Approximate Multidimensional (m-D) Polynomial Factorization into Linear m-D Polynomial Factors using Genetic Algorithms

N. E. MASTORAKIS and I. F. GONOS
Military Institutions of University Education (MIUE),
Hellenic Naval Academy, Department of Computer Science,
Terma Hatzikyriakou 18539, Piraeus, GREECE.

Abstract: In this paper, we solve the problem of approximate multidimensional (m-D) polynomial factors using Genetic Algorithms (GA's). The non-factorizable multidimensional polynomial is approximately factorized into linear factors in the sense of the least squares approach. Using GAs we obtain better results than other methods of minimization (numerical techniques, neural networks, etc.). The results are illustrated by means of a 2-D numerical example.
Keywords: Approximate Multidimensional (m-D) Polynomial Factorization into Linear m-D Polynomial Factors Using Genetic Algorithms

1. Introduction

Multidimensional Polynomial Factorization owns an importand role in multidimensional systems stability, filter realization as well as in Distributed Parameter Systems theory and in other relevant disciplines [1]÷[3].

In [4], the factorization in factors of one variable i.e. $f(z_1,...,z_m) = f_1(z_1)...f(z_m)$ and in factors with no common variables i.e. $f(z_1,...,z_m) = f_1(\bar{z}_1)...f_k(\bar{z}_k)$ $(\bar{z}_1,...,\bar{z}_k)$ are mutually disjoint groups of independent variables, has been investigated. In [5], the factorization is succeeded by considering the given polynomial as (1-D) polynomial with respect to z_j and applying the well known formulas from 1-D algebra. In [6], the factorization of the state-space model is achieved. The factorization of an *m*-D polynomial in factors where at least one factor contains no more than *m*-1 variables has been studied in [7], while in [8], the factorization is achieved by factorizing two other lower order polynomials. In [9], the problem of the (exact) *m*-D polynomial factorization in linear *m*-D polynomial factors has been completely solved. Papers [5]÷[9] actually publish the material of [3] which have been produced by the author.

If one type of factorization does not hold though it is desirable, the (optimum) approximation of the original polynomial by a factorizable (of the considered type) one is attempted.

Some interesting cases of *m*-D approximate factorization have already been presented in the papers [10], [11]. In this paper, after presenting the results of [9] and [12], the approximate *m*-D polynomial factorization into linear m-D polynomial factors is attempted using GA's. An example is also given.

2. Exact Factorization into Linear m-D Factors

The (exact) factorization of an m-D polynomial

$$f = f(z_1,...,z_m) = \sum_{i_1}^{N_1} ... \sum_{i_m}^{N_m} a(i_1,...,i_m) z_1^{i_1}...z_m^{i_m} \quad (1)$$

has been solved ([3] and [9]) in the case that the *m*-D polynomial can be written as a product of linear m-D factors:

$$f(z_1,\dots,z_m)=\prod_{i=1}^{N_1}\left(z_1+a_{i,2}z_2+\dots+a_{i,m}z_m+c_i\right) \tag{2}$$

The sufficient and necessary conditions for such a factorization are already known [3], [9]. In particular, the following Theorem is proved. From this Theorem, the values of the unknown coefficients $a_{i,2},\dots,a_{i,m},c_i$ are obtained as well as the necessary and sufficient conditions for the existence of such a factorization.

Theorem 1

For the polynomial given in (1) suppose that: $N_1=\max(N_1,\dots,N_m)$, $a(N_1,0,\dots,0)=1$.
It is considered that no monomial of $f(z_1,\dots,z_m)$ has degree greater than N_1. Iff (If and only if)

$$a(i_1,\dots,i_m)=\sum\left(\prod^{i_2}a_{i,2}\right)\dots\left(\prod^{i_m}a_{i,m}\right)\cdot\left(\prod^{N_1-i_1-\dots-i_m}C_1\right) \tag{3}$$

where the product

$$\prod^{i_k}a_{i,k_1} \qquad k=2,\dots,m$$

involves only i_k coefficients of N_1 factors a_{i,k_1} $i=1,\dots,N_1$: so the sum in the right-hand side of equality (3) involves all possible terms formed by spliting the integer N_1 into integers $i_1,\dots,i_m,N_1-i_1-\dots-i_m$ (therefore this sum has

$$\binom{N_1}{i_1,\dots,i_m}=\frac{N_1!}{i_1!\dots i_m!(N_1-i_1-\dots-i_m)!}$$

terms), then the above polynomial can be factored as in (2) where $-c_i$ are the roots of the 1-D polynomial $f(z_1,0,\dots,0)$ and

$$a_{i,k}=\left.\frac{df(z_1,\dots,0,z_k,0,\dots,0)}{dz_k}\right|_{\substack{z_k=0\\ z_1=-c}}\cdot\left.\frac{z_1+c_i}{f(z_1,0,\dots,0)}\right|_{z_1=-c_i} \tag{4}$$

Proof: The Proof can be found in [3] as well as in [9].

