Finding the Optimal Stable Projection for AR Systems: New Results and a Genetic Approach *

S. Cruces, F. Pérez-González†
BSP Group, ETSI Telecommunicación, Universidad de Vigo, 36200 Vigo-SPAIN

Abstract

In this paper we consider the problem of finding the optimal stable projection for autoregressive systems, presenting a detailed description of the coefficients distance function. New important results linked to the convergence of the algorithms are given. We also use a new approach to find the optimal solution that avoids local minima by means of genetic algorithms.

1 Introduction

Given the characterization of an AR system

\[ x_k = - \sum_{i=1}^{n} a_i x_{k-i} + u_k \]  

where \( u_k \) is the driving process and \( x_k \) is the output process, we analyze the problem of estimating the coefficients of the autoregressive filter from the observation of \( x_k \). Most of the existing methods working in the coefficients space \( E_n^c \) do not ensure stability of the resulting filter making necessary the provision of some method that projects it onto the stability set \( S_n \), minimizing the coefficients distance. In [5] it is shown that AR estimation has the property that the estimated coefficients are jointly Gaussian distributed, therefore the most natural way of measuring the closeness is the \( l_2 \) weighted norm in \( E_n^c \). Then the distance function \( D_k(c) \) results in

\[ D_k(c) = (c - a)^T W (c - a) \]  

where \( a \) represents the unstable polynomial, and \( c \) is the stabilized projection onto \( S_n \). The best approach to the optimal weighted matrix \( W \) is given by the inverse autocorrelation matrix of the data. This can be justified in terms of the second order moments of the distribution. When that information is not present, the reasonable choice will be the identity matrix.

Classical methods such as reflection of the unstable roots inside the unit circle (CU) or truncation of the reflection coefficients \( k_i \) which have magnitude greater than one, do not lead to an optimal solution, in the sense discussed above. This was the reason for the recent appearance of a new class of optimal methods [2], [3], [4]. Their common feature is to avoid working directly in coefficients space, since the stability set has a very complex description in \( E_n^c \), being better to work in alternative spaces where it is a convex set, as the two order sections coefficients space \( E_2^c \) or the reflection coefficients space \( E_n^r \). This has been proven to give considerable advantages on computational reduction and on the feasible search of the minimal distance solution.

The method presented in [2] seems to have the lowest computational load of the three, while maintaining similar or even better convergence results. This method decomposes the stabilization problem of finding \( C(z) \) into subproblems of successive projections onto low order convex sets. These sets are formed by second order sections coefficients \( P_k(z) = \left(z^2 + x_1 z + x_2\right) \), and \( P_{n-k}(z) = (z + x_1) \), when \( n \) is odd,

\[ C(z) = \prod_{k=1}^{[n/2]} P_k(z) \]  

This paper is organized as follows: in section 2, we describe the distance function in the coefficients and alternative spaces, considering first the unconstrained case and then the constrained to the stability boundary case. Finally, in section 3 we present a new genetic approach that avoids local minima, thus determining the minimal distance solution for the sake of comparison of other methods.

2 The distance function

Our initial analysis of the distance function will be for the unconstrained case, i.e., where no stability requirements are imposed. This is done because of its simplicity and because the results obtained this way
will remain valid for the constrained case, which we will study later.

2.1 Unconstrained case

The analysis of the distance function will be carried out in three different spaces: first, the actual coefficients space where the distance function is denoted \( D_c(e) \), the other spaces are \( E_m^r \) and \( E_k^r \), in which the distance function is built by the transformations

\[ V : x \rightarrow c \quad \text{and} \quad L : k \rightarrow c, \]

with respective distance functions \( D_v(x) \) and \( D_l(k) \), that is,

\[ D_v(x) = D_c \circ V(x) = D_c(e) \quad \text{(4)} \]
\[ D_l(k) = D_c \circ L(k) = D_c(c) \quad \text{(5)} \]

In the polynomial coefficients space \((E_n^r)\) the distance function is quadratic, as one can see in (2), therefore it has only one minimum for \( c = a \) and is a convex function that describes an hyperparaboloid.

2.1.1 Section coefficients space: \( E_m^r \)

We will first consider first order sections, so the transformed space will be \( E_1^r \), and later generalize the results to other order spaces.

