Purdue University
Purdue e-Pubs

ECE Technical Reports | Electrical and Computer Engineering

2-1-1997

On Power Reduction of FIR Digital Filters Using Constrained Least Squares Solution

Khurram Muhammad
Purdue University School of Electrical and Computer Engineering

Kaushik Roy
Purdue University School of Electrical and Computer Engineering

Follow this and additional works at: http://docs.lib.purdue.edu/ecetr

Muhammad, Khurram and Roy, Kaushik, "On Power Reduction of FIR Digital Filters Using Constrained Least Squares Solution" (1997). *ECE Technical Reports.* Paper 76.
http://docs.lib.purdue.edu/ecetr/76

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.

On Power Reduction of FIR Digital Filters Using Constrained Least Squares Solution

Khurram Muhammad
Kaushik Roy

TR-ECE 97-3
February 1997

School of Electrical
and Computer Engineering
Purdue University
West Lafayette, Indiana 47907-1285

On Power Reduction of FIR Digital Filters Using Constrained Least Squares Solution*

Khurram Muhammad and Kaushik Roy
khurram@ecn.purdue.edu and kaushik@ecn.purdue.edu
School of Electrical and Computer Engineering
Purdue University, West Lafayette
IN 47907-1285, USA.

Contact Person: Kaushik Roy
School of Electrical and Computer Engineering
Purdue University, West Lafayette
IN 47907-1285, USA.
Ph: 317-494-2361, Fax: 317-494-3371
e-mail: kaushik@ecn.purdue.edu

'This research was supported in part by ARPA F33615-95-C-1625), by NSF CAREER award (9501869-MIP), Rockwell Corp., and IBM Corp.

On Power Reduction of FIR Digital Filters Using Constrained Least Squares Solution

ABSTRACT

In this report, we apply constraint least squares solution (CLS) to the problem of reducing the number of operations in FIR digital filters with a motivation of reducing its power consumption. The constraints are defined by the maximum allowable add/subtract operations in forming the products which are used in computing the output. We show that truncation and rounding of coefficients can be viewed as power constrained least squares (PCLS) solutions. Further, we show that in dedicated DSP processor based architectures it is possible to reduce power by using PCLS coefficients along with appropriately modified multipliers. It is also shown that Booth multiplier effectively reduces the complexity of such filters, thereby increasing power savings. Finally, we show that typically 30% and 45% retluction in number of operations can be obtained for systems employing uncoded and Booth recoded multipliers, respectively.

I. INTRODUCTION

Recently, *constrained least square* (CLS) design of FIR filters has been proposed [2, 3] which compromises peak stopband gain for reducing total stopband energy. Such filters find application in *frequency division multiplexed* (FDM) communication systems employing narrow frequency bands for multi-access. This report investigates reduction of power consumed by such filters for application in personal communication systems. The objective of this report is to explore reduction in complexity of the operation of such filters by defining constraints on the number of operations (and hence, power) and employing CLS technique to obtain coefficients that satisfy given power constraints. We will refer such filters as *power constrained least square* (PCLS) filters.

Low power implementations of FIR filters are of interest in wireless receivers and they have been investigated at various levels of abstractions in literature [4, 5, 6]. One approach targets programmable DSP architectures for identification of factors which contribute to dissipated energy and finding methods which reduce power hungry operations. Methods proposed for reducing power dissipated in the multipliers and busses of a generic Harvard architecture based DSP processor use a host of techniques such as *coefficient scaling, coefficient ordering, selective coefficient negation, removing common sub-expressions* [4]. These techniques attempt to reduce power by identifying operations that are "redundant" in the sense that they repeat computational steps that do not yield new information by executing a power-consuming operation. Bus power reduction is proposed by *coefficient optimization* [5] which attempts to reduce the *Hamming distance* (HD) between successive co-efficients in order to reduce the activity of signals at one of the multiplier inputs. *Parallel processing architectures* have also been proposed for reducing power of these filters [6].

FIR filters have traditionally been designed using the Parks-McClellan algorithm [1] due to its simplicity and wide availability. This widely used algorithm is based on a minimax optimality criterion which minimizes the maximum amplitude distortion of the signal in the entire band. However, these filters have high stopband energies. Consequently, such filters are not preferable for multi-access communication systems [2, 3]. In contrast, filters based on *unconstrained least squares* (ULS) criterion have relatively small stopband energy which is advantageous in multi-access communication system based on *frequency division multiplex* (FDM). However, these filters have large gains at the edge of their stopbands, which causes distortion due to aliasing of the signal from the adjacent channel.

A compromise between the two extremes is obtained by employing the CLS solution which relaxes the peak distortion in the stopband for a large reduction in stopband energy [2, 3]. These methods provide a viable alternative in design of filters suitable for specific applications. In a portable computing and/or wireless communication scenario, CLS based methods provide filters that can provide better performance in terms of attenuation characteristics and rejection of energy in the undesired band [2, 3].

In this report, we approach the issue of power reduction from a different angle. We explore

reduction in complexity of FIR filters by reducing the switching activity required to compute the output. This is based on the premise that lower complexity leads to lower power dissipation. Our approach is to explore constrained least squares solution similar to [2, 3] by defining constraints on the number of additions in computing the products that yield the filter output. The philosophy is to compute a constrained coefficient vector which is nearest to a known optimal vector in a LS sense constrained by a maximum number of allowable add/subtract operations in computing the products. The maximum allowable add/subtract operations can be viewed as a constraint on power if appropriate multipliers discussed in section II-B are employed. Hence, this approach yields a PCLS solution.

We will show that if the original optimal coefficient vector is computed based on CLS solutions similar to [2, 3], the PCLS vector is the optimal power constrained coefficient vector for the filter satisfying the given specifications. For DSP processor based filter implementations, we show that PCLS filters can significantly reduce the complexity (and hence, power) without violating performance constraints. Further, based on the definition of *least square error* (LSE) a PCLS based approach yields a coefficient vector that is a rounded or a truncated coefficient vector. Alternatively, rounding or truncation can be viewed as generation of PCLS coefficients. The degree of rounding or truncation is dictated by the power constraints. Further, it is shown that reducing complexity by computing PCLS coefficients achieve reduction in power at the expense of increased energy in the stopband.

II. DEFINITIONS

Let F_p and F_s represent the passband and stopband frequencies, respectivelly, and A define the nominal passband gain. A_{min} and A_{max} define the minimum and maximum passband gains, respectively. Further, let δ_s define the peak stopband gain. Then

$$SPAR = 20\, log_{10} \frac{\delta_s}{A_{max}} \quad dB \tag{1}$$

is the *maxirrzum stopband to maximum passband gain* as defined in [2]. Further, let E_p and E_s define the passband and stopband energies [2]. Then

$$PSER = 10\, log_{10} \frac{E_p}{E_s} \quad dB \tag{2}$$

defines the *passband to stopband energy ratio.* Figure *1* shows a typical SPAR versus PSER curve for a FIR filter. As shown in the figure, the two extreme points on this curve correspond to Parks-McClellan and ULS solutions. The points between them correspond to CLS solutions that show a typical trade-off of SPAR and PSER [3]. For any given specifications on the desired impulse response of the FIR filter, we can construct a unique SPAR-PSAR tradeoff curve. It is noted that any FIR filter can be represented in the space comprising the positive quadrant of SPAR and PSER. We will show that the effect of constraining power (i.e. PCLS filter) is to move the filter away from

the SPAR-PSER tradeoff curve. When the constraints on power are removed, the desired filter moves back on the curve.

