Proceedings of the 41st IEEE
Conference on Decision and Control
Las Vegas, Nevada USA, December 2002

Sampled-Data Design of Interpolators Using The Cutting-Plane Method

Yuji Wakasa [1] Daisuke Yasufuku [2] Masaaki Nagahara [3] Yutaka Yamamoto [4]

Abstract

This paper presents a practically efficient method for designing an interpolator that minimizes the L^2-induced norm of the error system between the interpolator and a time-delay. The method is based on the so-called cutting-plane method for nondifferentiable convex optimization. The advantage of the proposed method is that it can solve design problems of practical size with a reasonable amount of computation. Numerical examples show the effectiveness of the proposed method in comparison with the conventional ones.

1 Introduction

Multirate signal processing techniques are now popular in digital signal processing and are used in many digital devices such as A/D, D/A converters, sample-rate converters and multirate filterbanks [8]. One of the most fundamental elements in such devices is an interpolator (or a decimator) which consists of an upsampler (or a downsampler) and a digital filter. It is therefore important to design a digital filter so that the whole multirate system achieves a desirable performance.

While the conventional design methods for digital filters are developed in the discrete-time domain and deal with continuous-time performance only indirectly, the recently proposed filter design methods [3, 5, 6, 7, 11, 12] via sampled-data control theory have made it possible to take analog-domain performance into account. In [3], it is shown that the design problem of an FIR interpolation filter is reduced to an optimization problem involving linear matrix inequalities (LMIs). However this technique is not applicable to design problems with practical upsampling factors such as $M = 32, 64$[1] because the sizes of the corresponding LMIs are too large to be numerically solved with today's advanced computing power. On the other hand, Nagahara and Yamamoto [6] presents that such large-scale design problems can be solved by decomposing them into some solvable subproblems. In this case, however, performance of the whole system is not guaranteed.

In this paper, we reduce the interpolator design problem to a convex optimization problem, and solve it using the cutting-plane method [4], instead of reducing it to LMIs. The cutting-plane method is an effective optimization method for a convex but nondifferentiable function, and has been shown to be effective in controller design applications [1]. By using the cutting-plane method, we show that the computational burden for the interpolator design is greatly reduced and that the problems with practical upsampling factors are solvable within a reasonable CPU time though they are intractable by the conventional methods.

2 Problem Formulation

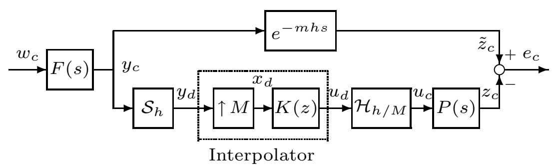

Figure 1: Signal reconstruction error system.

Consider the error system depicted in Figure 1. The incoming signal w_c first goes through an analog filter $F(s)$ and the filtered signal y_c becomes nearly (but not entirely) band-limited. $F(s)$ governs the frequency-domain characteristic of the analog signal y_c. This signal is then sampled by $\mathcal{S}_h$ to become a discrete-time signal y_d with sampling period h. This signal is usually stored or transmitted with some media (e.g., CD).

To restore y_c we usually let it pass through a digital filter, a hold device and then an analog filter. The present setup however places yet one more step: The discrete-time signal y_d is first upsampled by $\uparrow M$,

$$\uparrow M : y_d \mapsto x_d : x_d[k] = \begin{cases} y_d[\nu], \ k = M\nu, \ \nu = 0, 1, \dots \\ 0, \quad \text{otherwise} \end{cases}$$

[1] wakasa@i.kyoto-u.ac.jp, Department of Applied Analysis and Complex Dynamical Systems, Graduate School of Informatics, Kyoto University, Kyoto 606-8501, JAPAN.
[2] yasufuku@acs.i.kyoto-u.ac.jp, the same address as above.
[3] nagahara@acs.i.kyoto-u.ac.jp, the same address as above.
[4] yy@i.kyoto-u.ac.jp, the same address as above.

[1]In many commercial CD players, upsampling factors $M = 32, 64$ are adopted these days.

0-7803-7516-5/02/$17.00 ©2002 IEEE

and becomes another discrete-time signal x_d with sampling period h/M. The discrete-time signal x_d is then processed by a digital filter $K(z)$, becomes a continuous-time signal u_c by going through the 0-order hold $\mathcal{H}_{h/M}$ (that works in sampling period h/M), and then becomes the final signal by passing through an analog filter $P(s)$. An advantage here is that one can use a fast hold device $\mathcal{H}_{h/M}$ thereby making more precise signal restoration possible. The delay in the upper portion of the diagram corresponds to the fact that we allow a certain amount of time delay for signal reconstruction.