In order to locate the stationary points, we have to make zero the Jacobian, so using the chain rule we obtain

\[ \nabla_x D_v(x_v) = \nabla_c D_v(c_v) \cdot \nabla_x V(x_v) = 0. \quad \text{(6)} \]

From this expression we deduce that the minimum of \( D_v(c_v) \) is kept in \( D_v(x_v) \), but replicated as a consequence of the polyimage in \( E_1^r \) under \( V^{-1} \) for each polynomial coefficient \( c \). This follows from the fact that \( \nabla_x V(x_v) \) is always defined, so \( \nabla_c D_v(c_v) = 0 \) implies \( \nabla_x D_v(x_v) = 0 \). The confirmation of the character of minima stems from the positive definiteness of the Hessian matrix (7),

\[ \nabla^2_x D_v(x) = \nabla D_v(x) \cdot \nabla^2_c D_v(c) \cdot \nabla_x D_v(x) + \nabla_x D_v(c) \cdot \nabla^2_v(x) \quad \text{(7)} \]

Considering that each possible reordering of the sections \( x \) leads to the same result when applying the operator \( V \), it follows that the minima are symmetric with respect to the polysector planes\(^1\) in \( E_1^r \). Thus, if each section \( x_i \) has multiplicity \( v_i \) and \( d \) is the total number of different sections, it is easy to see that the number of global equivalent minima is

\[ N_{min} = \frac{n!}{\prod_{i=1}^{d} v_i!}. \quad \text{(8)} \]

All these minima have a common symmetry axis which coincides with the intersection of all the polysector planes, that is, it is formed by the coincidence of all sections and it is orthogonal to the subspace of the minima. Let \( M \) be the point on the symmetry axis equidistant to all the global minima, then we can state the following

Lemma 1 The equivalent global minima are non uniformly distributed on the surface of an \((n-1)\)-dimensional hypersphere of radius \( r \) and center in \( M \).

It can be shown [1] that \( M \) is the mean value of the location of all minima, \( M = [m, \ldots, m] \) where

\[ m = \frac{\sum_{i=1}^{n} x_i}{n} \quad \text{with} \quad \bar{c} = \frac{c_1}{n} \quad \text{(9)} \]

and \( r \) responds to the expression (10). For the case of all \( x_i \) real, this can be simplified further, resulting in \( r = r_r \), which only depends on the two first polynomial coefficients.

\[ r = \sqrt{\sum_{i=1}^{n} ||x||^2 - \frac{\bar{c}^2}{n}}; \quad r_r = \sqrt{\bar{c}^2 \cdot \frac{n-1}{n} - 2c_2} \quad \text{(10)} \]

The possible application of this result is related to the initialization of projection algorithms, since for high orders the density of minima grows faster than the

\(^1\)A polysector plane is the locus of those polynomials with at least one matching minimal section.
hypothesis’s area, so it is a good idea to initialize
from one random point on the hypersphere, which
implies no additional load. Figure 1 shows the sections
space and the minima geometry, where in this case
it can be checked that \( m = 0.8667 \) and \( r = 1.0033 \).
Figure 2 shows the distance function in the minima
plane and the circular-shaped valley where the min-
ima are located.

Examining equation (6) it can be readily seen that
new stationary points can be formed by applying the
transformation, since \( \nabla x V(x^+) = 0 \), is satisfied for
\( \nabla x D_x(e^+) = 0 \), even if \( \nabla c D_c(e^+) \neq 0 \). We then
have the following result

**Lemma 2** All the newly formed stationary points
are saddle points.

**Proof:** See [6].

In order to see where these saddle points are
exactly located, one has to solve the system of non-
linear equations in (6) where \( x^+ \) and \( e^+ = V(x^+) \neq 0 \)
are the unknowns. This system can be analytically
solved only for very low orders \( n \leq 3 \), as the com-
plexity increases with the order in a combinatorial
way. Instead, we can only try to place the saddle
points in a small subspace, which can be done by
finding the kernel of the gradient of the transformation
\( c = V(x) \), that results to be equivalent to solving the equation:

\[
det[\nabla x V(x)] = 0
\]  \( (11) \)

This is accomplished in appendix A.1, where it is
shown that

\[
det[\nabla x V(x)] = \prod_{i=1}^{n} (x_i - x_j) \]  \( (12) \)

so we arrive to the next conclusion:

**Lemma 3** All the saddle points for algorithms work-
ing with sections of order one are located in the hyper-
planes formed as a coincidence of two or more sec-
tions.