A. FIR *Filters* – Preliminaries

Consider a linear time-invariant FIR system of length M described by an input-output relationship of the form

$$y(n) = \sum_{i=0}^{M-1} \bar{b}_i \, x(n-i) \tag{3}$$

In this context, $\bar{b}_i$ represents the ith coefficient whereas $x(n - \mathrm{i})$ denotes the data sample at time instant n – i. Note that the over score is used to remind the reader that although $\bar{b}_i$ is a scalar quantity, it can be decomposed in to constituent bits. Hence, the coefficient vector $\mathrm{b} = [\bar{b}_0, \bar{b}_1, \ldots, \bar{b}_{M-1}]$. Without any loss of generality, we will assume that all coefficients are integers represented by N bits. In sign-magnitude form, the representation of integers is expressed as $\bar{b}_i = \sum_{j=0}^{N-2} 2^j \, b_{i,j}$ and the bit S at $(\mathrm{N} - 1)th$ position represents the sign of the number. It is 0 or 1 for positive or negative numbers, respectively. In twos complement notation an integer can be expressed as $\bar{b}_i = \sum_{j=0}^{N-2} 2^j \, b_{i,j} - S \, . \, 2^{N-1}$, where S is the sign bit defined above.

The system above can be implemented varying from the simplest form of a tapped-delay line or transversal filter to more robust cascade and lattice structures [1]. In this report, we will mainly concentrate on transversal FIR implementation that finds use in wide applications.

In a generic DSP implementation of a transversal FIR filter, the coefficients are held in a coefficient memory and are sequentially applied to the multiply-and-accumulate (MAC) unit. A separate memory holds data which is applied to the second input of the adder. A major source of power dissipation is the multiplier unit that computes $b_i \cdot x(n - \mathrm{i})$ for $\mathrm{i} = 0, 1, \ldots, \mathrm{M} - 1$. In a typical multiplier unit, each 1-bit of a multiplier corresponds to a shift-and-add (SAA) operation of the multiplicand. If the number of 1-bits or the Hamming Weight (HW) can be reduced in the multiplier, we can reduce the number of additions required to compute the product $b_i \, . \, x(n - \mathrm{i})$, thereby reducing power.

A commonly used multiplier unit employs the Booth's algorithm [7] for high speed and low power multiplication. The main idea is to recode the multiplier such that consecutive runs of 1-bits are represented by difference of two numbers each having only a single 1-bit. As an example, the sequence 11111111 can be represented as $100000000 - 1 = 1000000\bar{1}$ which uses two add and one subtract operations in calculating a product with this number. In contrast, the original multiplier would have caused eight add operations. Hence, recoded multiplier offers savings in power. Many variants of Booth recoding exist, most efficient of which is a technique called canonical *recoding* [7]. Non-canonical Booth recoding is more practical in implementation due to its simplicity [7, 8]. The difference between the two is illustrated by a simple example. Canonical recoding of 01110110111 yields $1000\bar{1}00\bar{1}00\bar{1}$ whereas non-canonical recoding gives $100\bar{1}10\bar{1}100\bar{1}$. The number of add/subtract operations are reduced from 8 to 4 in canonical and 6 in non-canonical recoding,

respectively. When using a Booth's multiplier [7, 8], although a simple reduction in HW would yield improvement in power, however, the hardware essentially remains under-utilized as many other possible coefficient codewords with a larger HW exist that would consume equivalent or lower power.

Table 1 shows the number of available coefficients as a function of word length, N, which consume equal power when used as multiplier for a given multiplicand. We will refer to the set of coefficients that would dissipate equal power as a *code class*. More formally, *two codes,* r *and* s *belong to the same code class if the amount of energy spent in forming the products* $p \cdot r$ *and* $p \cdot s$ *is the same for any arbitrary number* p. Table *1* shows the number of available code-words in different possible code classes when using an unrecoded or a canonical Booth recoded multiplier. The method for reducing power in this report is to constrain the coefficients to be a member of one of the specified code classes.

It is noted that there are two criteria for measuring the filter performance; the main criterion being its impulse response satisfying given specifications of maximum stopband attenuation and passband ripple, the other being its power dissipation performance. This work considers a LS solution constrained by the code class of the coefficients such that the modified coefficient vector is closest to the original vector b and all modified coefficients lie within given code classes. Further, the main criterion of a filter's performance is its adherence to specifications in frequency domain which must not be violated when complexity is reduced.

B. Hardware Considerations for Low Power

Power dissipation in a multiplier depends on the signal activity at the external and internal nodes. The signal activity at the external input nodes depend on the *Hamming Distance* (HD) between successively applied coefficients (or data) whereas the signal activity in the internal and external output nodes depends on several factors including the input signal probability. In a generic array multiplier consisting of rows of basic cells (see figure 2), restricting the code class of filter coefficients will not significantly reduce the activity of signal at the internal node;;.

In a typical DSP processor based system, an appropriate modification of generic multiplier is to "bypass" a cell row such that the contents of the row of cells are held at the previous values if the corresponding bit in the multiplier is a 0. The contents of a previous row of cells are simply passed over to the next row of the cells without changing any signal value in the internal nodes. This can be achieved by suitably modifying the *control* (CTRL) and *controlled/add/subtract/shift* (CASS) units in figure 2. As a result, the internal signal activity is reduced while forming the product. Without such a modification, the row of cells will be loaded with all 0's leading to a higher signal activity, thereby redeeming power optimization techniques useless.

In a dedicated architecture we can simply reduce the word length of the filter (coefficients. This would reduce the number of operations thereby saving power. Hence, for the two contrasting

Code Class	N=20		N=16		N=12	
	Unrecoded	Recoded	Unrecoded	Recoded	Unrecoded	Recoded
0	1	1	1	1	1	1
1	20	20	16	16	12	12
2	190	361	120	225	66	121
3	1140	3570	560	1638	220	570
4	4845	21760	1820	6864	495	1344
5	15504	84448	1820	6864	495	1344
6	38760	208208	4368	16632	792	1456
7	77520	315744	8008	22176	924	560
8	125970	274560	11440	14400	792	32
9	167960	119680	12870	3456	495	-
10	184756	19712	11440	128	220	-
11	167960	512	8008	-	66	-
12	125970	-	4368	-	12	-
13	77520	-	1820	-	1	-
14	38760	-	560	-	-	-
15	15504	-	120	-	-	-
16	4845	-	16	-	-	-
17	1140	-	1	-	-	-
18	190	-	-	-	-	-
19	20	-	-	-	-	-
20	1	-	-	-	-	-

Table 1: Distribution of binary codewords in various code classes for unrecoded and Booth recoded multipliers for $N = 20, 16$ and 12.

implementations (i.e. DSP processor based vs. dedicated hardware), complexity reduction takes different but equivalent forms.

For a general FIR filter, reducing the HD between successive coefficients severely distorts the transfer function of the filter and significant power savings using this method is hard to achieve. Reduction of required number of add/subtract operations, on the other hand can result in a proportional reduction in power. Hence, we will not consider the optimization of coefficients in order to reduce the HD between successive coefficients (or data). An interested reader is referred to [5] for more details.

III. CONSTRAINED LEAST SQUARES TECHNIQUE FOR FIR FILTER DESIGN

In this section, we will present the CLS approach used to compute the modified coefficient vector, $\mathbf{k} = [\bar{k}_0, \bar{k}_1, \ldots, \bar{k}_{M-1}]$. The original coefficient vector is represented by $\mathbf{b}$ and the maximum code class allowable is represented by κ. The vector $\mathbf{k}$ obtained using the minimization technique replaces $\mathbf{b}$ when computing equation **(3)** in the actual implementation. We will develop the CLS solution for two different LSE definitions, which will be referred to as *error definitions* **I** and **11,** respectively.

