Design of IIR Digital Filters with Discrete Coefficients Based on MLS Criterion

Masayoshi NAKAMOTO†, Member and Takao HINAMOTO†, Fellow

SUMMARY In this paper, we treat a design problem for IIR digital filters described by rational transfer function in discrete space. First, we form the filter design problem using the modified least-squares (MLS) criterion and express it as the quadratic form with respect to the numerator and denominator coefficients. Next, we show the relaxation method using the Lagrange multiplier method in order to search for the good solution efficiently. Additionally, we can check the filter stability when designing the denominator coefficients. Finally, we show the effectiveness of the proposed method using a numerical example.

key words: IIR digital filters, discrete coefficients, modified least-squares (MLS) criterion, lower bound estimation

1. Introduction

When digital filters are realized in hardware, we have to consider the finite-wordlength effects because the quantization of the filter coefficients causes undesirable changes on the specification. Especially, in IIR digital filter, these changes can cause the originally stable filter to become unstable.

The synthesis of low-sensitivity digital filter structures using coordinate transformation is an effective method to reduce the degradation of filter performance [1]–[5]. However, this method can be only used when that is described by the state-space model. So when the filter is described by rational transfer function, we cannot use this method, and have to design the filter coefficients in discrete space in order to avoid the filter degradation. If we search for the filter coefficients in discrete space, it takes enormous time to find the best solution. Hence, the design of digital filter with discrete coefficients is really hard.

Also, the mean square error criterion and minimax error criterion are commonly used when designing digital filters. However, it is hard to find the optimum coefficients which minimize these errors in IIR digital filters design. So the modified least-squares (MLS) criterion, which is the linear optimization problem, has been proposed in [6], and it is used for designing IIR digital filters [7]–[10]. Using MLS criterion, it is known that the mean square error and minimax error can be reduced.

In this paper, we propose the design method of IIR digital filter described by rational transfer function in discrete space. Here, we treat the approximation problem of given impulse response with discrete coefficients. We form the filter design problem based on MLS criterion and search for the good coefficients using tree model. To reduce the calculation cost, we make use of the lower bound estimation principle and show how to calculate the relaxation solution by the Lagrange multiplier method. Also, using the proposed method, the filter stability can be guaranteed by checking the filter poles while searching.

The design methods for discrete filter coefficients using the simulated annealing [11] and genetic algorithm [12] have been proposed, respectively. However, it is known that the search performance of simulated annealing and genetic algorithm depend on how to choose the initial parameters like the initial temperature or crossover rate. The proposed method does not need to choose such initial parameters and can be guaranteed the optimality in the MLS sense.

This paper is organized as follows: In the next section, we summarize MLS criterion proposed in [6]. We show the problem formulation by the tree model in Sect. 3, and propose the design algorithm based on the lower bound estimation in Sect. 4. In Sect. 5, using numerical example, we show the comparison between the simple roundoff method and proposed method, and confirm the effectiveness of the proposed method. Finally, a conclusion is presented in Sect. 6.

2. MLS Criterion [6]

In this paper, we treat the filter design problem in the time domain. That is, our goal is to approximate the given impulse response.

Let the desired transfer function be

\[ H(z) = \sum_{k=0}^{\infty} h_k z^{-k} \] (1)

where \( h_k \) is the given impulse response so that

\[ \sum_{k=0}^{\infty} |h_k| < \infty. \] (2)

We approximate \( H(z) \) using the rational function

\[ \hat{H}(z) = \frac{\hat{N}(z)}{\hat{D}(z)} \] (3)

with

\[ \hat{D}(z) = \sum_{k=0}^{m} a_k z^{-k} \]

Copyright © 2006 The Institute of Electronics, Information and Communication Engineers
\[ N(z) = \sum_{k=0}^{n} b_k z^{-k} \]

where \(a_0 = 1\).

