目白大学　経営学研究
4号　　　2006年3月　31－45

2次錐計画法によるディジタルフィルタの設計法

Integer Programming Approach to Design of Digital Filter Using Second Order Cone Problem Relaxation

平林隆一　伊藤利佳　藤江哲也
(Ryuichi HIRABAYASHI　Rika ITO　Tetsuya FUJIE)

【要　約】

情報化社会といわれる今日，おびただしい量の情報を、コンピュータを使ってどう経営に活用するかは企業経営にとって重要な課題である．中でも通信技術の発達は，迅速な意思決定や経営判断を迫られる現代においては欠かせない重要な技術といえる．そのため，通信技術のひとつであるディジタルフィルタの研究においてもここ30年でめざましい成果があげられている．
ディジタルフィルタは，ディジタル計測，ディジタル通信等において中心的な役割をなす信号処理回路であり、その設計における精度の高さのレベルや設計の高速化といった問題が非常に重要な研究課題のひとつとなっている．
そこで、本稿では、最適化手法のひとつである線形計画問題（LP）を緩和した問題に三角不等式を付加した方法および2次錐計画問題による緩和をもちいたディジタルフィルタの新たな設計方法を提案し，比較実験を行ったのでその結果も報告する．
キーワード：デジタルフィルタ，半正定値計画法，線形計画法，線形緩和，符号付き2連数，近似

Abstract:

In this paper, we propose a new approach for designing digital filters by one of the optimization methods, "second order cone problem relaxation (SOCP)" based on the semidefinite programming (SDP) relaxation methods.
And we show the solution with our proposed design method is better than that with SDP relaxation method through several numerical experiments.
keyword:
digital filter, optimization, semi-definite programming, SDP, Linear programming, LP, Linear programming relaxation, Signed power of two, approximation

ひらばやしりゅういち：経営学部経営学科教授
いとうりか：東邦大学理学部
ふじえてつや：兵庫県立大学経営学部

1 Introduction

Recently, many studies on a design method for linear phase FIR filters with discrete coefficients have been published [7]— [9], in which, a numerical representation by a sum of signed power of two (SP2) has been used in several methods.

It is a reason that a small number of non-zero digits is often required for a representation of the coefficients in a VLSI implementation of the filters. However, it is difficult to design filters with SP2 coefficients since it results in an integer programming problem~(IP) well-known as one of the NP-hard problems which can not be solved in polynomial time [11],

To overcome this difficulty W. -S. Lu [9] proposed an semidefinite programming (SDP) relaxation method for the design problem and showed that the approach has a good performance through numerical experimentation. SDP relaxation is a recently developed technique for approximating nonconvex programming problems such as integer programming problems in the field of mathematical programming [3]. It is well known that SDP problem is a convex programming problem and can be solved in polynomial time under some constraint qualification [6]. Hence, in this paper we introduce some new LP relaxation algorithm based on SDP relaxation. This proposed algorithm allows the design problem to be solved in polynomial time. In this method we formulate these design problems as LP relaxation problems with several simple constraints by modifying SDP relaxation. We strengthen the relaxation by adding more triangle inequalities on that LP relaxation problem since simple LP relaxation is usually too weak. Even adding several constraints, the optimization problem can be also solved in polynomial time. We demonstrate the conclusion that the solutions of LP relaxation with triangle inequalities is better than SDP relaxation through numerical experiments. Numerical experiments show that these triangle inequality constraints are valuable enough to improve the solutions of these design problems.

This paper consists of five sections. The following section is devoted to formulate the FIR filter design problem concerned as {-1,1}-optimization problem and requisite preliminaries including definitions and notations. In Section 3, a traditional design problem by using SDP relaxation algorithm and our new LP relaxation algorithm with triangle inequalities are introduced. Numerical experiments are made, in Section 4, in comparison with the results given from the SDP relaxation, to illustrate that our algorithm improve the solutions of objective values. Section 5 is a conclusion.

2 Design of digital filters by using {-1,1}-optimization method

In this section, we introduce the design method of digital FIR filters with SP2 coefficients using SDP problem by following Lu [9]. This design method is constructed by two steps: (1) solve the design problem of digital filters with desired frequency characteristics by using continuous variables, (2) formulate the design problem of digital problems with SP2 coefficients as a {-1,1}-optimization problem. To consider the structure of the {-1,1}-optimization problem obtained, we will convert the {-1,1}-optimization problem to a minimum cut problem with negative coefficients which belongs to the class NP-hard.

2. 1 Design problem of FIR digital filters with continuous coefficients

In this paper, we deal with a design problem of FIR digital filters with SP2 coefficients that minimize the weighted least square errors (WLS) , i.e., minimize the following function:

$$e = \int_0^\pi W(\omega)|H(e^{j\omega}) - H_d(\omega)|^2 d\omega, \tag{1}$$

where $W(\omega) \geq 0$ is a weight function and $H_d(\omega)$ is the desired frequency response function. In the first, we consider the continuous coefficient case. Then the design function of the FIR filter is:

$$H_c(z) = \sum_{k=0}^{N-1} h_k z^{-k}. \tag{2}$$

Now, we assume M is the total number of SP2 terms that can be used in $H(e^{j\omega})$ and m_k is the number of SP2 terms used in the k-th term of the frequency response $H(e^{j\omega})$. Then we denote

$$H(z) = \sum_{k=0}^{N-1} d_k z^{-k}, \tag{3}$$

where

$$\sum_{k=0}^{N-1} m_k = M. \tag{4}$$

The allocation of SP2 terms is determined, for example, by [8].

We assume that the absolute value of each SP2 term $\{d_k\}$ is in the interval $[2^0, 2^{-U}]$ where U is a natural number. Then, by (3),

$$d_k = \sum_{i=1}^{m_k} b_i^{(k)} 2^{-q_i^{(k)}}. \tag{5}$$

Here, we have $b_i^{(k)} \in \{-1, 1\}$ and $1 \leq q_i^{(k)} \leq U,\ (1 \leq i \leq m_k,\ 0 \leq k \leq N-1)$.