So far we have only considered the minimal sections
case. With higher order sections, the geometry of the
space changes in the sense that the number of poly-
sector planes and global minima is reduced, since any
possible reordering of the minimal components \( x_i \) into
the same section \( \alpha^{(s)} \) produces the same result. Then
the number of global equivalent minima has to be di-
vided by the permutations of the components \( x_i^{(s)} \) of
each section \( s \). Then Lemma 3 can be generalized to

**Theorem 1** The saddle points of the distance func-
tion in the space formed by the combination of co-
efficient sections \( \alpha \) with arbitrary order, are located

on the polysector planes, formed by the coincidence of
two or more components \( x_i \) belonging to different
sections \( \alpha^{(s)} \).

The proof can be found in Appendix A.2. Saddle
points would lock up any algorithm using the deriva-
tive as search information, so it is valuable to know
their location in stabilization algorithms. For example,
if one initializes a gradient based algorithm on a
polysector plane, it would always converge to a saddle
point, because of the symmetry of the distance that
forces the gradient to lie on the polysector. Con-
versely, one can avoid this problem by just initializ-
ing off the polysector planes. In successive projection
algorithms, a lock up would occur when the search di-
rection passes exactly through a saddle point, so the
probability of this happening is near to zero.

However, any type of algorithm would slow down
convergence when passing close to saddle point, as
it is illustrated in figure 3 that shows the lockup of
an optimization algorithm close to the saddle point
\( [1.2341, 1.2341, 0.0873] \), with the objective polyno-

![Figure 3: Distance vs. Iterations](image)

mial shown in figure 1.

2.1.2 Reflection coefficients space: \( E^k_{\tau} \)

In this space we have

\[
\nabla_k D_x(x) = \nabla c D_c(e) \cdot \nabla_k L(x) = 0 , \]  \( (13) \)

so the argument used to show that new critical points
may show up remains valid. As before, all the newly
formed stationary points are saddle points. In this
case, if \( |k_i| \neq 1 \) for \( i = 1, \cdots, n \) there is a unique mini-

mum, because there the Levinson-Durbin operator \( L \)
is injective. Next theorem determines the subspaces
which saddle points belong to,

**Theorem 2** All the newly formed stationary points
of the distance function \( D_k(x) \) belong to the subspace
\( \{ |k_i| = 1 ; \text{ for some } i \in \{ 1, \cdots, n \} \} \).

Due to the length of the proof, we refer the interested
reader to [1].
2.1.3 Constrained Case

When stability constraints are imposed, the previous analysis remains valid for those stationary points inside the stability region. However, the constraints enforce the apparition on the boundary region of new stationary points, that may be minima, saddle or maxima. There will be, in general, for unstable polynomials distant to the stability boundary, new local minima that difficult the projection task. Due to the complexity of the stability region, an analytical procedure for finding the stationary points location based on the Kuhn-Tucker multipliers results feasible only for very low orders \(n \leq 3\).

3 Genetic algorithms

Genetic algorithms (GA) are an emergent class of maximization algorithms based on the process of natural evolution, see [8] and references therein. Their main advantage is the robustness of the search and the efficient use of its operators. They can also solve minimization problems through the following transformation of the objective function,

\[
f(\delta) = D_{\text{max}} - D(\delta)
\]

Therefore, they are a good candidate for finding an approximation to the true global minimum in the stabilization problem. Once the global minimum valley of the distance function is found, one can tune the solution with a specialized optimal projection method, so the convergence to the true solution is generally guaranteed.

One important problem to solve when applying GA is the population codification. In our case, this can be accomplished by two alternative spaces: the reflection coefficients space or the second order sections space. In the first one, codification is easy because the stability region is a hypercube in the \(E^n\) space. In the \(E^2\) space, the stability region is a triangle, which difficulties the codification. A uniform gridding can be obtained by using two variables \(v_1, v_2 \in [0, 1]\) that define the position in the square of figure 4-a, through the transformation

\[
\alpha = -2v_1 + 2v_2
\]

\[
\beta = 1 - 2 \cdot \text{sgn}(1 - v_1 - v_2) \cdot (1 - v_1 - v_2)
\]

This codification results to be diploid since the square is folded by its diagonal into the stability triangle, as is represented in figure 4. Similar convergence results are obtained on both spaces. A typical parameter configuration for the GA follows. We used the elitist model, i.e., the best and worst individuals are propagated to the next generation. The codification was chosen to be binary, with a population size of 50 individuals (for a low/medium order polynomial), linear scaling, unit cross probability \(p_c\), mean mutation \(T_{\text{mut}}\) for the 10% of the population on each generation, and copy factor for the best \(\lambda\) of 1.1. The \(T_{\text{mut}}\) parameter is set to a high mutation rate as we are interested in having a great innovation source. At the same time \(\lambda\) is small to make the convergence to the global minimum slower but more robust. The elitist model is used because it is less sensitive to changes in the parameter configuration, achieving always acceptable convergence results.