The number of the relations (3) is l where $l=((\mathrm{N}_1+m)!/(N_1!m!))-1$ but one excludes the N_1 relations that in the left-hand side of (3) have $a(j,0,\dots,0)$ with $j=0,\dots,N_1-1$ and the $\mathrm{N}_1\cdot(m-1)$ relations that in the left-hand side of (3) have $a(j,0,\dots,0,1,0,\dots,0)$ with $j=0,\dots,N_1-1$.

Theorem 1 is applied only in the case where all the roots of $f(z_1,0,...,0)$ are simple. If there exists a multiple root i.e. there are $i_1,...,i_p$ such that $c_{i_1}=...=c_{i_p}$ an analogous theorem (Theorem 2 in [9]) should be applied.

Arranging all the results of these theorems in a logical order one can state the following algorithm:

Step 1: *Read* $a(i_1,...,i_m)$.

Step 2: *Arrange* the variables $z_1,...,z_m$ so that $N=N_1=\max(N_1,...,N_m)$.

Step 3: *Check*, if $a(N_1,0,...,0)\neq 0$ and $a(i_1,...,i_m)=0$ for $i_1+...+i_m>N_1$. If this is not the case, the method is not applied. -END

Step 4: *Let* $a(i_1,...,i_m):=\dfrac{a(i_1,...,i_m)}{a(N_1,0,...,0)}$

Step 5: *Find* (numerically) the roots $-c_i$ of the polynomial $f(z_1,0,...,0)$

Step 6: *Check* if $-c_i$ is a simple or a multiple root then *Find* $a_{i,k}$ from (4). If $-c_i$ is a $p-tuple$ root then *Find* $a_{i_1,k},...,a_{i_p,k}$ from Equations (13.1) (13.p) in [9]. Solve this system using the Vieta polynomial.

Step 7: *Check* the validity of (3). -END.

3. Problem Formulation and Solution

Suppose that the polynomial $f=f(z_1,...,z_m)$ (in (1)) is not factorized (using the previous method) into a product of linear *m*-D factors. However, in many engineering applications, especially in problems concerning realization of a certain *m*-D system, this type of polynomial factorization is necessary. For this reason, an attempt could be made to write approximately the above polynomial in a factorized form. So, first, we consider the unknown factorizable polynomial, where:

$$\tilde{f}(z_1,...,z_m)=\sum_{i_1=0}^{N_1}...\sum_{i_m=0}^{N_m}\tilde{a}(i_1,...,i_m)z_1^{i_1}...z_m^{i_m}=\prod_{i=1}^{N_1}\left(z_1+\tilde{a}_{i,2}z_2+...+\tilde{a}_{i,m}z_m+\tilde{c}_i\right) \quad (5)$$

and secondly we minimize the norm $\left\|f-\tilde{f}\right\|_2$ where

$$\left\|f-\tilde{f}\right\|_2^2=\left\|f(z_1,...,z_m)-\tilde{f}(z_1,...,z_m)\right\|_2^2$$

$$=\left\|f(z_1,...,z_m)-\prod_{i=1}^{N_1}\left(z_1+\tilde{a}_{i,2}z_2+...+\tilde{a}_{i,m}z_m+\tilde{c}_i\right)\right\|_2^2$$

$$= \sum_{\substack{i_1=0 \\ (i_1,\ldots,i_m)\neq(0,\ldots,0)}}^{N_1} \cdots \sum_{i_m=0}^{N_m} \left(a(i_1,\ldots,i_m) - \tilde{a}(i_1,\ldots,i_m)\right)^2 \qquad (6)$$

where the symbol $\tilde{}$ is used for the corresponding quantities of the unknown factorizable polynomial $\tilde{f}(z_1,\ldots,z_m)$.

In [12], the minimization has been attempted by using the Levenberg-Marquardt routine.

In the present paper, we use a new optimization technique using an appropriate Genetic Algorithm (GA).

A brief overview of the GAs methodology could be the following: Suppose that we have to maximize (minimize) the function $g(x)$ which is not necessary continuous or differentiable. GAs are search algorithms which initially were inspired by the process of natural genetics (reproduction of an original population, performance of crossover and mutation, selection of the best). The main idea for an optimization problem is to start our search no with one initial point, but with a population of initial points. The 2*n* numbers (points) of this initial set (called population, quite analogously to biological systems) are converted to the binary system. In the sequel, they are considered as chromosomes (actually sequences of 0 and 1).

The next step is to form pairs of these points who will be considered as parents for a "reproduction" (Fig.1)

$$\left.\begin{array}{l} 01100\,|\,100...11 \\ 00011\,|\,101...10 \end{array}\right\} \rightarrow \begin{array}{l} 01100101...10 \\ 00011100...11 \end{array}$$

parents children

Fig.1: Crossover

"Parents" come to "reproduction" where they interchange parts of their "genetic material". (This is achieved by the so-called *crossover*, Fig.1) whereas always a very small probability for a Mutation exists. (Mutation is the phenomenon where quite randomly - with a very small probability though - a 0 becomes 1 or a 1 becomes 0). Assume that every pair of "parents" gives *k* children.