A. Brute *Force* CLS Solution for Error Definition *I*

Let $\mathrm{B} = [b_{0,0}, b_{0,1}, \ldots, b_{0,N-2}, b_{1,0}, b_{1,1}, \ldots, b_{1,N-2}, \ldots, b_{M-1,0}, \ldots, b_{M-1,N-2}]^T$ be the vector decomposition of bits representing the original coefficient vector. Similarly, $\mathrm{K} = [k_{0,0}, k_{0,1}, \ldots, k_{0,N-2}, k_{1,0}, k_{1,1}, \ldots, k_{1,N-2}, \ldots, k_{M-1,0}, \ldots, k_{M-1,N-2}]^T$ represents the vector decomposition of bits of the modified vector which satisfies given power constraints. Further, let $\mathrm{W} = diag[2^0, 2^1, \ldots, 2^{N-2}, 2^0, 2^1, \ldots, 2^{N-2}, \ldots, 2^0, \ldots, 2^{N-2}]^T$ define a weight vector. We define the LSE, ε as

$$\varepsilon = (\mathbf{B} - \mathbf{K})^T W\, W^T\, (\mathbf{B} - \mathbf{K}) \tag{4}$$

In the sequel the LSE defined above will be referred to as error definition *I*. Note that the error surface ε is convex as $\partial^2\varepsilon/\partial k_{i,j}^2 > 0$ and $\partial^2\varepsilon/\partial k_{i,j}\partial k_{l,m} = 0$ for $i \neq l, j \neq \mathrm{m}$. The constrained minimization problem can be specified as

$$\min_K \quad \varepsilon = \min_K\, (\mathrm{B} - \mathbf{K})^T W\, W^T\, (\mathrm{B} - \mathrm{K}) \tag{5}$$

$$s.t. \qquad \mathbf{C}^T\, K = \mathbf{c} \tag{6}$$

where C is a matrix defining constraints on K specified in $\boldsymbol{c}$. The matrix $\boldsymbol{C}$ is initialized to all 1's in the first column and the vector $\boldsymbol{c}$ is initialized to M_K. This corresponds to the original constraint that sum of 1's of elements of K should be equal to M_K. The Lagrangian, J is given as [10]

$$J = \frac{1}{2}(\mathbf{B} - \mathbf{K})^T W\, W^T\, (\mathbf{B} - \mathbf{K}) - (C^T K - M\kappa)\lambda \tag{7}$$

Taking the derivative of the equation above with respect to W, we get

$$\frac{\partial}{\partial W} J = WW^T(K - B) - C\lambda = 0 \tag{8}$$

$$K_{CLS} = (WW^T)^{-1}(C\lambda + WW^T B) \tag{9}$$

$$= (WW^T)^{-1}C\lambda + K_{LS} \tag{10}$$

where $K_{LS} = \mathrm{B}$ and hence, the constrained least squares solution is only a modification of the least squares solution [10]. The value of λ can be found by invoking the constraint

$$C^T K_{CLS} = C^T\{(WW^T)^{-1}C\lambda + K_{LS}\} = M\kappa \tag{11}$$

which on solving gives

$$K_{CLS} = K_{LS} + (WW^T)^{-1}C\{C^T(WW^T)^{-1}C\}^{-1}(M\kappa - C^T K_{LS}) \tag{12}$$

Equation (12) can be used to obtain the desired solution to constrained LS problem, however, it does not guarantee integer values for K_{CLS}. In fact, the solution obtained using this equation will almost always yields real numbers for bits composing K. As a consequence we need a branch-and-bound ILP problem formulation for obtaining integer values that comprise our solution.

B. The Branch-and-Bound Method for ILP Solution

We require the desired solution to be composed of integer values (0 or 1 only) for the bit variables. As B is assumed to be known, all $b_{i,j}, i = 0, 1, \ldots, M-1, j = 0, 1, \ldots, N-2$ are either 0 or 1. Since, the CLS framework does not guarantee an integer solution to all $k_{i,j}, i = 0, 1, \ldots, M-1, \; j = 0, 1, \ldots, N-2$ we can use an ILP formulation for solving this problem. The LP relaxation conditions can be specified as

$$-\infty < k_{i,j} < +\infty \qquad i = 0, 1, \ldots, M-1, \quad j = 0, 1, \ldots, N-2 \tag{13}$$

such that $\sum_{i=0}^{M-1} \sum_{j=0}^{N-2} k_{i,j} = M\kappa$. Note that this sum follows from Equation (6).

The branch-and-bound method (see figure 3) is a dynamic program which creates a tree by assigning all possible values to the variable under consideration and recalculates optimal solutions using Equation (12) for all assumed values [9]. The tree is traversed in a depth-first manner and the original problem is broken down to less complex subproblems. At the root of the tree an arbitrary variable is selected and assigned either a 0 or 1. This assignment adds a column in the constraint matrix **C** and modifies this matrix. For example, initial solution is computed using equation (12) where the constraint in equation (6) has only one column. An assignment of 1 to say, $k_{0,N-2}$ would introduce an additional column in equation (6) with a 1 at the bit position corresponding to $k_{0,N-2}$ and a 1 in an additional row in $\boldsymbol{c}$. A different constraint set is associated with each node in the tree and hence, each node has an associated solution.

Equation (12) yields optimal solution to the CLS problem which is checked for all integer values. If all variables assignments are integers, the current LSE is compared against the lowest LSE obtained so far (which is initially assumed to be ∞). If less, the current solution replaces the best one obtained so far (Subproblems 5 and 7 in figure 3), otherwise, the tree is truncated at that node and search for optimal solution is resumed at the next node in the depth-first tree. If all values of the solution are not integers, the current LSE is compared against LSE of best solution. If the current LSE exceeds the best LSE, the tree is truncated at this point as any further solution will only increase the LSE which will not yield a better solution than what we already have (Subproblem 6 in figure 3) and search resumes at the next node in the depth-first tree.

Figure 3 shows the branch-and-bound algorithm. It is noted that the order. of assumptions regarding $k_{i,j}$'s can significantly reduce the search required to obtain the optimal solution. The fastest solution is obtained if variable assignments of coefficients progress from *most significant bit* (MSB) towards the *least significant bit* (LSB). Further, any heuristically derived solution can serve as an initial guess and reduce computation in determining the solution.

C. An Efficient Algorithm for Computing the LS solution

The major problem of branch-and-bound ILP solution is its computational complexity. Note that the formulated LSE in equation (4) comprises a sum of squares of the difference in the components of the original and modified vectors. The power constraint requires that the code-class of the

components of coefficient vector stay close to κ while the LSE between the modified and original coefficient vectors must be minimum.

Consider the PCLS solution for numbers expressed in sign-magnitude representation. The ILP branch and bound method terminates exploring a subtree if it finds that assignments to all variables in K are either 0 or 1. The greatest contribution to LSE is provided by a mismatcln between $b_{i,N-2}$ and $k_{i,N-2}$, for $i = 0, 1, \ldots, M-1$ and is equal to $2^{2(N-2)}$. This is followed by a mismatch in the next lower significant bits. Hence, to minimize LSE in equation (4), we must select bits in K to replicate bits in B from most significant bits towards least significant ones. Further, the sign bits of coefficients in the original and modified vectors must be identical for differences to be as small as possible. Note that this operation is equivalent to truncation of the original coefficient vector such that the bit position where coefficients are truncated is determined by the given power constraint. This follows directly because LSE in equation (4) comprises a sum of squares in which each individual bit mismatch contributes a weighted component to the total error.