For given $\{m_k,\, k = 0, \ldots, N-1\}$ and U, when an optimal continuous solution $H_c(z) = \sum_{k=0}^{N-1} h_k z^{-k}$ is obtained, it is easy to find the maximum SP2 number $\bar{d}_k$ and the minimum SP2 number $\underline{d}_k$ that satisfy $\bar{d}_k \leq h_k \leq \underline{d}_k$ whose $\underline{d}_k$ and $\bar{d}_k$ satisfy (5) for the given m_k.

Let $d_{mk} = (\underline{d}_k + \bar{d}_k)/2$ be the middle point of the interval $[\underline{d}_k, \bar{d}_k]$ and $\delta_k = (\underline{d}_k - \bar{d}_k)/2$ be the half length of the interval. Then, $\underline{d}_k$ and $\bar{d}_k$ are expressed as $d_{mk} + x_k\delta_k$ $(x_k = -1)$ and $d_{mk} + x_k\delta_k$ $(x_k = 1)$, respectively. Hence, the transfer function $H(z)$ with discrete coefficient function becomes

$$\begin{aligned} H(e^{j\omega}) &= \sum_{k=0}^{N-1} d_k^{(s)} e^{-jk\omega} && (6) \\ &= H_m(e^{j\omega}) + \boldsymbol{x}^T[\boldsymbol{c}_\delta(\omega) - j\boldsymbol{s}_\delta(\omega)] && (7) \end{aligned}$$

by using

$$d_k = d_{mk} + x_k \delta_k, \tag{8}$$

and

$$\begin{aligned} H_m(e^{j\omega}) &= \boldsymbol{d}_m^T[\boldsymbol{c}(\omega) - j\boldsymbol{s}(\omega)], \\ \boldsymbol{d}_m &= (d_{m0}, d_{m1}, \ldots, d_{m,N-1})^T, \\ \boldsymbol{c}(\omega) &= (1, \cos\omega, \ldots, \cos(N-1)\omega)^T, \\ \boldsymbol{s}(\omega) &= (0, \sin\omega, \ldots, \sin(N-1)\omega)^T, \\ \boldsymbol{c}_\delta(\omega) &= (\delta_0, \ldots, \delta_{N-1}\cos(N-1)\omega)^T, \\ \boldsymbol{s}_\delta(\omega) &= (0, \ldots, \delta_{N-1}\sin(N-1)\omega)^T, \\ \boldsymbol{x} &= (x_0, x_1, \ldots, x_{N-1})^T, \quad x_i \in \{-1, 1\}. \end{aligned} \tag{9}$$

By (7), we can easily verify that the objective function (1) becomes

$$e = \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + 2\boldsymbol{x}^T \boldsymbol{q} + \text{const}, \tag{10}$$

where

$$\begin{aligned} \boldsymbol{Q} &= \int_0^\pi W(\omega)[\boldsymbol{c}_\delta(\omega)\boldsymbol{c}_\delta^T(\omega) + \boldsymbol{s}_\delta(\omega)\boldsymbol{s}_\delta^T(\omega)]d\omega, \\ \boldsymbol{q} &= \int_0^\pi W(\omega)[E_r(\omega)\boldsymbol{c}_\delta(\omega) + E_i(\omega)\boldsymbol{s}_\delta(\omega)]d\omega, \\ E_r(\omega) &= \boldsymbol{d}_m^T \boldsymbol{c}(\omega) - H_{dr}(\omega), \\ E_i(\omega) &= \boldsymbol{d}_m^T \boldsymbol{s}(\omega) - H_{di}(\omega), \\ H_d(\omega) &= H_{dr}(\omega) - jH_{di}(\omega). \end{aligned} \tag{11}$$

Now, design problem of $H(z)$ with SP2 coefficients for minimizing weighted least square error becomes a $\{-1,1\}$-quadratic integer programming problems [9]:

$$\begin{array}{ll} \min & \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + 2\boldsymbol{q}^T \boldsymbol{x} \\ \text{sub. to} & \boldsymbol{x} \in \{-1, 1\}^N. \end{array} \tag{12}$$

2. 2 Derivation to the minimum cut problem

To know the structures of optimization problems is always important. In the following, we will point out that the problem (12) can be easily converted to a minimum cut problem with negative coefficients. Minimum cut problems with negative coefficients belong to a class of NP-hard. Hence, this shows (12) is a hard problem to be solved.

Let $G = (V, E)$ be a perfect graph with a vertex set $V = \{0, 1, \ldots, N\}$ and an edge set $E = \{ij \mid 0 \le i < j \le N\}$, and, the weight $w_e (e \in E)$ of the each edge be

$$\begin{cases} w_{jN} = -q_j \ (j = 0, \ldots, N-1), \\ w_{ij} = -Q_{ij} \ (0 \le i < j \le N-1). \end{cases} \tag{13}$$

Then the minimum cut problem of the graph G is:

$$\begin{array}{ll} \min & 4\sum_{e \in E} w_e y_e + \boldsymbol{e}^T \boldsymbol{Q} \boldsymbol{e} + 2\boldsymbol{q}^T \boldsymbol{e} \\ \text{sub. to} & \boldsymbol{y} : \text{a 0-1 cut vector of } G \end{array} \tag{14}$$

or equivalently

$$
\begin{array}{ll}
\min & \sum_{ij \in E} w_{ij}(x_i - x_j)^2 + \boldsymbol{e}^T \boldsymbol{Q} \boldsymbol{e} + 2\boldsymbol{q}^T \boldsymbol{e} \\
\text{sub. to} & \boldsymbol{x} \in \{-1, 1\}^{N+1},
\end{array}
\tag{15}
$$

where $\boldsymbol{e} = (1, \ldots, 1)^T \in \mathbb{R}^N$. It is easy to see that problem (12) and (15) are equivalent each other. Here, we denote, without loss of generality, $x_N = 1$ in (15). Since, as we pointed out that minimum cut problem with negative coefficients belongs to the class of NP-hard, there will no hope to develop efficient algorithms to solve (12) directly.