4 Conclusions

In this paper we have presented an extension of the results in [2]. We have studied the properties of the distance function for various working spaces, related to the problem of finding the optimal stable projection of an unstable polynomial. We have characterized the subspaces that contain critical points, for algorithms based on both the reflection coefficients and arbitrary order sections. The results obtained are of value in the initialization and lookup avoiding of those algorithms. For the constrained to the stability boundary case, an analytic solution is only possible for low order polynomials. Another application
of these methods is to optimally stabilize adaptive algorithms working on the coefficients space, each time they run out of the stability region [6].

In addition, we have given a technique using genetic algorithms that finds the global minimum, for which we have presented adequate values for the parameters and a solution to the codification problem.

A Appendix

A.1 Determinant of the Jacobian

The polynomial \( C(z) \) is decomposed in sections of order one

\[
z^n + \sum_{k=1}^{n} c_k z^{n-k} = (z + x_i) \prod_{j=1, j \neq i}^{n} (z + x_j)
\]

Taking partial derivatives w.r.t. each \( x_i \) one has:

\[
\frac{\partial c_k}{\partial x_i} = \text{Coeff. of } z^{n-k} \text{ in } \prod_{j=1, j \neq i}^{n} (z + x_j) = \gamma_{k,i}
\]

Then the Jacobian matrix is

\[
\nabla_x V(x) = \Gamma_{n \times n} = (\gamma_{k,i}); k, i = 1, \ldots, n
\]

Using the relation \( \nabla_x V(x) = TA \) obtained in [7], where \( T \) is a lower triangular matrix with ones in the diagonal and \( A \) is the Vandermonde matrix of the polynomial roots \( \mathbf{x} \), the determinant of the Jacobian is equivalent to the well-known determinant of the Vandermonde matrix, then we arrive to (12).

A.2 Proof of Theorem 1

Assume that the algorithm works with an arbitrary combination of \( l \) sections of different orders. In order to distinguish arbitrary order sections from first order sections, we will utilize the \( \alpha \) symbol for the former. Each section \( \alpha^{(s)} \) is labeled with the superindex \( s \) and will have degree \( l_s \). In analogy with equation (17) the polynomial is split into the product of sections. Given the transformations \( V_{x \rightarrow c} \), \( V_{\alpha \rightarrow c} \) and \( V_{x \rightarrow \alpha} \) between sections and coefficients, we can use the chain rule and apply determinants to derive a relation involving them:

\[
det[\nabla_x V_{x \rightarrow c}] = det[\nabla_x V_{\alpha \rightarrow c}] \cdot det[\nabla_x V_{x \rightarrow \alpha}]
\]

The last determinant can be solved easily, because the coefficients of \( \alpha^{(s)} \) only depend on the minimal sections \( \{ x_i^{(s)} ; i = 1, \cdots, l_s \} \) building it and are independent to the others. Hence, the jacobian matrix can be expressed in the next block diagonal format

\[
\nabla_x V_{x \rightarrow \alpha}(x) = \begin{bmatrix}
[\Gamma_1] & [0] & [0] \\
[0] & \ddots & [0] \\
[0] & [0] & [\Gamma_l]
\end{bmatrix}
\]

where the block matrix \( \{ \Gamma_s ; s = 1, \cdots, l \} \) is similar to (19) but applied to the \( s \) section components. Then, we have

\[
det[\nabla_x V_{x \rightarrow \alpha}(x)] = \prod_{s=1}^{l} \det[\Gamma_s].
\]

The left hand term of (20) has been already calculated in (A.1), so we can use (22) and (20), to write

\[
det[\nabla_x V_{x \rightarrow \alpha}(x)] = \frac{\prod_{i< \theta} (x_i - x_j)}{\prod_{s=1}^{l} \prod_{i \leq j} (x_i^{(s)} - x_j^{(s)})}
\]

Eliminating common terms we arrive to:

\[
det[\nabla_x V_{\alpha \rightarrow c}(\alpha)] = \prod_{i < j} (x_i^{(s)} - x_j^{(s)})
\]

This determinant is zero only on the polysector plane, e.g. those having the same minimal component \( x_s \) on two or more different sections \( \alpha \). The proof is then completed.

References