In the present framework, it is possible that there is no bit position at which truncation of coefficients *exactly* meets the power constraint. In such a case, truncation at one bit location gives a coefficient vector which consumes lower power than allowable, whereas truncation at the next bit position results in violation of the constraint. Hence, we have a situation in which we must select L out of M bits such that $L < M$ and each selected bit contributes exactly the same amount of error in the LSE. In such a case, we propose selection of most sensitive bits where the *sensitivity of a bit is defined as the contribution to error between the transfer functions of the original coefficient vector and a modified coefficient vector with the bit turned on in addition to the preselected higher significant bits.* The algorithm is presented as follows.

copy the sign bit of b; to k;, for $0 \leq i \leq M-1$

set $i = MSB$ and $sum = 0$

while $sum + \sum_{j=0}^{M-1} b_i^j < M\kappa$ **do**

  set $sum = sum + \sum_{j=0}^{M-1} b_i^j$

  copy bit j of b_i to k_i and set $i = i - 1$

end while

compute and store sensitivities and j for b_i^j, $s.t.\ b_i^j \neq 0$, $0 \leq j \leq M-1$

sort sensitivities of $b_i^j, 0 \leq j \leq M-1$ in decreasing order

select the ith bit of b_i^j's from the first $M\kappa - sum$ entries in this table

Note that as the above algorithm uses truncation, it can introduce a bias in the output which can be easily removed by adding an extra overhead of one multiply operation. As the length of filter increases, the effect of this extra overhead operation become less significant and can be ignored for practical values of filter lengths.

Finally, the algorithm above can be easily modified on Booth recoded multiplier. In this case, the original coefficients are replaced by Booth recoded coefficients and the same algorithm is used to select the low power Booth recoded coefficients. The results obtained are uncoded back to original form. Note that Booth recoding increase the number of code-words available in power constrained solution. Hence, significantly more complexity reduction is expected when using these multipliers.

D. PCLS Solution for Error Definition II

The LSE definition formulated in equation (4) is minimized by using truncation as means of reducing power consumption by eliminating add/subtract operations. However, other LSE formulations are also possible. We define

$$\varepsilon^* = (\mathbf{w}^T\,\mathbf{b} - \mathbf{w}^T\,\mathbf{k})\,(\mathbf{w}^T\,\mathbf{b} - \mathbf{w}^T\,\mathbf{k})^T \tag{14}$$

$$= (\mathbf{b} - \mathbf{k})^T\,\mathbf{w}\,\mathbf{w}^T(\mathbf{b} - \mathbf{k}) \tag{15}$$

where $\mathrm{w} = [2^0, 2^1, \ldots, 2^{N-2}]$. Note that this is in contrast with the definition in equation (4). This sum comprises the square errors of vector components comprising the differences $e; = \bar{b}_i - \bar{k}_i$ for $0 \leq i \leq \mathrm{M} - 1$. To minimize this sum, we must minimize the difference between the individual components of the two vectors given the power constraint. Clearly, the CLS solution for obtained for this framework is different from the one presented earlier. As an example, the closest number to 00111111 given $\kappa = 1$ is 01000000 rather than 00100000 which would be selected 'by the algorithm in the previous section for a multiplier using no Booth recoding.

The algorithm presented in III-C can be easily modified to find the PCLS solution for the new error formulation. The difference $(\bar{b}_i - \bar{k}_i)^2$ for $0 \leq \mathrm{i} \leq \mathrm{M} - 1$, given $\kappa = \infty$ (no power constraint) is minimized by selecting $\bar{b}_i = \bar{k}_i$. Again, we observe that the minimum contribution to LSE due to $(\bar{b}_i = \bar{k}_i)^2$ occurs when $\bar{b}_i$ is closest to $\bar{k}_i$ and it does not violate the power constraint. This is achieved by rounding $\bar{k}_i$ to bit position such that κ bits are 1's. Note, however, that this strategy may not yield the minimum LSE as in one coefficient we may discard bits at higher significant positions and select ones in another coefficient at lower significant positions. For example, given $\mathrm{M} = 2$ and $\kappa = 2$, 01010111 is rounded to 01010000 and 00000011 is rounded to 00000011. The LSE is 49. In contrast LSE due to rounding 01011000 and 00000100 is only 2. A simple rounding scheme will not ensure smallest LSE given that $2\kappa = 4$.

The algorithm for finding PCLS is given below. In the algorithm $round(\bar{b}_j, \mathrm{i})$ returns $\bar{b}_j$ rounded at bit position i. The sign bits are simply copied. Moving from MSB towards LSB, we keep on rounding the original coefficients at bit position i, $1 \leq \mathrm{i} \leq \mathrm{M} - 1$ until we reach a position j such that the power constraint is not met at j and is violated at $\mathrm{j} + 1$. As in previously presented algorithm, we propose using the sensitivities of bits at $\mathrm{j} + 1$ computed in the frequency domain for selecting the remaining bits. We note that the PCLS solution based on error definition II is obtained by mere rounding. Hence, *for an original filter obtained using a LS criterion, appropriate rounding yields the optimal PCLS filter.*

copy the sign bit of $\bar{b}_i$ to $\bar{k}_i$, for $0 \leq i \leq M-1$

set $i = MSB$ and sum $= 0$

set $\bar{t}_j = round(\bar{b}_j, i), 0 \leq j \leq M-1$

while sum $+ \sum_{j=0}^{M-1} \bar{t}_i^j < M_K$ **do**

  set sum = sum $+ \sum_{j=0}^{M-1} \bar{t}_i^j$

  set $i = i - 1$

  set $\bar{t}_j = round(\bar{b}_j, i), 0 \leq j \leq M-1$

end while

set $k_j = round(\bar{b}_j, i), 0 \leq j \leq M-1$

compute and store sensitivities and j for b_i^j, $s.t.\ b_i^j \neq 0$, $0 \leq j \leq M-1$

sort sensitivities of $b_i^j, 0 \leq j \leq M-1$ in decreasing order

set $\bar{k}_j = round(\bar{b}_j, i)$ from the first $M_K - sum$ entries in this table

IV. PERFORMANCE OF CLS FIR FILTER COEFFICIENTS

In this section we present some results using the proposed complexity reduction techniques. We will show that complexity reduction using the proposed PCLS approach yields filters with acceptable performance and results in power savings as a consequence of reduced complexity. The results are presented for filters employing both unrecoded and Booth recoded multipliers.

A. Array *Multipliers*

Figure 4 shows the PCLS coefficients obtained using error definition I for a 51 tap linear phase least squares FIR filter with $F_p = 1/4$, $F_s = 3/10$, $R_p = 3$ dB, $R_s = -57$ dB. Throughout this section, we assume that the original coefficients are expressed as 16-bit sign magnitude numbers and the filter is implemented using a DSP processor. The PCLS coefficients were obtained for 35% reduction in complexity. Clearly, the PCLS coefficients satisfy the performance constraints, however, they reduce the complexity of the filter substantially thereby reducing its power consumption. Figure 5 shows the PCLS coefficients for the filter with same specifications obtained using Parks-McClellan algorithm. The algorithm yields a filter with 33 taps. The complexity reduction of this filter is 28%. Clearly, PCLS coefficients do not substantially affect the filter performance by reducing its complexity.

Figure 6 shows the frequency response of a 85 tap LS filter obtained for $F_p = 11/40$, $F_s = 3/10$, $R_p = 3$ dB, $R_s = -50$ dB. The PCLS coefficients for this figure have been obtained for error definition II reducing the complexity by 31% when the original coefficients are assumed to be represented by 16 bits. Note that the PCLS coefficients give acceptable filter performance, however, it increases the PSER. Refer to figure 1 and note that the effect of constraining the complexity is to move on a horizontal line with SPAR fixed and varying PSER. Further, PCLS coefficients are

optimal coefficients for LSE given by ε^* as the original optimal coefficients have also been obtained using LS solution. The pole-zero plot of this filter is shown in figure 7. Note that the zeros of the filter are moved from their original locations, however, the filter specifications are not violated. Relaxing the power constraints moves the zeros back to their original locations. Further, the higher is the order of the filter, the more tightly clustered its zeros are, and hence, we need more bits to distinguish between them. However, this result reveals that the complexity of DSP processor based filter implementations can be reduced using PCLS coefficients and appropriate multiplier structures.