The system $K(z)(\uparrow M)$ is called an interpolator. The objective here is to design the digital FIR filter $K(z)$ for given $F(s)$ and $P(s)$ so that the error of signal restoration is minimized. Let $K(z)$ be an FIR filter, since the FIR filter is often used in digital devices due to numerical stability and ease of implementation. Also let $\mathcal{T}_{ew}$ denote the input/output operator from w_c to e_c. Our design problem is as follows:

Problem 1 Given stable and strictly proper $F(s)$, stable and proper $P(s)$, an upsampling factor M and a delay factor m, find an FIR filter $K(z)$ that minimizes the L^2-induced norm of $\mathcal{T}_{ew}$:

$$\|\mathcal{T}_{ew}\| := \sup_{w_c \in L^2[0,\infty)} \frac{\|\mathcal{T}_{ew} w_c\|_2}{\|w_c\|_2}.$$

3 Interpolator Design

A difficulty in Problem 1 is that it contains the upsampler $\uparrow M$, so that it makes the overall system multirate. Another difficulty is that it involves a continuous time-delay, and hence it is an infinite-dimensional problem.

In [3], it is shown that Problem 1 is reduced to an equivalent single-rate finite-dimensional problem. However the assumption required for deriving the equivalent discrete-time system is not often satisfied because it is quite strict in practical signal processing settings [9]. To avoid this difficulty, the fast sample/fast hold (FSFH) approximation is known to be effective [9, 10, 11]. This method approximates continuous-time inputs and outputs (w_c, e_c) with discrete-time inputs and outputs (denoted w_d and e_d, respectively) via a sampler and hold that operate in the period h/N where N is an integer.

The FSFH method needs no stringent assumptions such as the one above, and reduce the infinite-dimensional problem to a finite-dimensional problem. However the obtained discrete-time system is of quite high order, so that it tends to be difficult to solve the corresponding design problem via LMI approaches because of computational burden. We here solve the problem using the cutting-plane method which is a kind of convex optimization method. Owing to this method, such a design problem is shown to be solvable within a reasonable CPU time.

In this section, we first reduce Problem 1 to a convex optimization problem for an approximate discrete-time system via the FSFH method, and then apply the cutting-plane method to it.

3.1 Reduction to a convex optimization problem

For the sake of simplicity, suppose that we make use of l steps of the discrete-time signal y_d with sampling period h with an FIR filter K with upsampling factor M. Then K becomes an Ml-tap filter and is represented by

$$K(z, \boldsymbol{\alpha}) := \sum_{n=1}^{Ml} \alpha_n z^{-n+1},$$

where $\boldsymbol{\alpha} := [\alpha_1 \ \cdots \ \alpha_{Ml}]^T \in \Re^{Ml}$ is a design parameter.

In order to reduce the multirate problem into a single-rate problem, we begin by defining the downsampler $\downarrow M$ by

$$\downarrow M : x_d \mapsto y_d, \ y_d = x_d[Mk], \ k = 0, 1, \ldots,$$

and the discrete-time lifting $\mathbf{L}_M$ and its inverse $\mathbf{L}_M^{-1}$ as

$$\begin{aligned} \mathbf{L}_M &:= (\downarrow M) \begin{bmatrix} 1 \\ z \\ \vdots \\ z^{M-1} \end{bmatrix}, \\ \mathbf{L}_M^{-1} &:= \begin{bmatrix} 1 \ z^{-1} \ \cdots \ z^{-M+1} \end{bmatrix} (\uparrow M). \end{aligned}$$

As shown in Figure 2, the operator $\mathbf{L}_M$ stacks together the discrete-time signals

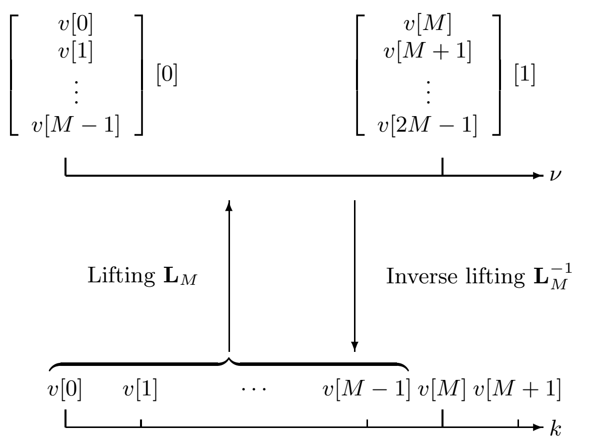

Figure 2: Discrete-time lifting.