By the reproduction the population of the "parents" are enhanced by the "children" and we have an increasement of the original population because new members were added (parents always belong to the considered population). The new population has now 2*n*+*kn* members. Then the process of *natural selection* is applied. According the concept of natural selection, from the 2*n*+*kn* members, only 2*n* survive. These 2*n* members are selected as the members with the higher values of *f*, if we attempt to achieve maximization of *g* (or with the lower values of *g*, if we attempt to achieve minimization of *g*). By repeated iterations of reproduction (under crossover and mutation) and natural selection we can find the minimum (or maximum) of *g* as the point to which the best values of our population converge. The termination criterion is fulfilled if the mean value of *g* in the 2*n*-members population is no longer improved (maximized or minimized). More detailed overviews of GAs can be found in Goldberg (1989) and Eberhart R., Simpson P. and Dobbins R. (1996), Grefenstette (1986), Kosters, Kok and Floréen (1999). Recent results and applications can be found in Mastorakis (1998a) and (1998b) and Mastorakis (1999).

In our problem of *m*-D polynomial factorization, we wish to minimize *g* where $g = \left\| f - \tilde{f} \right\|_2$ over $\tilde{a}(i_1,\ldots,i_m)$. To this end, every $\tilde{a}(i_1,\ldots,i_m)$ is converted to the binary system and is considered as part of a big chromosome: 100110010|001000111|...|111001010 where every part corresponds to a particular $\tilde{a}(i_1,\ldots,i_m)$. If we suppose that every $\tilde{a}(i_1,\ldots,i_m)$ is converted to a *t*-bits binary number, for the "chromosome" of $\tilde{a}(i_1,\ldots,i_m)$ we need *Mt* bits, where *M* is the number of $\tilde{a}(i_1,\ldots,i_m)$. Our search starts with a randomly generated population of such 2*n* chromosomes. In a quite random manner, this population is split into pairs of parents that will be crossed i.e. they will interchange their genetic material (with *c* crossovers) always under a very small probability *P* for mutation (for example *P*=0.01).

By this reproduction, a new population of 2*n*+*kn* members will be formed, since each pair of parents give birth to *k* children. The new population is filtered and only the 2*n* better members (here "better" means the 2*n* lower values of *f*, $g=\left\|f-\tilde{f}\right\|_2$) remain to the population, the other are deleted. This is the so-called natural selection. By repeated iterations of reproduction (under crossover and mutation) and natural selection we can find the minimum of $g=\left\|f-\tilde{f}\right\|_2$ as the point to which the best values of our population converge. The termination criterion is: "the mean value of *f* in the population is no longer improved". The algorithm is summarized as follows

STEP A: Find (randomly) the initial population of 2n members
STEP B: Split the population (randomly) into n pairs
STEP C: Make c crossovers and from each pair of parents take k children. Every bit of every child has P probability for a mutation
STEP D: Find the new population 2n+2k (*parents+children*)
STEP E: From the new population select the 2n members with the lower values of f.
STEP F: If the absolute value of f is < å, then STOP, otherwise go to STEP B.

The present GA is the basic GA and one can use further more sophisticated schemata. In many cases GAs find the global minimum of the minimization problem in question, in spite of the fact of its slow-convergence speed.

4. Example

The Example refers to a 2-D polynomial which can be, for example, the characteristic polynomial of a 2-D filter.

$$f(z_1,z_2)=z_1^2+z_2^2+2.4z_1z_2+5.25z_1+5.4z_2+6.3 \qquad (7)$$

After the calculations, it is seen that the above necessary and sufficient conditions (Eq.(3)) for factorization into linear 2-D factors are not satisfied. Therefore, the approximation of $f(z_1,\ldots,z_m)$ by the factorizable polynomial $\tilde{f}(z_1,\ldots,z_m)$ is attempted:

$$\tilde{f}(z_1,z_2)=\prod_{i=1}^{2}\left(z_1+\tilde{a}_{i,2}+\tilde{c}_i\right)$$

or in a simpler notation

$$\tilde{f}(z_1, z_2) = (z_1 + qz_2 + p)(z_1 + uz_2 + r) =$$

$$= z_1^2 + quz_2^2 + (u+q)z_1 z_2 + (p+r)z_1 + (qr+up)z_2 + pr$$

where

$$p = \tilde{c}_1, \qquad q = \tilde{a}_{1,2} \qquad r = \tilde{c}_2 \qquad u = \tilde{a}_{2,2}$$

Therefore the minimum of $\left\| f - \tilde{f} \right\|_2$ or equivalently the minimum of $\left\| f - \tilde{f} \right\|_2^2$ is considered, where:

$$\left\| f - \tilde{f} \right\|_2^2 = (1 - qu)^2 + (2.4 - u - q)^2 +$$

$$+ (5.25 - p - r)^2 + (5.4 - qr - up)^2 + (6,3 - pr)^2$$

Using the Levenberg-Marquardt routine in [12] the following solution is obtained:

$$p\text{=}3.3262, \quad q\text{=}1.8217, \quad r\text{=}1.8966, \quad u\text{=}0.57775 \tag{8}$$

and

$$\left\| f - \tilde{f} \right\|_2^2 = 0.004 \tag{9}$$

This is the result found in a previous publication[12].