To obtain the filter coefficients \(a_k\) \((k = 1, 2, \cdots, m)\), \(b_k\) \((k = 0, 1, \cdots, n)\), we minimize the following function

\[ e^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |H(e^{j\omega})D(e^{j\omega}) - N(e^{j\omega})|^2 d\omega \]  

(4)

which is called MLS criterion. Using Parseval’s theorem, (4) can be written as

\[ e^2 = \sum_{k=0}^{m} \sum_{l=0}^{m} a_k r_{l-k} a_{l'} - 2 \sum_{k=0}^{m} \sum_{l=0}^{n} a_k h_{l-k} b_l \]

\[ + \sum_{i=0}^{n} b_i^2 \]  

(5)

where

\[ r_k = \sum_{k=0}^{\infty} h_{k} h_{k'+k}. \]  

(6)

(5) can be expressed in this form

\[ e^2 = e^2(a, b) = a^T R a - 2a^T H b + b^T b \]  

(7)

with

\[ R = \begin{pmatrix}
  r_0 & r_1 & \cdots & r_m \\
  r_{-1} & r_0 & \cdots & r_{m-1} \\
  \vdots & \vdots & \ddots & \vdots \\
  r_{-m} & r_{1-m} & \cdots & r_0
\end{pmatrix} \]

\[ H = \begin{pmatrix}
  h_0 & h_1 & \cdots & h_n \\
  h_{-1} & h_0 & \cdots & h_{n-1} \\
  \vdots & \vdots & \ddots & \vdots \\
  h_{-m} & h_{1-m} & \cdots & h_{n-m}
\end{pmatrix} \]

\[ a = (a_0, a_1, \cdots, a_m)^T \]

\[ b = (b_0, b_1, \cdots, b_n)^T. \]

Since \(e^2\) is quadratic with respect to the filter coefficients \(a\) and \(b\), the optimal solution which minimizes (7) can be obtained by differentiating (7) with respect to \(a\) and \(b\) and equating the resulting expression to zero. Thus we have

\[ a = \frac{1}{\mu} (R - HH^T)^{-1} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \]  

(8)

where \(\mu\) is the \((1, 1)\)-th elements of \((R - HH^T)^{-1}\). Also, the optimal \(b\) is obtained as

\[ b = H^T a. \]  

(9)

3. Problem Formulation by the Tree Model

Consider the IIR digital filter with finite-wordlength coefficients whose integral part is expressed as \(K\) bit and the decimal part is expressed as \(L\) bit. That is

\[ a_k = \pm \sum_{i=L}^{K-1} s_k(i) 2^i, \quad k = 1, 2, \cdots \]  

(10)

\[ b_k = \pm \sum_{i=L}^{K-1} t_k(i) 2^i, \quad k = 0, 2, \cdots \]  

(11)

where \(s_k(i) = 0, 1, t_k(i) = 0, 1\). Assume that \(S_a\) is the set of \(a_k\), and \(S_b^l\) is the set of \(l\) dimensional vectors whose elements are \(a_k\). Also, assume that \(S_b\) is the set of \(b_k\), and \(S_b^l\) is the set of \(l\) dimensional vectors whose elements are \(b_k\).

Hence the problem is as follows

\[ \min e^2(a, b), \quad a \in S_a^m, \quad b \in S_b^{n+1}. \]  

(12)

\(a_k\) and \(b_k\) can be taken \(2^{K+L+1} - 1\) values taking into account the sign bit. So the combinations of coefficients are

\[ \psi = (2^{K+L+1} - 1)^{m+n+1}. \]  

(13)

If \(m, n\) or \(K, L\) are large numbers, it is hard to search for the optimal solution. For example, consider \(K = 2, L = 4\), if \(m = n = 3\),

\[ \psi = 127^2 = 532875860165503 \approx 5.3 \times 10^{14} \]  

(14)

and if \(m = n = 4\),

\[ \psi = 127^3 = 8594754748609397887 \approx 8.6 \times 10^{18}. \]  

(15)

From (14), (15), it is obvious that to search every combination is almost impossible.

In this paper, we consider the structure of tree as shown in Fig.1 where assuming \(a_i\) and \(b_i\) can be taken three discrete values \(d_i, d_2, d_3\). In the tree model, we call that \(a_0 = 1\) is root, the discrete values \(d_1, d_2\) etc. are branch values, the coefficients \((a_1, b_0\) etc.) are branch coefficients, and the branch which is located in the

[Image: Fig.1 The structure of tree.]
bottom of tree is leaf. So the leaf corresponds to the solution in this problem.

Here we optimize the denominator coefficients before the numerator coefficients, because we can estimate the filter stability before optimizing the numerator coefficients.

