$1.105800e+4$	PARLS	173%
ss_20	3653	913	$3.261300e+4$	$1.884100e+4$	PARLS	73%
ss_30	3652	913	$3.038700e+4$	$1.556800e+4$	PARLS	95%
ss_40	3653	913	$3.592300e+4$	$2.237500e+4$	PARLS	61%
ss_50	3652	913	$3.578300e+4$	$2.169700e+4$	PARLS	65%
ss_60	3653	913	$2.924200e+4$	$2.482800e+4$	PARLS	18%
ss_70	3652	913	$2.826900e+4$	$2.488500e+4$	PARLS	14%
ss_80	3653	913	$3.604300e+4$	$3.195270e+4$	PARLS	13%
ss_90	3652	913	$3.032900e+4$	$2.960300e+4$	PARLS	2%

Acknowledgement

This work has been developed thanks to TRACER project, MCYT Spain, TIC2002-04498-C05-01.

References

Goldberg, D. E. 1989, Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley.

Gómez, J. *et al.* 2001, "Sintonización de un sistema paralelo para la Identificación de Sistemas", In: Univ. de Extremadura (ed.): I Congreso Español de Algoritmos Evolutivos, 131–138.

Gómez, J. *et al.* 2002, "Diseño de un coprocesador reconfigurable para un algoritmo adaptativo paralelo de Identificación de Sistemas". In: Univ. de Granada (ed.): II Jornadas sobre Computación Reconfigurable y Aplicaciones, 221–226.

Gómez, J., Sánchez, J., Vega, M. 2003, "Using Neural Networks in a Parallel Adaptive Algorithm for the System Identification Optimization", Lecture Notes in Computer Science, 2687: Artificial Neural Nets Problem Solving Methods, 465–472.

Kirkpatrick, S., Gelatt, C., Vecchi, M. 1983, "Optimization by Simulated Annealing", *Science*, **220**, 4598, 671–680.
Ljung, L. 1991, System Identification Toolbox. The Math Works Inc.
Ljung, L. 1999, System Identification – Theory for the User. Prentice-Hall, London.
NOAA's National Geophysical Data Center (2004). Sunspot numbers. http://www.ngdc.noaa.gov/stp/SOLAR/ftpsunspotnumber.html
Rechenberg, I. 1973, Evolutionsstrategie: Optimierung technischer systeme nach prinzipien der biolgischen evolution. Formmann-Holzboog Verlag.
ROB, Royal Observatory of Belgium, Brussels (2004). Sunspot Data Series. http://sidc.oma.be/html/sunspot.html
Söderström, T. *et al.* 1989, System Identification. Prentice-Hall, London.