In the sequel, the previously described GA will be used with n=5, k=4, t=12, M=6, and P=0.01 and the following solution is obtained:

$$p\text{=}1.8787, \quad q\text{=}0.5537, \quad r\text{=}3.3578, \quad u\text{=}1.8702 \tag{10}$$

The evolution of *p, q, r, u* and their final convergence to the above values is shown in Fig. 2, 3, 4, and 5. Using these values for *p, q, r, s, t, u* we obtain:

$$\left\| f - \tilde{f} \right\|_2^2 = 0.0022 \tag{11}$$

which is an improvement compared with the result of [12]. In Table.1, the evolution and the convergence of the coefficients *p, q, r, u* and the optimum value of *f* in every generation is also shown.

Therefore

$$\tilde{f}(z_1, z_2) = (z_1 + 0.5537z_2 + 1.8787)(z_1 + 1.8702z_2 + 3.3578)$$

or in an expanded form:

$$\tilde{f}(z_1, z_2) = z_1^2 + 1.0355z_2^2 + 2.4239z_1z_2 + 5.2465z_1 + 5.3838z_2 + 6.3083$$

So, one can write $f \cong \tilde{f}$ i.e.

$$f(z_1, z_2) \cong z_1^2 + 1.0355z_2^2 + 2.4239z_1z_2 + 5.2465z_1 + 5.3838z_2 + 6.3083$$