3 SDP relaxation and LP relaxation

Following Lu [9], we reformulate (12) as

$$
\begin{array}{ll}
\min & \boldsymbol{Q} \bullet \boldsymbol{X} + 2\boldsymbol{q}^T \boldsymbol{x} \\
\text{sub. to} & \boldsymbol{X} - \boldsymbol{x}\boldsymbol{x}^T = \boldsymbol{O}, \\
& \boldsymbol{x} \in \{-1, 1\}^N,
\end{array}
\tag{16}
$$

where $Q \bullet X = \sum_{i,j} Q_{ij} X_{ij}$. Then, since $X_{ii} = x_i^2 = 1$ $(i = 0, \ldots, N-1)$, we obtain an relaxation problem of (12) by

$$
\begin{array}{ll}
\min & \boldsymbol{Q} \bullet \boldsymbol{X} + 2\boldsymbol{q}^T \boldsymbol{x} \\
\text{sub. to} & X_{ii} = 1 \ (i = 0, \ldots, N-1), \\
& \boldsymbol{X} - \boldsymbol{x}\boldsymbol{x}^T \succeq \boldsymbol{O}.
\end{array}
\tag{17}
$$

Here, $A \succeq \boldsymbol{O}$ denotes that A is positive semidefinite. Between a cut vector $\boldsymbol{y} \in \{0, 1\}^E$ of the graph G and $(\boldsymbol{x}, \boldsymbol{X})$ that satisfies (16), the following equations hold:

$$
\left\{
\begin{array}{l}
x_j = -2y_{jN} + 1 \ (j = 0, \ldots, N-1), \\
X_{ij} = -2y_{ij} + 1 \ (0 \le i < j \le N-1).
\end{array}
\right.
\tag{18}
$$

We can use softwares to solve SDP in polynomial times, for example, SeDuMi [12].

It is easily verified that the next triangle inequalities hold [10]:

$$
\left.
\begin{array}{r}
x_i + x_j + X_{ij} \ge -1, \\
x_i - x_j - X_{ij} \ge -1, \\
-x_i - x_j + X_{ij} \ge -1, \\
-x_i + x_j - X_{ij} \ge -1.
\end{array}
\right\}
\tag{19}
$$

Here, we denote $0 \le i < j \le N-1$. And, for $0 \le i < j < k \le N-1$,

$$
\left.
\begin{array}{r}
X_{ij} + X_{ik} + X_{jk} \ge -1, \\
X_{ij} - X_{ik} - X_{jk} \ge -1, \\
-X_{ij} - X_{ik} + X_{jk} \ge -1, \\
-X_{ij} + X_{ik} - X_{jk} \ge -1
\end{array}
\right\}
\tag{20}
$$

hold. (19) is the triangle inequalities for the vertex set $\{i, j, N\}$, and (20) is the triangle inequalities for the vertex set $\{i, j, k\}$. Therefore, the next optimization problem

$$\begin{array}{ll} \min & \boldsymbol{Q} \bullet \boldsymbol{X} + 2\boldsymbol{q}^T \boldsymbol{x} \\ \text{sub. to} & X_{ii} = 1 \ (i = 0, \ldots, N-1), \\ & (19), (20), \\ & \boldsymbol{X} - \boldsymbol{x}\boldsymbol{x}^T \succeq \boldsymbol{O} \end{array} \tag{21}$$

strengthen the problem (17).

Since it is easier to solve LP problems than SDP problem, we will relax the SDP relaxation problem to LP relaxation problem. The following minimization problem

$$\begin{array}{ll} \min & 2\sum_{i<j} Q_{ij} X_{ij} + 2\boldsymbol{q}^T \boldsymbol{x} + \sum_{i=0}^{N-1} Q_{ii} \\ \text{sub. to} & (19), \\ & -1 \leq x_i \leq 1 \ (i = 0, \ldots, N-1) \end{array} \tag{22}$$

is a LP relaxation problem with a bounded optimal solution. Adding (20) as constraints,

$$\begin{array}{ll} \min & 2\sum_{i<j} Q_{ij} X_{ij} + 2\boldsymbol{q}^T \boldsymbol{x} + \sum_{i=1}^{N-1} Q_{ii} \\ \text{sub. to} & (19), (20), \\ & -1 \leq x_i \leq 1 \ (i = 0, \ldots, N-1), \end{array} \tag{23}$$

we have a strengthened problem for (22).

4 Second Order Cone Problem Relaxation

It is now popular to use semidefinite programming (SDP) ([3], [4]) for relaxation problems whenever possible. Although SDP relaxation gives a good bound, the computational cost of solving SDP is so expensive, in spite of efforts to develop better algorithms to solve SDP, that it is still difficult to use SDP problems for solving large problems [10]. Second-order cone programming (SOCP) is an optimization problem having linear constraints and second-order cone constraints. SOCP is a special case of symmetric cone programming [2], which also includes SDP and LP as special cases. Since primal-dual interior-point algorithms were developed for both SOCP, several programs have been implemented to solve SOCP [1]. Numerical experiments show that the computational cost of solving SOCP is much less than that of SDP, and similar to LP.Kim and Kojima [5] first pointed out that SOCP can be used to relax integer programming problem. We denote by $\mathcal{S}(n)$ the set of $n \times n$ real symmetric matrices. Also, $\mathcal{S}(n)^+$ denotes the set of $n \times n$ positive semidefinite matrices. For $X, Y \in \mathcal{S}(n)$,

$$X \bullet Y := \sum_{i,j} X_{ij} Y_{ij}$$

and $X \succeq Y$ if and only if $X - Y \in \mathcal{S}(n)^+$. The second-order cone $\mathcal{K}(r)$ is defined by

$$\mathcal{K}(r) := \left\{ \boldsymbol{x} \in \mathbb{R}^r \;\middle|\; s_1 \geq \sqrt{\sum_{j=2}^{r} x_j^2} \right\}.$$

The vector $\boldsymbol{e}_j \in \mathbb{R}^n$ is the zero except for the j-th component, which is 1.