$v[k],\ k = 0,1,\ldots$ to vector-valued signals $[v[M\nu], v[M\nu+1], \ldots, v[M\nu+M-1]]^T,\ \nu = 0,1,\ldots$ and increases the sampling period by M. Also define the generalized hold $\tilde{\mathcal{H}}_h$ that works in sampling period h

$$
\begin{aligned}
&\tilde{\mathcal{H}}_h : (l^2)^M \ni \tilde{u}_d \mapsto u_c \in L^2,\ u_c(kh+\theta) = \mathbf{H}(\theta)\tilde{u}_d[k],\\
&\qquad \theta \in [0,h),\ k = 0,1,2,\ldots,\\
&\mathbf{H}(\theta) := \begin{cases} [1\ 0\ 0\ \cdots\ 0],\ \theta \in [0, h/M)\\ [0\ 1\ 0\ \cdots\ 0],\ \theta \in [h/M, 2h/M)\\ \quad\cdots\\ [0\ 0\ 0\ \cdots\ 1],\ \theta \in [(M-1)h/M, h). \end{cases}
\end{aligned}
$$

Then it is seen that $\tilde{\mathcal{H}}_h = \mathcal{H}_{h/M}\mathbf{L}_M^{-1}$ [3, 7]. Moreover define the variable of z-transform for sampling period h by $\zeta := z^M$ and the lifting of the interpolator by $\tilde{K}(\zeta,\boldsymbol{\alpha}) := \mathbf{L}_M K(z,\boldsymbol{\alpha})(\uparrow M)$. It follows from these definitions that

$$
\mathcal{H}_{h/M} K(z,\boldsymbol{\alpha})(\uparrow M) = \tilde{\mathcal{H}}_h \tilde{K}(\zeta,\boldsymbol{\alpha}). \tag{1}
$$

We see from (1) that the multirate system in Figure 1 is equivalently reduced to a single-rate system with sampling period h. The filter $\tilde{K}(\zeta,\boldsymbol{\alpha})$, i.e., the lifting of the interpolator $K(z,\boldsymbol{\alpha})(\uparrow M)$ can be written as [3]

$$
\tilde{K}(\zeta,\boldsymbol{\alpha}) = W(\boldsymbol{\alpha})V(\zeta), \tag{2}
$$

where

$$
\begin{aligned}
W(\boldsymbol{\alpha}) &:= \begin{bmatrix} \alpha_1 & \alpha_{M+1} & \cdots & \alpha_{M(l-1)+1}\\ \alpha_2 & \alpha_{M+2} & \cdots & \alpha_{M(l-1)+2}\\ \vdots & \vdots & \ddots & \vdots\\ \alpha_M & \alpha_{2M} & \cdots & \alpha_{Ml} \end{bmatrix},\\
V(\zeta) &:= \left[1, \zeta^{-1}, \ldots, \zeta^{-(l-1)}\right]^T.
\end{aligned}
$$

The single-input/M-output filter $\tilde{K}(\zeta,\boldsymbol{\alpha})$ is the one to be designed.

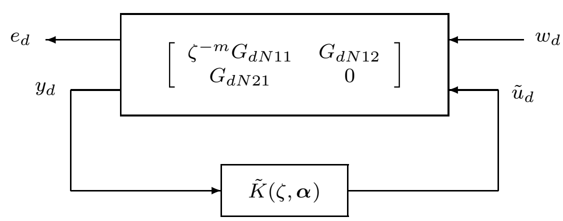

Figure 3: Discrete-time system via the FSFH approximation.

We next derive an approximate discrete-time system via the FSFH method. Suppose that the sampling period h/M in the interpolator is divided into L subintervals by the FSFH method. As a result, the sampling period h in the above single-rate system is divided into $N := ML$ subintervals. For the continuous-time system

$$
\begin{bmatrix} F(s) & -P(s)\\ F(s) & 0 \end{bmatrix} =: \left[\begin{array}{c|cc} A_c & B_{c1} & B_{c2}\\ \hline C_{c1} & 0 & D_{c12}\\ C_{c1} & 0 & 0 \end{array}\right],
$$

the approximate discrete-time system as shown in Figure 3 is given by the following formulas [11]:

$$
\begin{bmatrix} G_{dN11}(\zeta) & G_{dN12}(\zeta)\\ G_{dN21}(\zeta) & 0 \end{bmatrix} =: \left[\begin{array}{c|cc} A_d & B_{N1} & B_{N2}\\ \hline C_{N1} & D_{N11} & D_{N12}\\ C_{N2} & 0 & 0 \end{array}\right],
$$