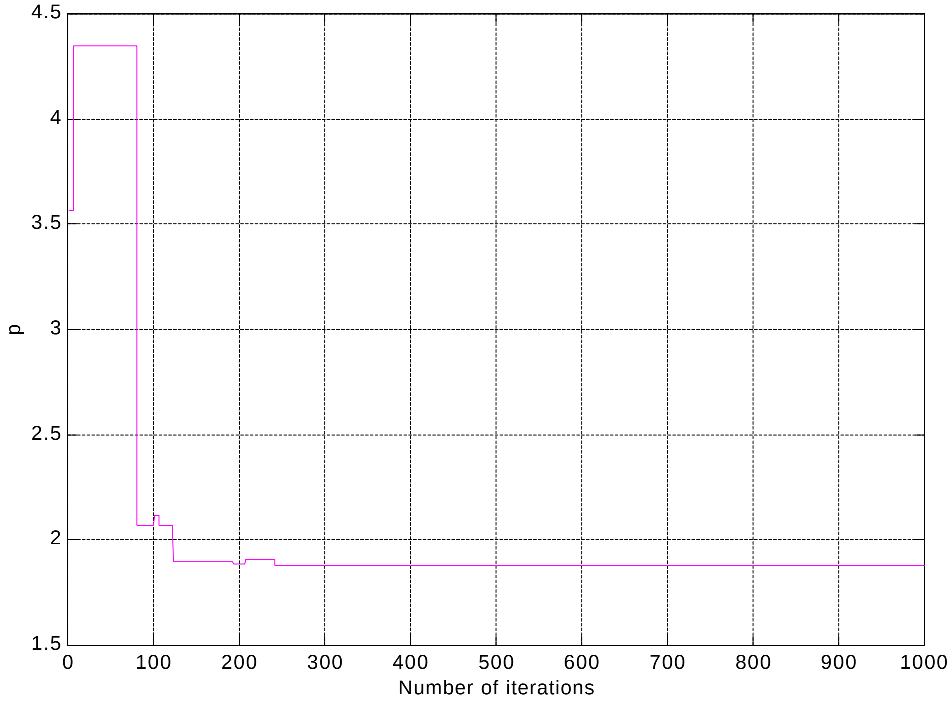

Fig.2. Convergence of the coefficient *p* in every generation

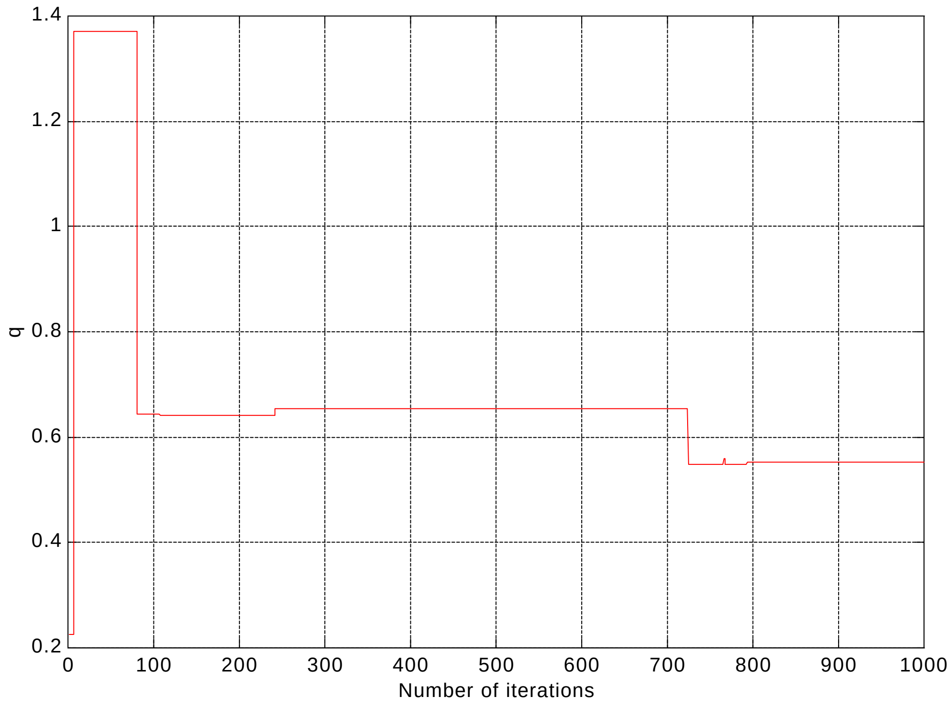

Fig.3. Convergence of the coefficient *q* in every generation

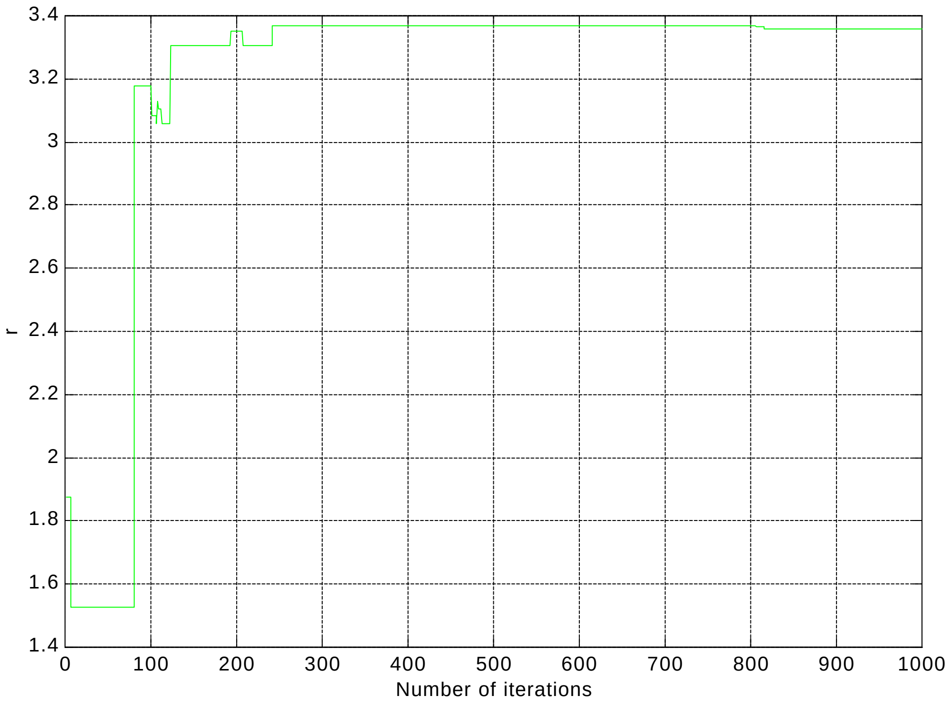

Fig.4. Convergence of the coefficient r in every generation

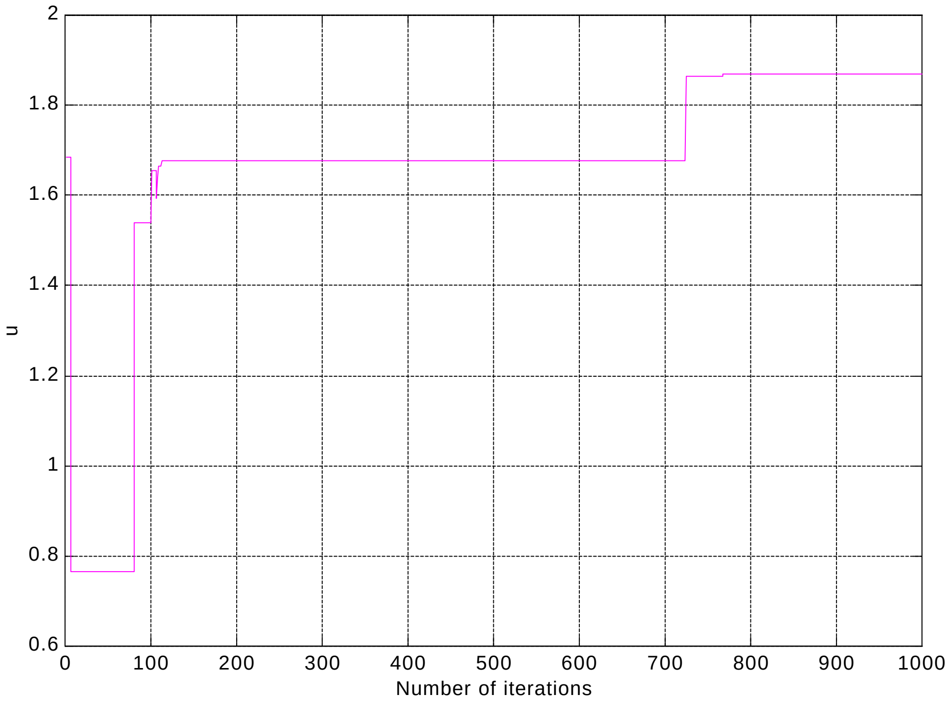

Fig.5. Convergence of the coefficient u in every generation