The basic idea of a second-order cone relaxation problem is as follows.
Since the next relation

$$\boldsymbol{X} - \boldsymbol{x}\boldsymbol{x}^T \succeq \boldsymbol{O} \;\Leftrightarrow\; \boldsymbol{C} \bullet (\boldsymbol{X} - \boldsymbol{x}\boldsymbol{x}^T) \geq 0 \;\; ({}^\forall \boldsymbol{C} \succeq \boldsymbol{O}) \tag{24}$$

holds then

$$\boldsymbol{X} - \boldsymbol{x}\boldsymbol{x}^T \succeq \boldsymbol{O}.$$

Then we can relax the next equation,

$$\boldsymbol{C} \bullet (\boldsymbol{x}\boldsymbol{x}^T - \boldsymbol{X}) \leq 0 \;\; (\boldsymbol{C} \in \mathcal{C}) \tag{25}$$

for some $\mathcal{C} \subseteq \mathcal{S}(n)$.

4.1 Kim-Kojima Method

Set $\boldsymbol{e}_i\boldsymbol{e}_i^T$ $(i = 0, \ldots, K)$, $\boldsymbol{Q}$ as $\mathcal{C}$.
We can rewrite the constraints of (17) using $\mathcal{C}$.

$$\begin{aligned} \boldsymbol{C} \bullet (\boldsymbol{x}\boldsymbol{x}^T - \boldsymbol{X}) &= \boldsymbol{e}_i\boldsymbol{e}_i^T \bullet (\boldsymbol{x}\boldsymbol{x}^T - \boldsymbol{X}) = \boldsymbol{e}_i\boldsymbol{e}_i^T \bullet \boldsymbol{x}\boldsymbol{x}^T - \boldsymbol{e}_i\boldsymbol{e}_i^T \bullet \boldsymbol{X} \\ &= x_i^2 - X_{ii} \leq 0, \end{aligned} \tag{26}$$

$$\begin{aligned} \boldsymbol{C} \bullet (\boldsymbol{x}\boldsymbol{x}^T - \boldsymbol{X}) &= \boldsymbol{Q} \bullet (\boldsymbol{x}\boldsymbol{x}^T - \boldsymbol{X}) = \boldsymbol{Q} \bullet \boldsymbol{x}\boldsymbol{x}^T - \boldsymbol{Q} \bullet \boldsymbol{X} \\ &= \boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{Q} \bullet \boldsymbol{X} \leq 0. \end{aligned} \tag{27}$$

Therefore we can obtain the relaxation problem.

$$\begin{array}{ll} \text{min} & \boldsymbol{Q} \bullet \boldsymbol{X} + 2\boldsymbol{q}^T\boldsymbol{x} \\ \text{sub. to} & X_{ii} = 1 \;\; (i = 0, \ldots, K), \\ & x_i^2 - X_{ii} \leq 0 \;\; (i = 0, \ldots, K), \\ & \boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{Q} \bullet \boldsymbol{X} \leq 0. \end{array} \tag{28}$$

Here let $z = \boldsymbol{Q} \bullet \boldsymbol{X}$ and $X_{ii} = 1$, then we can obtain the next problem

$$\begin{array}{ll} \text{Minimize} & z + 2\boldsymbol{q}^T\boldsymbol{x} \\ \text{subject to} & x_i^2 \leq 1 \;\; (i = 0, \ldots, K), \\ & \boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} - z \leq 0. \end{array} \tag{29}$$

In the second constraints,the the $\boldsymbol{Q} \bullet \boldsymbol{X} = z$ holds, then SOCP for $\mathcal{C}$, is equivalent to the next equation.

$$\begin{array}{ll} \min & \boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + 2\boldsymbol{q}^T\boldsymbol{x} \\ \text{sub. to} & -1 \leq x_i \leq 1 \ (i = 0, \dots, K). \end{array} \tag{30}$$

4. 2 Muramatsu-Suzuki Method (1)

This method is a way for the SDP relaxation of 15 as follows.

$$\begin{array}{ll} \min & \sum_{ij \in E} w_{ij}(x_i - x_j)^2 + \boldsymbol{e}^T\boldsymbol{Q}\boldsymbol{e} + 2\boldsymbol{q}^T\boldsymbol{e} \\ \text{sub. to} & X_{ii} = 1 \ (i = 0, \dots, K), \\ & \boldsymbol{X} - \boldsymbol{x}\boldsymbol{x}^T \succeq \boldsymbol{O} \end{array} \tag{31}$$

and

$$\begin{array}{l} \boldsymbol{e}_i\boldsymbol{e}_i^T \ (i = 0, \dots, K), \\ (\boldsymbol{e}_i + \boldsymbol{e}_j)(\boldsymbol{e}_i + \boldsymbol{e}_j)^T \ (0 \leq i < j \leq K), \\ (\boldsymbol{e}_i - \boldsymbol{e}_j)(\boldsymbol{e}_i - \boldsymbol{e}_j)^T \ (0 \leq i < j \leq K) \end{array} \tag{32}$$

are chosen as $\mathcal{C}$ [10]. We can rewrite the constraints of (17) using $\mathcal{C}$.