$$
\begin{aligned}
A_d &:= e^{A_c h}, \quad A_f := e^{A_c h/N},\\
[B_{1f}\quad B_{2f}] &:= \int_0^{h/N} e^{A_c t}\,[B_{c1}\quad B_{c2}]\,dt,\\
B_{N1} &= \left[A_f^{N-1}B_{1f}\ \ A_f^{N-2}B_{1f}\ \ \cdots\ \ B_{1f}\right],\\
B_{N2} &= \left[A_f^{N-1}B_{2f}\ \ A_f^{N-2}B_{2f}\ \ \cdots\ \ B_{2f}\right]\mathcal{Q},\\
C_{N1} &= \begin{bmatrix} C_{c1}\\ C_{c1}A_f\\ \vdots\\ C_{c1}A_f^{N-1} \end{bmatrix},\\
C_{N2} &= C_{c1},\\
D_{N11} &= \\
&\begin{bmatrix} 0 & 0 & \cdots & 0\\ C_{c1}B_{1f} & 0 & \cdots & 0\\ C_{c1}A_f B_{1f} & C_{c1}B_{1f} & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ C_{c1}A_f^{N-2}B_{1f} & C_{c1}A_f^{N-3}B_{1f} & \cdots & 0 \end{bmatrix},\\
D_{N12} &= \\
&\begin{bmatrix} D_{c12} & 0 & \cdots & 0\\ C_{c1}B_{2f} & D_{c12} & \cdots & 0\\ C_{c1}A_f B_{2f} & C_{c1}B_{2f} & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ C_{c1}A_f^{N-2}B_{2f} & C_{c1}A_f^{N-3}B_{2f} & \cdots & D_{c12} \end{bmatrix}\mathcal{Q},\\
\mathcal{Q} &:= \mathrm{diag}\{\underbrace{\tilde{\mathcal{Q}}, \ldots, \tilde{\mathcal{Q}}}_{M}\} \in \Re^{N\times M},\\
\tilde{\mathcal{Q}} &:= [1, \ldots, 1]^T \in \Re^L.
\end{aligned}
$$

From Figure 3, the approximate discrete-time system of $\mathcal{T}_{ew}$ is

$$
\begin{aligned}
T_{dN,ew}(\zeta,\boldsymbol{\alpha}) &:= \zeta^{-m}G_{dN11}(\zeta)\\
&\quad + G_{dN12}(\zeta)\tilde{K}(\zeta,\boldsymbol{\alpha})G_{dN21}(\zeta). \quad (3)
\end{aligned}
$$

For this approximate system,

$$
\lim_{N\to\infty} \|T_{dN,ew}\|_\infty = \|\mathcal{T}_{ew}\|
$$

is guaranteed [10].

It is seen from (2) and (3) that $T_{dN,ew}(\zeta,\boldsymbol{\alpha})$ is affine in the design parameter $\boldsymbol{\alpha}$. Therefore, there exist $H_{ew,i}(\zeta), i = 0,\ldots,Ml$ such that $T_{dN,ew}(\zeta,\boldsymbol{\alpha})$ is rewritten as

$$
T_{dN,ew}(\zeta,\boldsymbol{\alpha}) = H_{ew,0}(\zeta) + \sum_{i=1}^{Ml} \alpha_i H_{ew,i}(\zeta).
$$

Summarizing, Problem 1 is reduced to the following problem.

Problem 2 Given stable and strictly proper $F(s)$, stable and proper $P(s)$, an upsampling factor M, a delay factor m and natural numbers l, L, find $\boldsymbol{\alpha} \in \Re^{Ml}$ that minimizes

$$\phi(\boldsymbol{\alpha}) := \|T_{dN,ew}(\zeta, \boldsymbol{\alpha})\|_\infty. \tag{4}$$

In the same manner as in [3], Problem 2 can be described by LMIs. For practical upsampling factors such as $M = 32, 64$, however, the approximate discrete-time systems are of high orders and the corresponding Riccati solutions are of large size. The resulting LMIs are too large to be solved with today's advanced PCs. On the other hand, ϕ is convex with respect to $\boldsymbol{\alpha}$, and therefore Problem 2 is a convex optimization problem. Also, the variable in Problem 2 is only $\boldsymbol{\alpha}$ whose size is irrelevant to N and m. It is thus more effective to solve Problem 2 directly by a convex optimization method.

3.2 Solution via the cutting-plane method

The function ϕ (4) is convex but not in general differentiable [1]. Because of this, we cannot apply gradient-based optimization methods to Problem 2. However, when a subgradient of ϕ can be obtained, we can apply the cutting-plane method [4] to this problem. The subgradient is defined as follows.