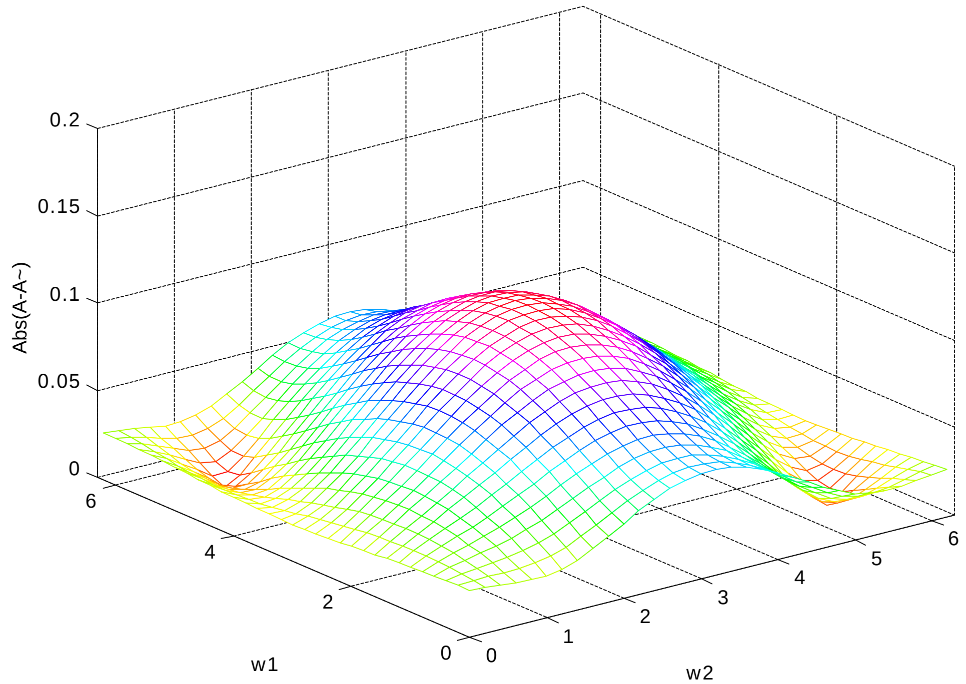

Fig.6. $Abs\left(f-\tilde{f}\right)$ versus w_1, w_2 : $w_1 \in [0, 2\pi]$, $w_2 \in [0, 2\pi]$. In the figure, *f*'s are denoted by A's

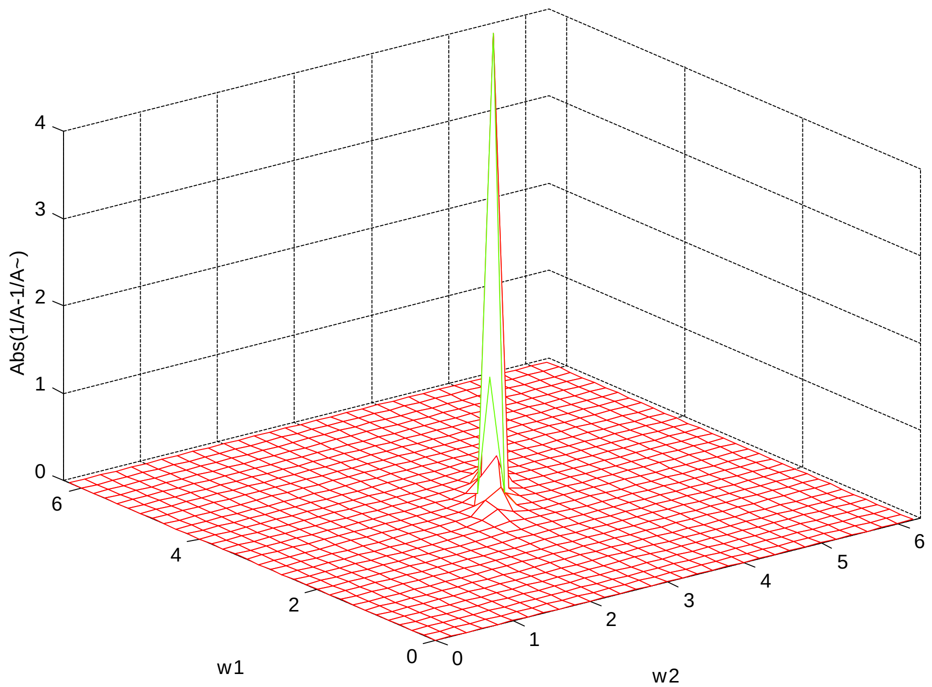

Fig.7. $Abs\left(\frac{1}{f}-\frac{1}{\tilde{f}}\right)$ versus w_1, w_2 : $w_1 \in [0, 2\pi]$, $w_2 \in [0, 2\pi]$. In the figure, *f*'s are denoted by A's