$$\begin{array}{l} \boldsymbol{C} \bullet (\boldsymbol{x}\boldsymbol{x}^T - \boldsymbol{X}) \\ = \boldsymbol{e}_i\boldsymbol{e}_i^T \bullet (\boldsymbol{x}\boldsymbol{x}^T - \boldsymbol{X}) \\ = \boldsymbol{e}_i\boldsymbol{e}_i^T \bullet \boldsymbol{x}\boldsymbol{x}^T - \boldsymbol{e}_i\boldsymbol{e}_i^T \bullet \boldsymbol{X} \\ = x_i^2 - X_{ii} \leq 0, \end{array} \tag{33}$$

$$\begin{array}{l} \boldsymbol{C} \bullet (\boldsymbol{x}\boldsymbol{x}^T - \boldsymbol{X}) \\ = (\boldsymbol{e}_i + \boldsymbol{e}_j)(\boldsymbol{e}_i + \boldsymbol{e}_j)^T \bullet (\boldsymbol{x}\boldsymbol{x}^T - \boldsymbol{X}) \\ = x_ix_i + x_ix_j + x_jx_i + x_jx_j - (X_{ii} + X_{ij} + +X_{ji} + X_{jj}) \\ = (x_i + x_j)^2 - (X_{ii} + 2X_{ij} + X_{jj}) \leq 0, \end{array} \tag{34}$$

$$\begin{array}{l} \boldsymbol{C} \bullet (\boldsymbol{x}\boldsymbol{x}^T - \boldsymbol{X}) \\ = (\boldsymbol{e}_i - \boldsymbol{e}_j)(\boldsymbol{e}_i - \boldsymbol{e}_j)^T \bullet (\boldsymbol{x}\boldsymbol{x}^T - \boldsymbol{X}) \\ = x_ix_i - x_ix_j - x_jx_i - x_jx_j - (X_{ii} - X_{ij} - X_{ji} - X_{jj}) \\ = (x_i - x_j)^2 - (X_{ii} - 2X_{ij} - X_{jj}) \leq 0. \end{array} \tag{35}$$

Here the object value is achieved using the next relation.

$$\begin{array}{ll} & (x_i - x_j)^2 \implies X_{ii} - 2X_{ij} + X_{jj} = 2 - 2X_{ij} \\ \text{Minimize} & \sum_{ij \in E} w_{ij}(2 - 2X_{ij}) + \boldsymbol{e}^T\boldsymbol{Q}\boldsymbol{e} + 2\boldsymbol{q}^T\boldsymbol{e} \\ \text{subject to} & X_{ii} = 1 \ (i = 0, \dots, K), \\ & x_i^2 - X_{ii} \leq 0 \ (i = 0, \dots, K), \\ & (x_i + x_j)^2 - (X_{ii} + 2X_{ij} + X_{jj}) \leq 0 \ (0 \leq i < j \leq K), \\ & (x_i - x_j)^2 - (X_{ii} - 2X_{ij} + X_{jj}) \leq 0 \ (0 \leq i < j \leq K). \end{array} \tag{36}$$

When we remove X_{ii} and put $z_{ij} = 2 - 2X_{ij}$ $(ij \in E)$, the following problem is obtained. Here $2 + 2X_{ij} = 4 - z_{ij}$.

$$\begin{array}{ll} \text{Minimize} & \displaystyle\sum_{ij\in E} w_{ij}z_{ij} + \boldsymbol{e}^T\boldsymbol{Q}\boldsymbol{e} + 2\boldsymbol{q}^T\boldsymbol{e} \\ \text{subject to} & x_i^2 \le 1 \ (i = 1, \ldots, n), \\ & (x_i + x_j)^2 + z_{ij} \le 4 \ (0 \le i < j \le K), \\ & (x_i - x_j)^2 - z_{ij} \le 0 \ (0 \le i < j \le K). \end{array} \tag{37}$$

However in this case, the optimal solution is automatically obtained as follows.

$$\begin{cases} x_i = 0 \ (i = 1, \ldots, n), \\ z_{ij} = \begin{cases} 4 \ (w_{ij} < 0), \\ 0 \ (w_{ij} \ge 0), \end{cases} (0 \le i < j \le K). \end{cases}$$

4.3 Muramatsu-Suzuki Method(2) [10]

Therefore we consider the next problem with fixing $x_0 = 1$.

$$\begin{array}{ll} \text{Minimize} & \displaystyle\sum_{j=0}^{K} w_{0j}(1 - x_j)^2 + \sum_{0\le i<j\le K} w_{ij}(x_i - x_j)^2 + \boldsymbol{e}^T\boldsymbol{Q}\boldsymbol{e} + 2\boldsymbol{q}^T\boldsymbol{e} \\ \text{subject to} & \boldsymbol{x} \in \{-1, 1\}^{K+1}. \end{array} \tag{38}$$

Here we can use the definition of w_e and the equation

$$(1 - x_j)^2 = 1 - 2x_j + x_j^2 = 2 - 2x_j.$$

(38)can be written as follows.

$$\begin{array}{ll} \text{Minimize} & \displaystyle -\sum_{0\le i<j\le K} Q_{ij}(x_i - x_j)^2 + 2\sum_{j=0}^{K} q_j x_j + \boldsymbol{e}^T\boldsymbol{Q}\boldsymbol{e} \\ \text{subject to} & \boldsymbol{x} \in \{-1, 1\}^{K+1}. \end{array} \tag{39}$$

Then we can derivate the next equation from by adopting the method of MS(1).

$$\begin{array}{ll} \min & \displaystyle -\sum_{0\le i<j\le K} Q_{ij}z_{ij} + 2\sum_{j=0}^{K} q_j x_j + \boldsymbol{e}^T\boldsymbol{Q}\boldsymbol{e} \\ \text{sub. to} & x_i^2 \le 1 \ (i = 0, \ldots, K), \\ & (x_i + x_j)^2 + z_{ij} \le 4, \\ & (x_i - x_j)^2 - z_{ij} \le 0, \\ & (0 \le i < j \le K). \end{array} \tag{40}$$