4. Design Method with Discrete Coefficients

4.1 Lower Bound Estimation Principle

To reduce the search space, we have to cut the subtrees which do not include the good solution by using the lower bound estimation principle as follows.

Let the subtrees consisting of the elements of \( a \) and \( b \) be \( t_a \) and \( t_b \), respectively, where

\[
t_a \in S^m_a
\]

(16)

\[
t_b \in S^{n+1}_b
\]

(17)

Assume that \( t_b \) is located under \( t_a \) so that we check the filter stability. If the filter is unstable we do not search \( t_b \).

For example,

\[
t_a = \{a_1 = d_2, a_2 = d_1, a_3 = d_2\}
\]

(18)

\[
t_b = \{b_0 = d_3, b_1 = d_1, b_2 = d_2, b_3 = d_1\}
\]

(19)

is one of the examples where \( d_1, d_2 \) and \( d_3 \) are the discrete values.

We divide \( t_a \) into two subtrees \( t'_a \) and \( t''_a \) as

\[
t_a = \{t'_a, t''_a\}
\]

(20)

where

\[
t'_a \in S^l_a
\]

(21)

\[
t''_a \in S^{m-l}_a
\]

(22)

and assume that \( t''_a \) is located under \( t'_a \).

In the above example,

\[
t'_a = \{a_1 = d_2, a_2 = d_1\}
\]

(23)

\[
t''_a = \{a_3 = d_2\}
\]

(24)

is one of the possible combinations.

Next, we optimize the elements of \( t''_a \) with continuous values and replace \( t_a, t''_a \) with \( \overline{t}_a, \overline{t}_a'' \) respectively, that is

\[
\overline{t}_a = \{t'_a, \overline{t}_a''\}.
\]

(25)

Also, we optimize the elements of \( t_b \) with continuous values and replace \( t_b \) with \( \overline{t}_b \). We call it the relaxation solution of \( t_a \).

Then it is obvious

\[
\varepsilon^2(t_a, t_b) \geq \varepsilon^2(\overline{t}_a, \overline{t}_b)
\]

(26)

holds. If

\[
\varepsilon^2(\overline{t}_a, \overline{t}_b) \geq \varepsilon_p^2
\]

(27)

where \( \varepsilon_p^2 \) is the evaluated value of the best solution already found (the temporary solution), then

\[
\varepsilon^2(t_a, t_b) \geq \varepsilon^2(\overline{t}_a, \overline{t}_b) \geq \varepsilon_p^2
\]

(28)

holds and this indicates there is no better solution (leaf) under the subtree \( t''_a \). So we do not have to search \( t''_a \) and \( t_b \).

Similarly, divide \( t_b \) into \( t'_b \) and \( \overline{t}_b'' \) as

\[
t_b = \{t'_b, \overline{t}_b''\}
\]

(29)

where the elements of \( t'_b \) and \( \overline{t}_b'' \) are both the discrete values. Note that \( a \) is a constant vector in this case.

Next, consider the relaxation solution of \( t_b \) as well as \( \overline{t}_a \). Let

\[
\overline{t}_b = \{t'_b, \overline{t}_b''\}
\]

(30)

be the relaxation solution of \( t_b \) where the elements of \( \overline{t}_b'' \) are continuous values which minimize \( \varepsilon^2 \).

Then, if

\[
\varepsilon^2(t_a, \overline{t}_b) \geq \varepsilon_p^2
\]

(31)

we do not have to search \( t''_b \).

This is the lower bound estimation using the relaxation solution. Based on this principle we can find the best solution which minimizes \( \varepsilon^2 \) faster than when we search the all combination (13). In the next section, we show how to calculate the relaxation solution.

4.2 Constraints of Coefficients

Consider some of the elements of \( a \) having the values \( p_1, p_2, \ldots, p_k \) as

\[
a_i = p_i
\]

\[
a_i = p_i
\]

\[
\vdots
\]

\[
a_i = p_k
\]

Then using \( k \) vectors \( c_1, c_2, \ldots, c_k \), it can be written by

\[
a^T c_1 = p_1
\]

(32)

\[
a^T c_2 = p_2
\]

\[
\vdots
\]

\[
a^T c_k = p_k
\]

(33)

where the \( i \)th elements of \( c_1 \) is 1, other elements are 0.