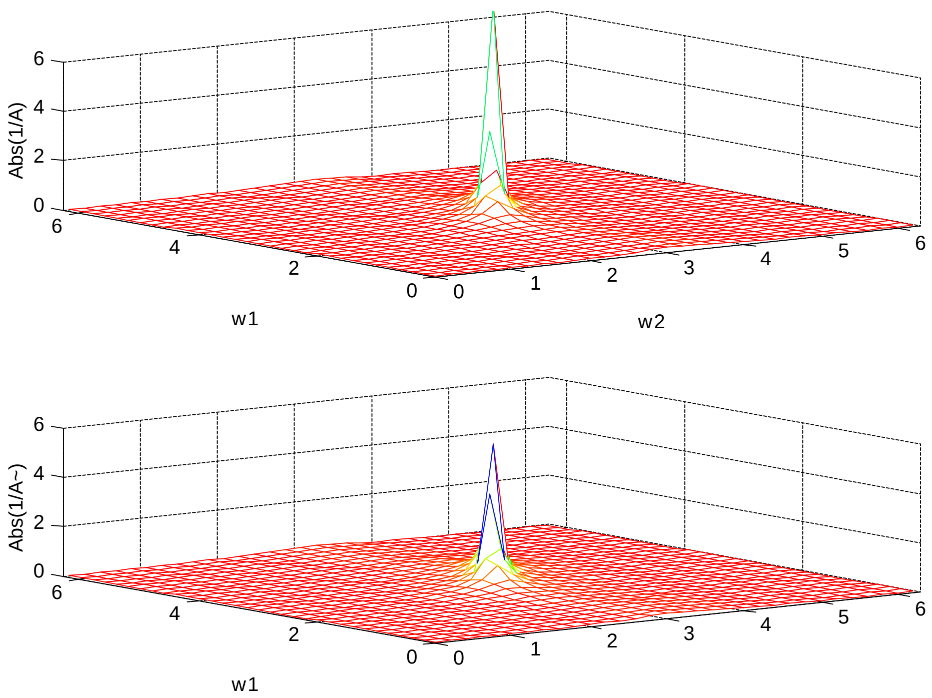

Fig.8. $Abs\left(\frac{1}{f}\right)$ and $Abs\left(\frac{1}{\tilde{f}}\right)$ versus w_1, w_2 : $w_1 \in [0, 2\pi]$, $w_2 \in [0, 2\pi]$. In the figure, *f*'s are denoted by A's

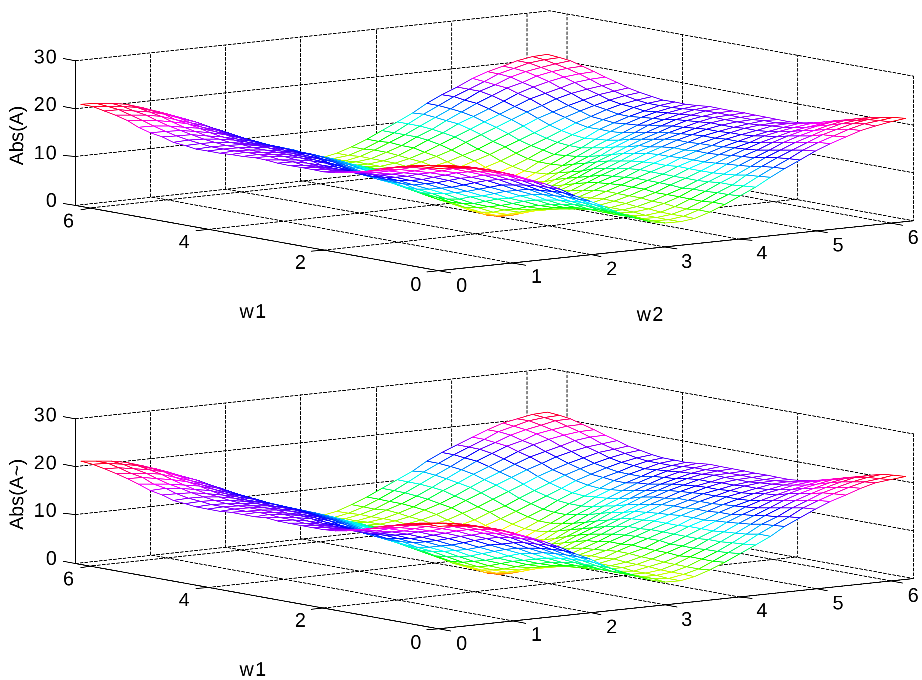

Fig.9. $Abs(f)$ and $Abs\left(\tilde{f}\right)$ versus w_1, w_2 : $w_1 \in [0, 2\pi]$, $w_2 \in [0, 2\pi]$. In the figure, *f*'s are denoted by A's