(Proof) To replace $(x_i - x_j)^2$ into z_{ij} the next relations are used.

$$\begin{array}{rl} & x_i, x_j \in \{-1, 1\}, \ (x_i - x_j)^2 = z_{ij} \\ \Longleftrightarrow & x_i, x_j \in \{-1, 1\}, \ (x_i - x_j)^2 \le z_{ij}, \ (x_i - x_j)^2 \ge z_{ij} \\ \Longleftrightarrow & x_i, x_j \in \{-1, 1\}, \ (x_i - x_j)^2 \le z_{ij}, \ (x_i + x_j)^2 \le 4 - z_{ij} \end{array}$$

When $\{-1,1\}$ constraints are relaxed to continuous function, (40)can be obtained. ∎
However the next relation must hold.

$$\left\{(\boldsymbol{x},\boldsymbol{z})\;\middle|\;\begin{array}{l} x_i^2\le 1\ (i=0,\ldots,K),\\ \left.\begin{array}{l}(x_i+x_j)^2+z_{ij}\le 4,\\ (x_i-x_j)^2-z_{ij}\le 0\end{array}\right\}\\ \qquad (0\le i<j\le K)\end{array}\right\} \supseteq \left\{(\boldsymbol{x},\boldsymbol{z})\;\middle|\;\begin{array}{l} x_i^2\le 1\ (i=0,\ldots,K),\\ \left.\begin{array}{l} x_i+x_j+(2-z_{ij})/2\ge -1,\\ x_i-x_j-(2-z_{ij})/2\ge -1,\\ -x_i-x_j+(2-z_{ij})/2\ge -1,\\ -x_i+x_j-(2-z_{ij})/2\ge -1\end{array}\right\}\\ \qquad (0\le i<j\le K)\end{array}\right\} \tag{41}$$

The next feasible solution is a convex set involving $(1,1,0),(1,-1,4),(-1,1,4),(-1,-1,0)$.

$$\left\{(x_i,x_j,z_{ij})\;\middle|\;\begin{array}{l} -1\le x_i,x_j\le 1,\\ (x_i-x_j)^2\le z_{ij},\\ (x_i+x_j)^2\le 4-z_{ij}\end{array}\right\}$$

On the other hand, the following equation

$$\left\{(x_i,x_j,z_{ij})\;\middle|\;\begin{array}{l} -1\le x_i,x_j\le 1,\\ x_i+x_j+(2-z_{ij})/2\ge -1,\\ x_i-x_j-(2-z_{ij})/2\ge -1,\\ -x_i-x_j+(2-z_{ij})/2\ge -1,\\ -x_i+x_j-(2-z_{ij})/2\ge -1\end{array}\right\}$$

is a convex hull of $(1,1,0),(1,-1,4),(-1,1,4),(-1,-1,0)$. Therefore (41)holds.
Then SOCP relaxation(40) is weaker than LP relaxation(22) as a relaxation, but it is possible to obtain better solution with heuristic method.

5 Comparison of The Relaxation Problem

From the theoretical view point, SDP relaxation with triangle inequalities are stronger than SDP relaxation, LP relaxation, or LP relaxation with triangle inequalities. However, SDP relaxation with triangle inequalities is a rather heavy relaxation for programming techniques and computation. Hence, we simply compare the SDP relaxation, LP relaxation and, LP relaxation with triangle inequalities through numerical experiments.

We show the theoretical relation of largeness of the optimal solution in the figure 6.

$(A) SDP + triangleinequalities$ $\quad (B) SDP$
$(C) SOCP$ $\quad (D) LP + triangleinequalities$
$(E) LP + triangleinequalities$ $\quad (F) SOCP$
$(G) SOCP + triangleinequalities$

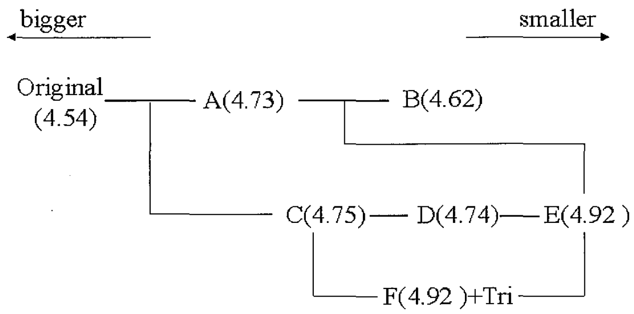

Figure 1: The relation of the objective value

6 Comparison of the relaxation problems by computational experiments

We executed some computational experiments to certify and compare the performance of the proposed filter design method with different algorithms.In the numerical experiments, the specification of the filter design problem is basically same as Lu [9], hence, FIR filter is an odd degree and even symmetric linear phase lowpass filter. The design specification is as follows: the normalized passband is $[0, \omega_p] = [0,0.225]$, stopband is $[\omega_s,1] = [0.275,1]$, $W(\omega) = 1$ on $[0, \omega_p]$, $W(\omega) = 500$ on $[\omega_s,0.5]$, L=12. And, we set each m_k=2. The CPU used is mobile Pentium III 650 MHz, memory is 192 M bytes. All problems are solved by SeDuMi~(Ver.1.03) [12].The CPU time contains only the execution time of SeDuMi. How to obtain an {-1,1}-solution from the solution of the relaxation problems is as follows:

(1) For the SDP relaxation problem, let the solution of the relaxation problem $(\tilde{x}, \tilde{X})$

(1-1) sign($\tilde{x}$)

(1-2) let v_i ($i = 0,\cdots, N-1$) be the eigen vectors of $\tilde{X}$, and set sign(v_i) ($i = 0,\cdots, N-1$).

(1-3) Use the Goemans-Williamson's randomized algorithm [4].

Select the best solution of the above solutions. (Lu [9] exploits (1-1) and (1-2) for the maximal eigen vectors, However, we recommend the above, since it does not take so much CPU time, and may improve the solution.)