Figure 8 shows the frequency response of 201 tap LS FIR filter. $F_p = 1/10$, $F_s = 9/80$, $R_p = 3$ dB, $R_s = -55$ dB. Figure 9 shows the pole-zero plot for this filter. The complexity reduction is about 31%, however, the specifications of the original filter are not violated by the PCLS filter.

Finally, we observe that complexity of FIR filters can be further reduced by numerically robust implementations such as the cascaded form. As compared to transversal implementations the transfer characteristics of cascaded implementations remain closer to the original characteristics when numerical roundoff or truncation is applied [1]. Hence, numerically robust implementations are more amenable to complexity reduction techniques presented in this report.

B. Booth Multiplier

Earlier we pointed out that Booth multiplier significantly increases the code-words available for a given code class. Hence, use of this multiplier can further reduce the power requirements of the filter as it allows more code-words to be used without significantly increasing the power.

Figure 10 shows the frequency transfer characteristics of the filter of figure 4 with PCLS coefficients obtained for error definition I. The savings in complexity for this filter is 45% in comparison to 35% obtained for unrecoded multiplier. Clearly, Booth multiplier helps further reduce the power consumption without violating the filter specifications. Note that the multiplier in this example uses non-canonical recoding.

Figure 11 shows the frequency response of the 85-tap filter shown in figure 6 using PCLS coefficients for error definition I and non-canonical Booth recoding. These coefficients yield a 46% savings in complexity by reducing the number of add/subtract operations in contrast to 31% savings obtained earlier. Further, in contrast to the unrecoded multiplier, the PSER for this filter is better than the one shown in figure 6.

Finally, in figure 12 we show the frequency response of the PCLS coefficient 201-tap filter of figure 8. The savings in complexity for this filter is 46.5%. As the original filter is a LS filter, the constrained solution yields the optimal solution given the power constraints. Again, we note that in contrast to 31% savings in the unrecoded multiplier the reduction in complexity when using Booth multiplier is significantly higher. Table 2 summarizes the complexity reduction using PCLS coefficients for the example filters considered.

V. CONCLUSION

In this report, we approach the problem of reducing the power consumption of FIR digital filters by formulating strategy for finding constrained least squares solution and defining constraints as the maximum allowable add/subtract operations in forming the products in equation (3). We show that in dedicated DSP processor based architectures it is possible to reduce power by reducing the complexity of the filters using PCLS coefficients and appropriately modified multipliers. Further, Booth multiplier reduces the complexity of the filter even further, thereby increasing the power savings. In dedicated implementations, this translates to reducing the word length of the filter coefficients. Further, we develop a mathematical frame work in which complexity of filters can be expressed as a CLS problem and show that rounding and truncation can be viewed as means of obtaining PCLS coefficients depending on how we choose the LSE.

References

[1] J. G. Proakis and D. G. Manolakis, "*Digital* Signal Processing: Principles, Algorithms, and *Applications*,". *McMillan* Publishing Company, New York, 1992.

[2] I. W. Selesnick, M. Lang, and C. S. Burus, "Constrained Least Square Design of FIR Filters without Specified Transition Bands," IEEE Trans. Signal Processing, Vol. 44, No. 8, pp. 1879–1892, Aug. 1996.

[3] J. W. Adams, "FIR Digital Filters with Least-Squares Stopbands Subject to Peak-Gain Constraints.,"IEEE Trans. Circuits and Systems, Vol. 39, No. 4, pp. 376–388, Apr. 1991.

[4] M. Mehendale, SD. Sherlekar, G. Venkatesh, "Techniques for Low Power Realization of FIR Filters," Proc. 1995 Asia and South Pacific Design, pp. 447-450, Aug. 1995.

[5] M. Mehendale, SD. Sherlekar, G. Venkatesh, "Coefficient Optimization for Low Power Realization of FIR," Proc. 1995 IEEE Workshop on VLSI Signal Processing, pp. 352-361, Oct. 1995.

[6] D. N. Pearson, K. K. Parhi, "Low-power FIR Digital Filter Architectures," *Proc.* 1995 IEEE International Symposium on Circuits and Systems-ISCAS 95, pp. 231-234, Apr. 1995.

[7] I. Koren, "Computer Arithmetic Algorithms," Pentice-Hall, Englewoods, NJ, 1993.

[8] N. H. E. Weste, K. Eshraghian, "Principles of CMOS VLSI Design: A Systems Perspective," 2nd Edition, Addison Wesley, 1993.

[9] W. L. Winston, "Introduction to Mathematical Programming: Applications and Algorithms," 2nd Edition, ITP, 1995.

[10] L. L. Scharf, "Statistical Signal Processing: Detection, Estimation, and Time Series Analysis," Addison Wesley, 1991.