Iteration	p	q	r	u	Error	Average
1	3.56506348	0.22558594	1.87573242	1.68566895	1.87557664	398.43030321
50	4.34472656	1.36962891	1.52819824	0.76611328	0.57609493	1.43750536
100	2.07202148	0.64306641	3.17761230	1.53808594	0.15749219	0.79348546
150	1.89624023	0.64160156	3.30505371	1.67651367	0.02595842	0.02907346
200	1.88452148	0.64160156	3.35192871	1.67651367	0.02098866	0.02788834
250	1.87866211	0.65332031	3.36950684	1.67651367	0.01732266	0.02057745
300	1.87866211	0.65332031	3.36950684	1.67651367	0.01732266	0.01898460
350	1.87866211	0.65332031	3.36950684	1.67651367	0.01732266	0.01828951
400	1.87866211	0.65332031	3.36950684	1.67651367	0.01732266	0.01827482
450	1.87866211	0.65332031	3.36950684	1.67651367	0.01732266	0.01808567
500	1.87866211	0.65332031	3.36950684	1.67651367	0.01732266	0.01808567
550	1.87866211	0.65332031	3.36950684	1.67651367	0.01732266	0.01803991
600	1.87866211	0.65332031	3.36950684	1.67651367	0.01732266	0.01802975
650	1.87866211	0.65332031	3.36950684	1.67651367	0.01732266	0.01802975
700	1.87866211	0.65332031	3.36950684	1.67651367	0.01732266	0.01792763
750	1.87866211	0.54785156	3.36950684	1.86401367	0.00422413	0.01639007
800	1.87866211	0.55371094	3.36950684	1.86987305	0.00317873	0.00375104
850	1.87866211	0.55371094	3.35778809	1.86987305	0.00283540	0.00306969
900	1.87866211	0.55371094	3.35778809	1.86987305	0.00283540	0.00304036
950	1.87866211	0.55371094	3.35778809	1.87023926	0.00282928	0.00290152
1000	1.87866211	0.55371094	3.35778809	1.87023926	0.00282928	0.00290152

Table.1. Convergence of the coefficients *p, q, r, u* every generation

5. Conclusion

In this paper, an *m*-D polynomial, which is not exactly factorized into linear m-D polynomial factors, is considered. This polynomial can be approximately factorized into linear m-D factors in the sense of the least square approach using Genetic Algorithms. It is known that in many cases GA's find the global minimum of the minimization problem in question, in spite of the fact of its slow-convergence speed. This technique of approximate multidimensional factorisation using GA's can be proved very useful in *m*-D filters' design (where the solution to the typical realization problem is a simple cascade realization). The same is true for the modern 2-D and 3-D networks theory as well as for the DPS (Distributed Parameter Systems) design since, in the most cases, the exact *m*-D factorization is impossible. Finally, some properties of m-D systems like controllability, observability and minimallity are considerably facilitated (even if the exact factorization cannot be achieved) by the approximate factorization.

References

[1] S.G.Tzafestas (ed.), "Multidimensional Systems: Techniques and Applications", Marcel-Dekker, New York, 1986.

[2] T.Kaczorek, "Two-Dimensional Linear Systems", Lecture Notes in Control and Information Sciences Vol.68, Springer-Verlag, Berlin, 1985.

[3] N.E.Mastorakis, "Multidimensional Polynomials: Algorithms", Ph.D. Thesis, National Technical University of Athens. 1992.

[4] N.J.Theodorou and S.G.Tzafestas, "Reducibility and factorizability of multivariable polynomials", Control-Theory and Advanc. Techn., Vol.1, 1985, pp.25-46.

[5] N.E.Mastorakis, S.G.Tzafestas and N.J.Theodorou, “A simple Multidimensional Polynomial Factorization method”, IMACS-IFAC Intern. Symp. on Math and Intelligent Models in System Simulation, pp.VII.B.1-1, Brussels, September 1990.

[6] N.E.Mastorakis and N.J.Theodorou, “State-Space Model Factorization in *m*-Dimensions. Application in Stability”, Found. Comp. Decision Science, Vol.17, No.1, pp.55-61, 1992.

[7] N.E.Mastorakis and N.J.Theodorou, “Operators Method for m-D Polynomials Factorization”, Found. Comp. Decision Science, Vol.15, No.3-4, pp.159-172, 1990.

[8] N.E.Mastorakis, S.G.Tzafestas and N.J.Theodorou, “A Reduction Method for Multivariable Polynomial Factorization”, International Symposium on Signal Processing, Robotics and Neural Networks (SPRANN-94), IMACS-IEEE. Lille - France, Proceedings Session 2-D Systems, April 1994.

[9] N.E.Mastorakis, N.J.Theodorou and S.G.Tzafestas, “Multidimensional Polynomial Factorization in Linear m-D Factors”, Int. Journ. of Syst. Sci., Vol.23 No.11, pp.1805-1824, 1992.

[10] N.E.Mastorakis and N.J.Theodorou, “Approximate Factorization of Multidimensional Polynomials Depending on a Parameter l ”, Bulleting of Electronics of the Polish Academy, Vol.40,No.1, pp.47-51, 1992.

[11] N.E.Mastorakis and N.J.Theodorou: “Exact and Approximate Multidimensional Polynomial Factorization. Application on Measurement Processing”. Foundations of computing and decision science, 1994.

[12] Nikos E. Mastorakis, “Approximate Multidimensional (*m*-D) Polynomial Factorization into Linear *m*-D Polynomial Factors”', IMACS – IEEE: CESA ’96, Lille, France, July 9-12,1996.