(2) For LP relaxation problems, we exploited the sign vector v_i of the solution of the relaxation problems $\tilde{x}$. In Figure 2, the comparison of the computational time is shown.

7 Examination

We compared LP relaxation methods and LP relaxation methods which have triangle inequalities with SDP relaxation method by Lu [9] through numerical experiments.

Table 1: Comparisons of upper bounds ($*$ is the best solution).N=3,...,59

N	WLS	LP (22)	LP + Tri (23)	SDP (17)	SOCP (5.1)	SOCP (40)
3	1.33298000	1.33300000*	1.33300000*	1.33300000*	0.13330000*	0.13330000*
5	0.53726000	0.62513000	0.57498000*	0.57498000*	0.62513000	0.57498000*
7	0.29413000	0.42797000*	0.42797000*	0.42797000*	0.81664000	0.67103000
9	0.22475000	0.37866000	0.24704000*	0.24704000*	0.03786600	0.37866000
11	0.11542000	2.53401000	0.13730000*	0.13730000*	0.01906700	0.16360000
13	0.10269000	0.10543000*	0.10543000*	0.10543000*	0.1068400	0.11025000
15	0.05214000	0.22109000	0.13274000*	0.13274000*	0.2439700	0.23921000
17	0.04903000	0.69010000	0.15858000*	0.23590000	0.2447400	0.2692700
19	0.02473000	0.08771000*	0.08771000*	0.0877100*	0.9404000	0.10978000
21	0.02385000	0.06999000	0.06875000*	0.06933000	0.0693100	0.007409000
23	0.01197000	0.11259000	0.03368000*	0.03368000*	0.0621900	0.05704000
25	0.01169000	0.02655000	0.02606000*	0.02606000*	0.0978400	0.06608000
27	0.00585000	0.08714000	0.01104000*	0.01689000	0.06417000	0.01948000
29	0.00576000	0.02310000	0.02308000*	0.02360000	0.22940000	0.06722000
31	0.00288000	0.01258000*	0.01258000*	0.0141500	0.18130000	0.01592000
33	0.00285000	0.04553000*	0.04553000*	0.04553000*	0.1778600	0.0506800
35	0.00142000	0.04666000	0.03882000*	0.03938000	0.16596000	0.0401300
37	0.00141000	0.06321000	0.06231000*	0.06263000	0.14685000	0.06774000
39	0.00071000	0.08347000	0.08196000*	0.08196000*	0.12704000	0.08547000
41	0.00070000	0.11712000*	0.11711000*	0.1172100	0.13182000	0.01191400
43	0.00035000	0.15770000	0.08531000*	0.08531000*	0.12777000	0.1028400
45	0.00035000	0.09710000	0.08983000*	0.09100000	0.12320000	0.0119100
47	0.00018000	0.15553000	0.05823000*	0.05823000*	0.13123000	0.07244000
49	0.00018000	0.05953000	0.05781000*	0.05781000*	0.13152000	0.07174000
51	0.00009000	0.12833000	0.06984000*	0.07022000	0.08614000	0.09478000
53	0.00009000	0.07236000	0.06895000*	0.07080000	0.08600000	0.10152000
55	0.00004000	0.04949000	0.04946000*	0.04979000	0.13261000	0.05069000
57	0.00004000	0.05035000*	0.05035000*	0.0503500*	0.15121000	0.05320000
59	0.00002000	0.05359000	0.05345000*	0.05350000	0.10096600	0.05643000

By the numerical experiments, we found (1) all the solutions of LP with triangle inequalities (23) are optimal solutions of the original problem (12), that is, all the solutions of (23) are {-1, 1}-integer solutions and automatically optimal solutions for (12) Numerical experiments show that it is worthwhile to add triangle inequalities to LP relaxations.

As for the computational time, the filter design algorithm using SOCP relaxation (Kim and Kojima) is fastest. Then comes the filter design algorithm using SDP relaxation. Even if we use LP relaxation with triangle inequalities, the computational time is at most 8 minutes. For N less than or equal to 35, each algorithm solves the problems within 10 seconds. At now, all triangle inequalities are included in the relaxation problems, we can also develop an algorithm that has triangle inequalities as cutting planes.

Table 2: Comparisons of upper bounds ($*$ is the best solution).N=61,...,101

N	WLS	LP (22)	LP + Tri (23)	SDP (17)	SOCP (5.1)	SOCP (40)
61	0.00002000	0.05425000	0.05409000*	0.05409000*	0.1096400	0.05712000
63	0.00001136	0.06680117	0.06023354*	0.06922768	0.10918304	0.07255869
65	0.00001135	0.06076647	0.06037857*	0.07385944	0.10237367	0.07921838
67	0.00000576	0.06485766	0.06436583*	0.07316518	0.09888601	0.07799757
69	0.00000576	0.07369399	0.06372345*	0.07403431	0.09675582	0.07912869
71	0.00000293	0.07242553	0.06856167*	0.08889189	0.09644421	0.09354363
73	0.00000293	0.07988543	0.06845793*	0.08370451	0.09012794	0.08727217
75	0.00000149	0.08863230	0.07385780*	0.08370117	0.09013248	0.08727230
77	0.00000149	0.08985698	0.07316518*	0.07786689*	0.08571961	0.08288307
79	0.00000076	0.08408650	0.07338401*	0.07756317	0.08562250	0.07902403
81	0.00000076	0.10605943	0.08998092	0.06683611*	0.08275478	0.07000302
83	0.00000039	0.09294880	0.08370451*	0.06638491	0.08275476	0.07005689
85	0.00000039	0.09385693	0.08370117*	0.06613772*	0.08907031	0.06936813
87	0.00000020	0.08346876	0.07759562*	0.06616929	0.08910672	0.06927017
89	0.00000020	0.08614930	0.07748277*	0.06341097	0.08257557	0.06643142
91	0.00000010	0.06958735	0.06537696*	0.06360760	0.08257557	0.06643142
93	0.00000010	0.07349370	0.06537680*	0.06638491	0.08275476	0.07005689
95	0.00000005	0.07015094	0.06596579*	0.06613772	0.08907031	0.06936813
97	0.00000005	0.06964551	0.06560719*	0.06616929	0.08910672	0.06927017
99	0.00000003	0.06824348	0.06333848*	0.06341097	0.08257557	0.06643142
101	0.00000003	0.06658692	0.06333839*	0.06360760	0.08257557	0.06643142