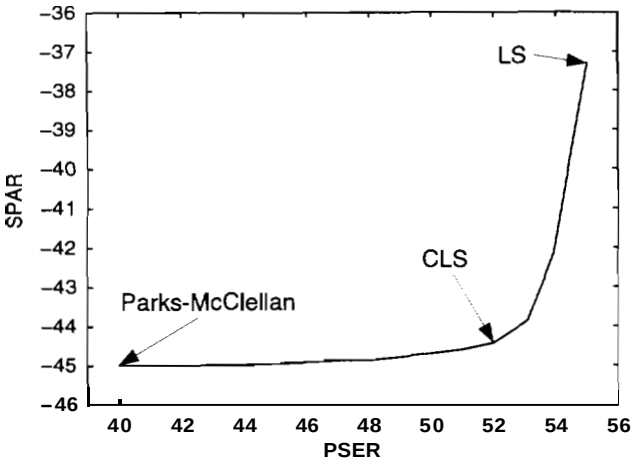

Figure 1: Typical SPAR-PSER curve for FIR filters

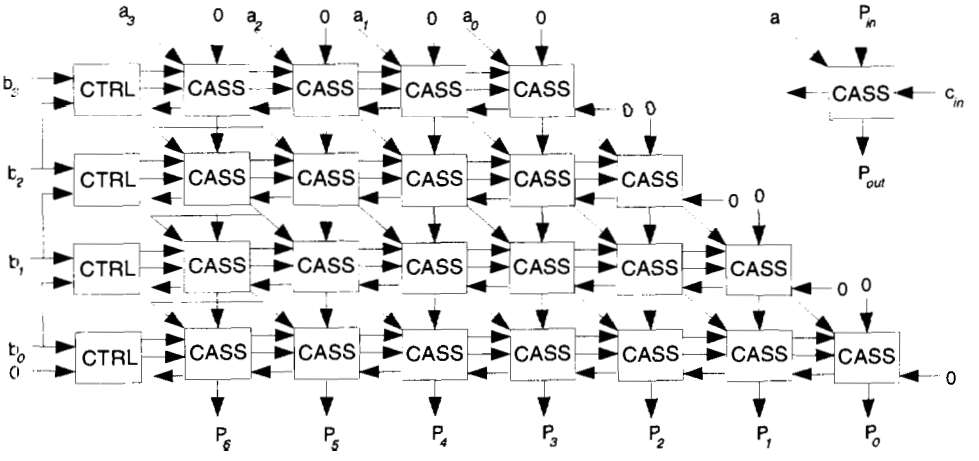

Figure 2: A generic Booth's algorithm array multiplier

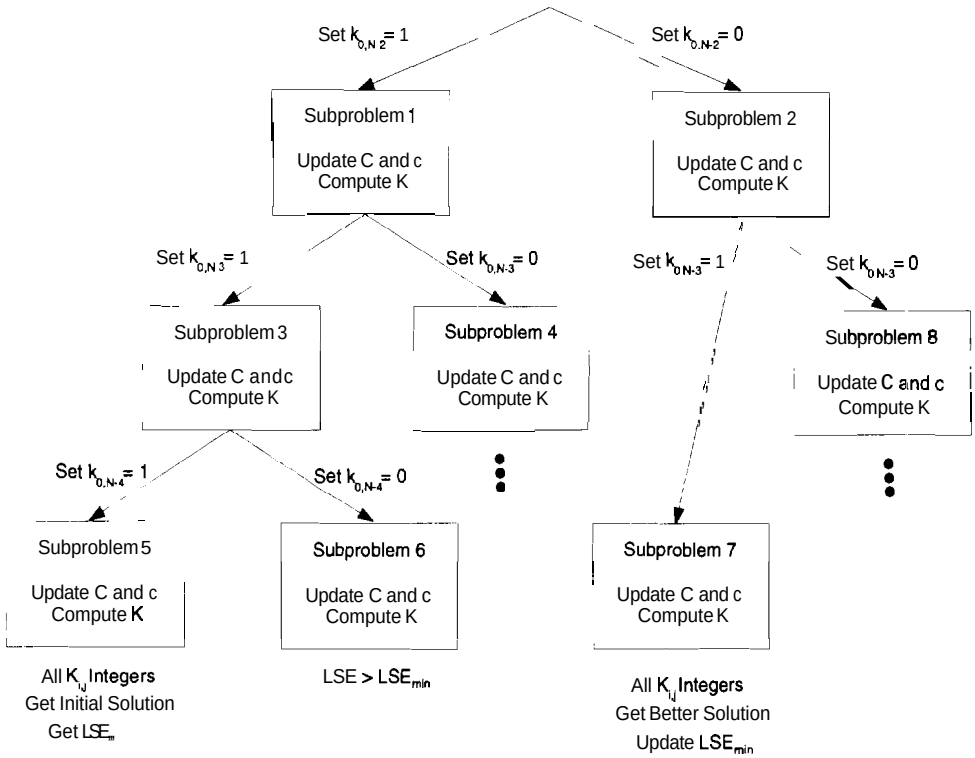

Figure 3: Branch and bound technique for ILP solution

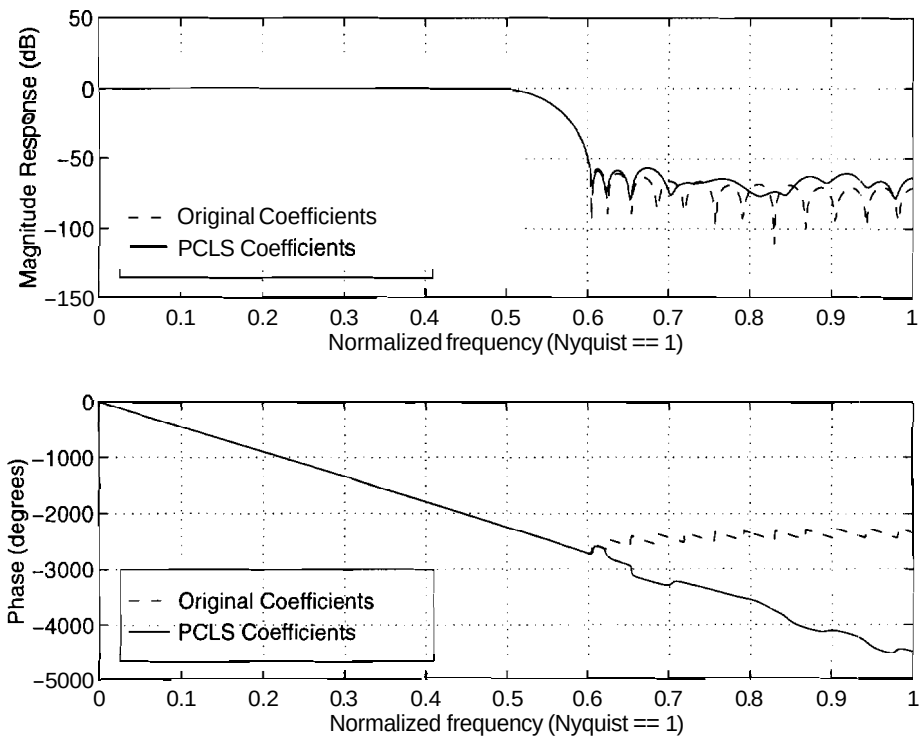

Figure 4: Response of PCLS vs. original coefficients for a 51 tap linear phase least squares FIR filter with $F_p = 1/4$, $F_s = 3/10$, $R_p = 3$ dB, $R_s = -57$ dB.

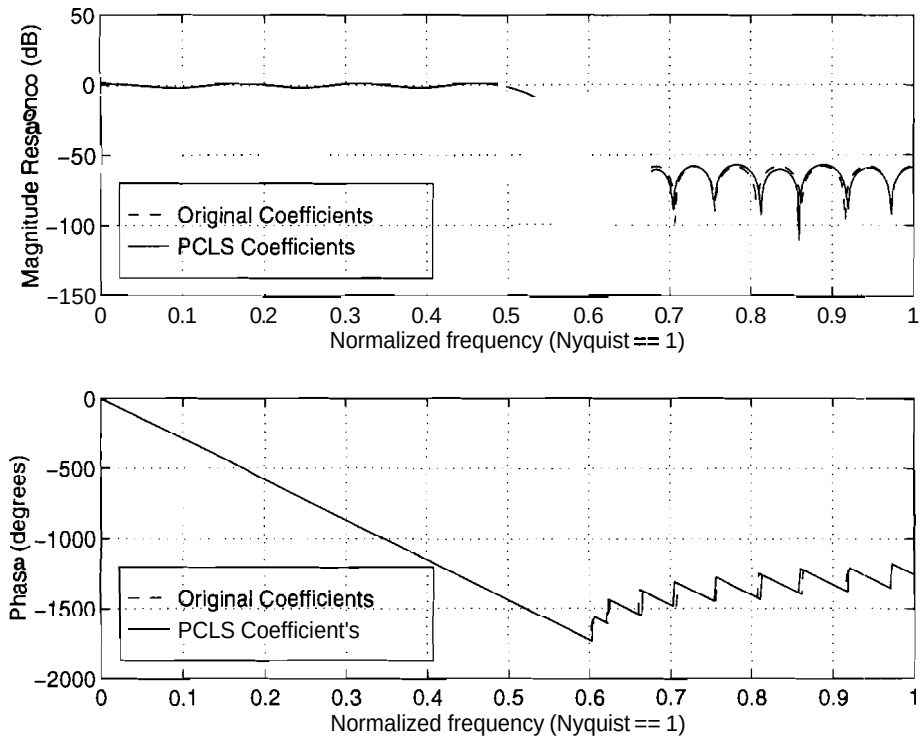

Figure 5: Response of PCLS vs. original coefficients for a 33-tap filter with specifications as in figure 4. The original filter was designed using Parks-McClellan algorithm.

Filter Specification	Error Definition	Complexity Reduction	
		Unrecoded	Booth Recoded
figure 4	I	35%	45%
figure 5	I	28%	44%
figure 6	II	31%	46%
figure 8	II	31%	46.5%

Table 2: Summary of reduction in number of operations using the PCLS technique.