We make the matrix as

\[
C = (c_1, c_2, \ldots, c_k)
\]

(34)

From (32) and (33) it follows that

\[
\begin{align*}
a^T C &= p^T
\end{align*}
\]

(35)

For example, \( C \) and \( p \) corresponding to \( a_0 = 1, a_3 = 0.5, a_4 = 1.25 \) are as follows
\[
C = \begin{bmatrix}
1 & 0 & 0 \\
0 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]
\[
p = (1, 0.5, 1.25)^T.
\]

Note that \(e_k\) has an orthogonality
\[
e_k^T e_l = \begin{cases} 1 & (k = l) \\ 0 & (k \neq l) \end{cases}
\]
so
\[
C^T C = E
\]
holds where \(E\) stands for the identity matrix of appropriate dimension.

4.3 Relaxation Solution

(a) Relaxation solution of \(a\)

The relaxation solution of \(a\) can be obtained to calculate \(a\) and \(b\) to minimize \(\varepsilon^2(a, b)\) under constraints
\[
a^T C = p^T.
\]

By using the Lagrange multiplier method, define
\[
J(a, b, \lambda) = \frac{1}{2}(a^T Ra - 2a^T Hb + b^T b) - (a^T C - p^T)\lambda.
\]
Differentiating \(J(a, b, \lambda)\) with respect to \(a\), \(b\), and \(\lambda\) yields
\[
\begin{align*}
\frac{\partial J(a, b, \lambda)}{\partial a} &= Ra - Hb - CA = 0 \\
\frac{\partial J(a, b, \lambda)}{\partial b} &= -H^T a + b = 0 \\
\frac{\partial J(a, b, \lambda)}{\partial \lambda} &= C^T a - p = 0
\end{align*}
\]
then we have
\[
\lambda = (C^T (R - HH^T)^{-1} C)^{-1} p.
\]
Substituting (43) into (40) yields
\[
\begin{align*}
a &= (R - HH^T)^{-1} C (C^T (R - HH^T)^{-1} C)^{-1} p \\
b &= H^T a
\end{align*}
\]
In the coefficients obtained by (45), some coefficients have discrete values and others have continuous values. All elements of \(b\) are continuous.

(b) Relaxation solution of \(b\)

In the case when \(b\) is optimized, the denominator coefficients \(a\) have been fixed. Hence \(a\) can be regarded as a constant. So the relaxation solution of \(b\) can be obtained by calculating \(b\) minimizing \(\varepsilon^2(a, b)\) under constraints
\[
b^T C = p^T.
\]
By using the Lagrange multiplier method, define
\[
J(a, b, \lambda) = \frac{1}{2}(a^T Ra - 2a^T Hb + b^T b) - (b^T C - p^T)\lambda.
\]
Differentiating \(J(a, b, \lambda)\) with respect to \(b\) and \(\lambda\) yields
\[
\begin{align*}
\frac{\partial J(a, b, \lambda)}{\partial b} &= -H^T a + b - CA = 0 \\
\frac{\partial J(a, b, \lambda)}{\partial \lambda} &= C^T b - p = 0
\end{align*}
\]
then we have
\[
\lambda = p - C^T H^T a
\]
since \(C^T C\) is a unit matrix. Substituting (49) into (47) yields
\[
b = H^T a + C(p - C^T H^T a).
\]
In the coefficients obtained by (50), some coefficients are discrete and others are continuous. Note that all elements of \(a\) are discrete values in (50).

4.4 Design Algorithm

First, we decide the filter order \((m, n)\) and coefficients wordlength \(K\) and \(L\). Next, we design the filter coefficients using the algorithm summarized as follows.

Before showing the algorithm, decide some parameters used in the algorithm as follows. (Fig. 2 shows an example of \(d\) and \(B(d)\).)

\(d\): The depth parameter.
\(B(d)\): The branch parameter in the depth \(d\).
\(B_0\): The width of branch.

**STEP 1:** \(d = 1, B(d) = 1 (d = 1, 2, \cdots)\). Assume the

\[d=1\]

\[d=2\]

\[d=3\]

\[d=4\]

\[B(d)=\]

\[\text{Fig. 2} \quad \text{Example of } d \text{ and } B(d) \text{ in the tree model.}\]
value of (7) of the first temporary solution is \( \infty \). Go to STEP 2.