There is a result of two type of SOCP relaxation, SOCP(1) and SOCP(2). SOCP(1) is Kim-Kojima's method and SOCP(2) is Muramatsu-Suzuki method. As for computational time of SOCP, it took least seconds to obtain the solution in using SOCP(1). On the contrary to the theory, computational time of SOCP(2) is bigger than LP and SDP in the range from 3 to 101(N). However the number of variables increase, it is possible that the computational time of SOCP(1) and SOCP(2) is smaller than other methods clearly.

8 Conclusion

In this thesis, we proposed LP based on the relaxation techniques to solve the design problem of FIR filters with SP2 coefficients under LMS criterion. And, we compared this relaxation technique and the SDP relaxation through numerical experiments. By the numerical example, LP based on the relaxation technique seems to work fairly good. To contrary to our predict, computational time of SOCP(2) is bigger than SDP when variable N=101. We try o find the

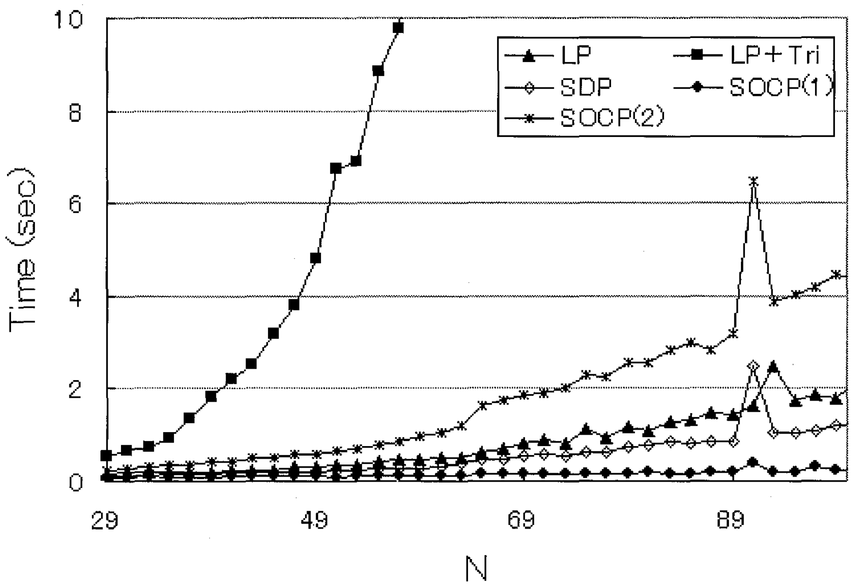

Figure 2: the comparison of the computatiomal time

cause and need to execute numerical experiments on bigger scale.

References

[1] E.D. Andersen, C. Roos, T. Terlaky, “On implementing a primal-dual interior-point method forconic quadrratic optimization,” Helsink School of Economics and Buisiness Administration, Helsinki, Finland.

[2] L. Faybusovich, 1995, “Jordan algebras, symmetric cones and interior point methods,” Technical Report, Department of Mathematics, University of Notre Dame, Notre Dame, IN, USA.

[3] T. Fujie and M. Kojima, 1997, “Semidefinite relaxation for nonconvex programs,” *Jornal of Global Optimization*, vol. 10, pp. 367-380.

[4] M.X. goemans nad S.P. williamson, 1995, “Improves approximation algoriths for maximum cut and satisfiability programs using semidefinite programming,” *Jornal of Assoc. Comut. Math.*, vol. 42, pp. 1115-1145.

[5] S. Kim and M. Kojima, 2001, “Second Order Cone Programming Relaxation of Nonconvex Quadratic Optimization Problems”, Vol .15, No.3-4 , 201-224.

[6] M.Kojima,S.Shindoh, and S.Hara, 1997, “Interior-point methods for the monotone semidefinite linear complementarity problems,” *Soietiy for Industrial Applied Mathiematics (SIAM) Journal on Optimization*, Vol.7, pp.86-125.

[7] Y.C. Lim and S.R. Parker, 1983, “FIR filter design over a discrete power-of-two coefficient space,” *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-31, pp.583-591.

[8] Y.C. Lim, R. Yang, D. Li and J. Song, 1998, “Signed power-of-two (SPT) term allocation scheme for the design of digital filters,” *IEEE Trans. Circuits and Systems, Analog and Digital*

Signal Processing, vol.46, No.5, pp.577-584.

[9] W.-S. Lu, 2001, "Design of FIR Filters with discrete coefficients: A Semidefinite Programming Relaxation Approach", *Proc. IEEE International Symposium on Circuits and Systems*,, Vol. II pp.297-300.

[10] M. Muramatsu and T Suzuki, 2001, "A new second-order cone programming relaxation," Technical Report, CS-01-01, Department of computer Science, the Universigy of Electro-Communications, Tokyo, Japan.

[11] C.H. Papadimitriou and K. Steiglits, 1982, *Combinatorial Optimzation*, Prentice-Hall.

[12] J.F. Sturm, SeDuMi Version 1.03, (Download link: http:// www2.Unimaas.nl/ ~sturm/ sofware/ sedumi.html).

[13] H. Wolkowica, R. Saigal, and L. Vandenberghe ed., 2000, *Handbook on Semidefinite Programming*, Kluwer Academic.