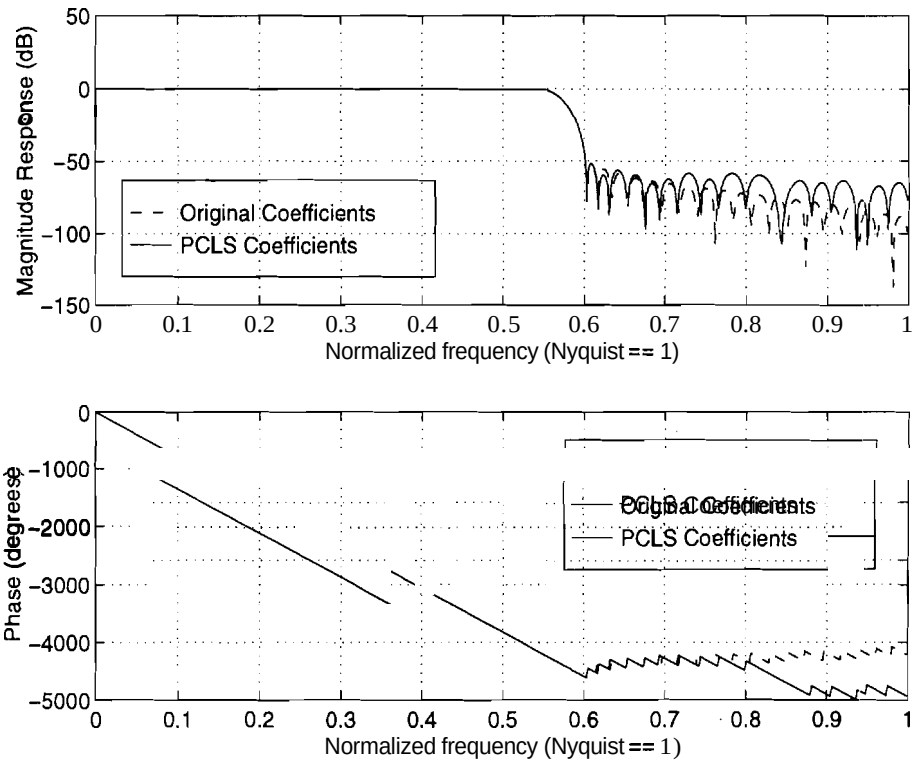

Figure 6: Frequency response of PCLS coefficients obtained using error definition II for a 85 tap LS filter. $F_p = 11/40$, $F_s = 3/10$, $R_p = 3$ dB, $R_s = -50$ dB.

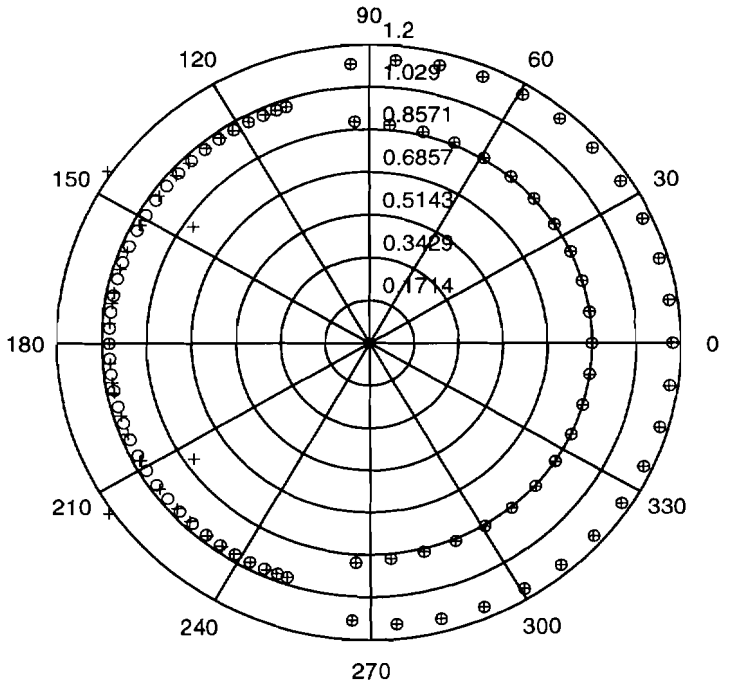

Figure 7: Pole-zero plot of the filter in figure 6 for original and PCLS coefficients. Zeros for original coefficients are represented by 'o' whereas zeros for PCLS coefficient vector are shown by '+'

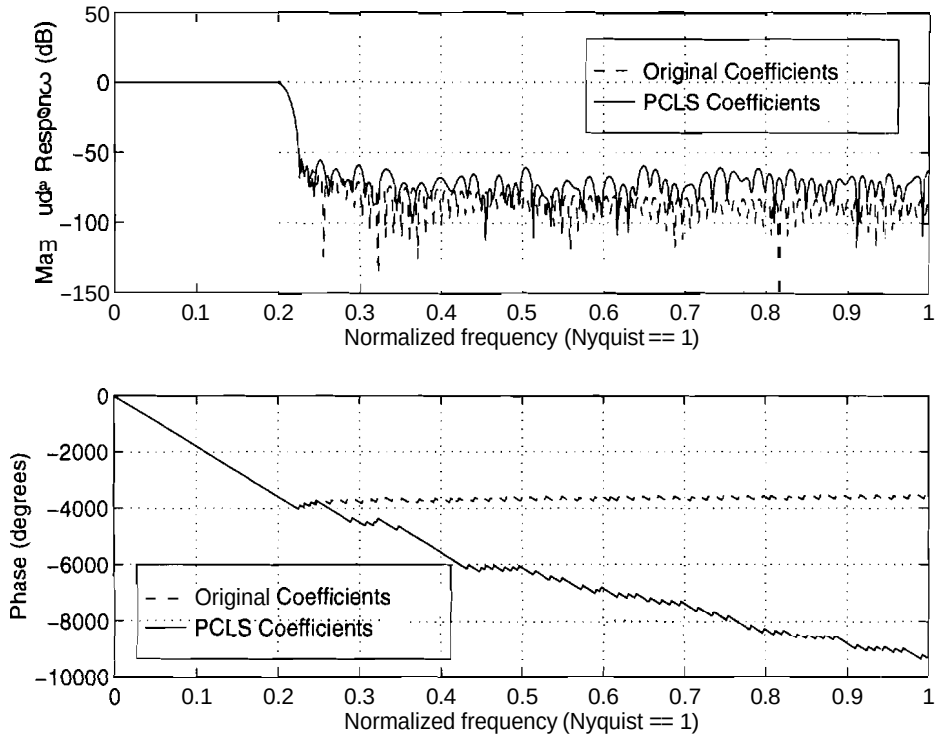

Figure 8: Frequency response of PCLS coefficients obtained using error definition II for a 201 tap LS FIR filter. $F_p = 1/10$, $F_s = 9/80$, $R_p = 3$ dB, $R_s = -55$ dB.