**STEP 2:** \( B(2) \leq B_w \)? If yes, go to STEP 3. If not, stop.

**STEP 3:** \( B(d) \leq B_w \)? If yes, go to STEP 4. If not, \( d = d - 1, B(d) = B(d) + 1 \) and go to STEP 4.

**STEP 4:** If \( 1 \leq d \leq m \), go to STEP 5. If \( d = m + 1 \), go to STEP 7. If \( m + 2 \leq d \leq m + n + 1 \), go to STEP 8. If \( d = m + n + 2 \) go to STEP 9.

**STEP 5:** If \( d = 1 \), calculate the continuous coefficients which minimize (7) and obtain the result of (7) by using (8). If not, obtain the relaxation solution using (45),(45) and calculate (7). Go to STEP 6.

**STEP 6:** If the value of (7) is better than the temporary solution, select the branch coefficient and get the branch values, \( d = d + 1, B(d) = 1 \) and go to STEP 2. If not, \( B(d) = B(d) + 1 \) and go to STEP 2.

**STEP 7:** Check the stability of the filter. If stable, go to STEP 8. If not, \( B(d) = B(d) + 1 \) and go to STEP 2.

**STEP 8:** Obtain the relaxation solution using (50) and calculate (7). Go to STEP 6.

**STEP 9:** Calculate (7). If the value of (7) is better than the temporary solution, update the temporary solution. \( B(d) = B(d) + 1 \) and go to STEP 2.

4.5 Selection Rule of Branch Coefficient

The selection rule of coefficients is summarized as follows. First, consider the coefficients in the continuous space

\[
\alpha_1, \alpha_2, \cdots, \alpha_r
\]

Calculate the distance of its nearest discrete value and put into

\[
\delta_1, \delta_2, \cdots, \delta_r
\]

respectively. Then select \( \alpha_k \), which is corresponding to \( \delta_k \), as the new branch coefficient where

\[
\delta_k = \min(\delta_1, \delta_2, \cdots, \delta_r).
\]

We show the relationship between \( B(d) \) (the order of search) and the corresponding discrete value (branch value) in Table 1 where

\[
\Delta = 2^{-L}, \quad |r \pm i\Delta| \leq 2^K - 2^{-L}
\]

and \( \alpha \) is the nearest discrete value of branch coefficient \( \alpha_k \). Also, \( \Delta \) is the difference of discrete coefficients. That is, we search the nearest discrete value first. The selection rule of \( b_k \) is the same in the case of \( a_k \).

<table>
<thead>
<tr>
<th>( \alpha )</th>
<th>( \alpha + \Delta )</th>
<th>( \alpha - \Delta )</th>
<th>( \alpha + 2\Delta )</th>
<th>( \alpha - 2\Delta )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_k )</td>
<td>( b_k )</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5. Design Example

In this example, the problem is to approximate the impulse response given by

\[
h_k = \begin{cases} 
10 \exp(-0.1032(k - 4)^2) & (k \geq 0) \\
0 & (k < 0).
\end{cases}
\]

Assuming that \( K = 2 \) and \( L = 4 \). In the case of continuous coefficients, the results of \( e^2 \), \( E_2 \) and \( E_\infty \) are summarized in Table 2 where

\[
E_2 = \sum_{i=0}^{100} \left| h_i - \hat{h}_i \right|^2 \times 100
\]

\[
E_\infty = \max_{0 \leq i \leq 100} |h_i - \hat{h}_i| \times 100
\]

and \( \hat{h}_i \) is the approximating impulse response.

The results of the applications of the simple roundoff method and proposed method to this example are summarized in Table 3.

<table>
<thead>
<tr>
<th>( a_k )</th>
<th>( b_k )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_k )</td>
<td>( b_k )</td>
</tr>
</tbody>
</table>

\( ^1 \)We show how to select branch coefficient and branch value in Sect. 4.5
In this paper, we have proposed the design method for IIR digital filter with discrete coefficients. We have formed the filter design problem based on MLS criterion and search for the good solution using the lower bound estimation of relaxation problem. Also, the stability of the filter designed by the method can be guaranteed. Finally, we have confirmed the effectiveness of the proposed method using a numerical example.

**References**