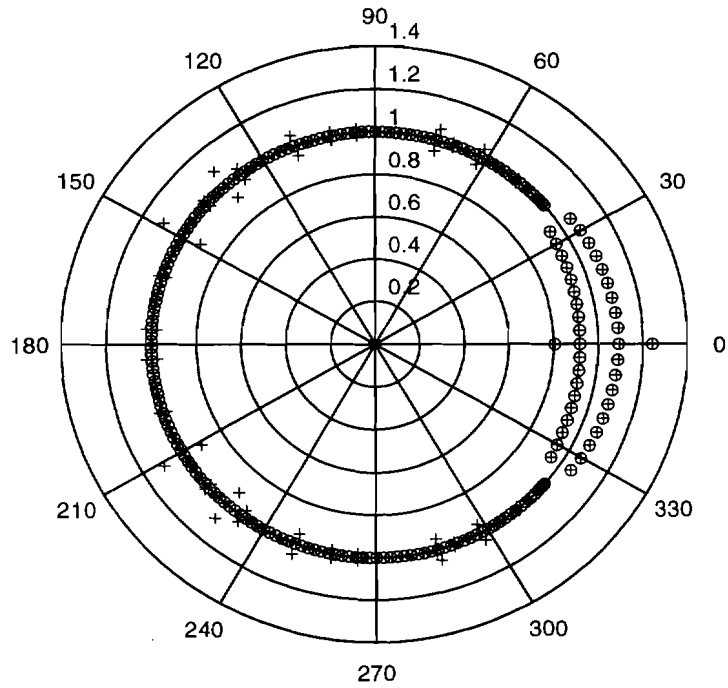

Figure 9: Pole-zero plot of the filter of figure 8 for original and PCLS coefficients. Zeros for original coefficients are represented by 'o' whereas zeros for PCLS coefficient vector are shown by '+'

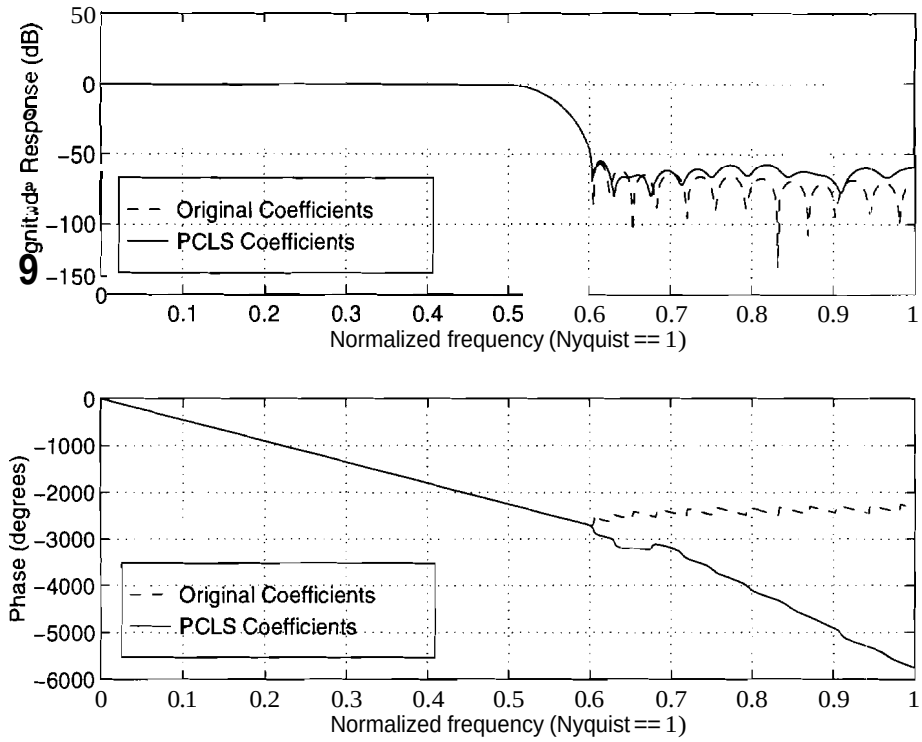

Figure 10: Frequency response of the filter of figure 4 for Booth multiplier.

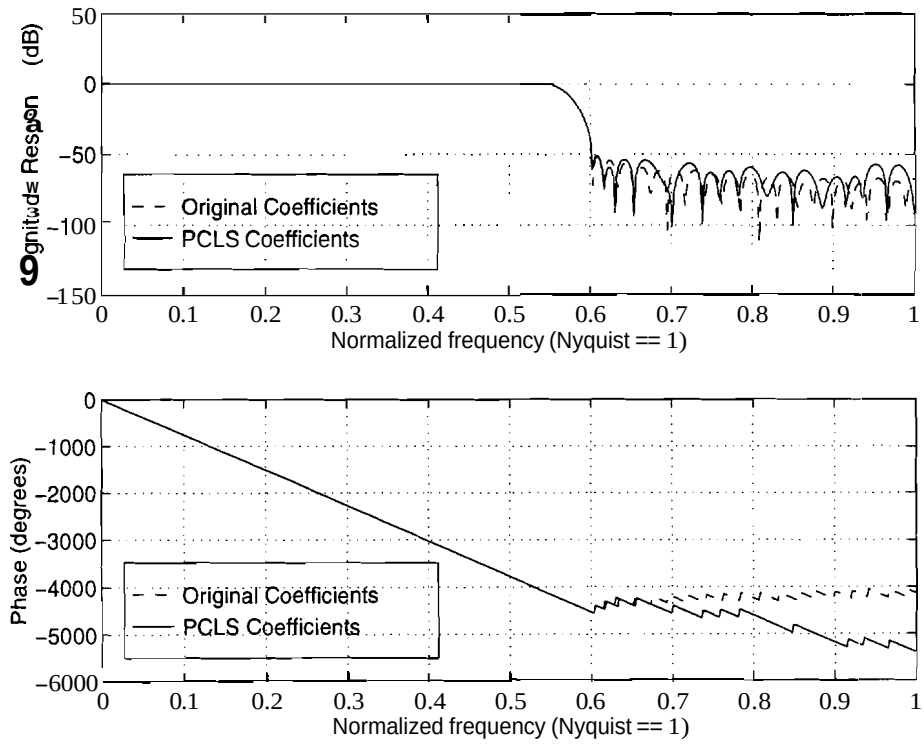

Figure 11: Frequency response of the filter of figure 6 for Booth multiplier.

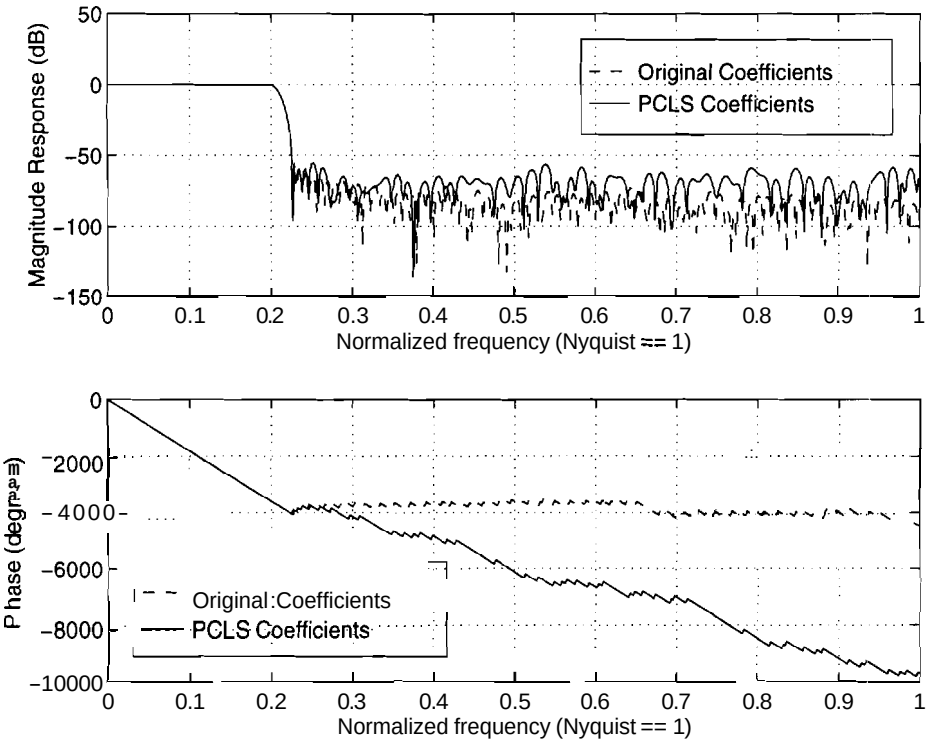

Figure 12: Frequency response of the filter of figure 8 for Booth multiplier.