Definition 1 Suppose that $f : \Re^n \to \Re$ is convex and $\boldsymbol{x}_0 \in \Re^n$. Then we say that $g \in \Re^n$ is a *subgradient* of f at $\boldsymbol{x}_0$ if

$$f(\boldsymbol{x}) \geq f(\boldsymbol{x}_0) + g^T(\boldsymbol{x} - \boldsymbol{x}_0) \quad \text{for all } \boldsymbol{x} \in \Re^n.$$

When f is nondifferentiable at $\boldsymbol{x}_0$, the subgradient is not uniquely determined. To use the cutting-plane method, however, we have only to compute one subgradient at every point. The following lemma gives a subgradient of ϕ at $\boldsymbol{\alpha}_0$ [1].

Lemma 1 *Let ω_0 be a frequency at which $\phi(\boldsymbol{\alpha}_0)$ (i.e., $\|T_{dN,ew}(\zeta, \boldsymbol{\alpha}_0)\|_\infty$) is achieved. Also, for the singular value decomposition*

$$T_{dN,ew}(e^{j\omega_0}, \boldsymbol{\alpha}_0) = U\Sigma V^*,$$

let u_0 and v_0 be the first columns of U and V, respectively. Then a subgradient of ϕ at $\boldsymbol{\alpha}_0$ is given by

$$\phi^{\rm sg}(\boldsymbol{\alpha}_0) := \begin{bmatrix} \mathrm{Re}(u_0^* H_{ew,1}(e^{j\omega_0})v_0) \\ \vdots \\ \mathrm{Re}(u_0^* H_{ew,Ml}(e^{j\omega_0})v_0) \end{bmatrix}. \tag{5}$$

Proof: Let $\boldsymbol{\alpha}_0 = [\alpha_{0,1}, \alpha_{0,2}, \ldots, \alpha_{0,Ml}]^T$. Then the following relation holds for all $\boldsymbol{\alpha} \in \Re^{Ml}$.

$$\begin{aligned}
& \phi(\boldsymbol{\alpha}_0) + (\phi^{\rm sg}(\boldsymbol{\alpha}_0))^T(\boldsymbol{\alpha} - \boldsymbol{\alpha}_0) \\
= \; & u_0^* T_{dN,ew}(e^{j\omega_0}, \boldsymbol{\alpha}_0)v_0 \\
& + \begin{bmatrix} \mathrm{Re}(u_0^* H_{ew,1}(e^{j\omega_0})v_0) \\ \vdots \\ \mathrm{Re}(u_0^* H_{ew,Ml}(e^{j\omega_0})v_0) \end{bmatrix}^T (\boldsymbol{\alpha} - \boldsymbol{\alpha}_0) \\
= \; & \mathrm{Re}\left(u_0^* H_{ew,0}(e^{j\omega_0})v_0 + \sum_{i=1}^{Ml} \alpha_{0,i} u_0^* H_{ew,i}(e^{j\omega_0})v_0 \right) \\
& + \mathrm{Re}\left(\sum_{i=1}^{Ml} (\alpha_i - \alpha_{0,i}) u_0^* H_{ew,i}(e^{j\omega_0})v_0 \right) \\
= \; & \mathrm{Re}\left(u_0^* H_{ew,0}(e^{j\omega_0})v_0 + \sum_{i=1}^{Ml} \alpha_i u_0^* H_{ew,i}(e^{j\omega_0})v_0 \right) \\
= \; & \mathrm{Re}\left(u_0^* T_{dN,ew}(e^{j\omega_0}, \boldsymbol{\alpha})v_0 \right) \\
\leq \; & \sup_{\|u\|=\|v\|=1, \omega \in \Re} \mathrm{Re}(u^* T_{dN,ew}(e^{j\omega}, \boldsymbol{\alpha})v) \\
= \; & \phi(\boldsymbol{\alpha}).
\end{aligned}$$

Hence (5) gives a subgradient of f at $\boldsymbol{\alpha}_0$. ∎

We apply the following cutting-plane algorithm [1, 4] to Problem 2.

Cutting-plane algorithm

Step 0. Determine search regions $\{\boldsymbol{\alpha} \,|\, \boldsymbol{\alpha}_{\min} \leq \boldsymbol{\alpha} \leq \boldsymbol{\alpha}_{\max}\}$, $\{L \,|\, L_{\min} \leq L \leq L_{\max}\}$, select $\boldsymbol{\alpha}_1$ and set ϵ.

Step 1. Set $k = 1$, and repeat Steps 2 – 5 until the following stopping criterion is satisfied:

$$(U_k - L_k)/U_k \leq \epsilon.$$

Step 2. Compute $\phi(\boldsymbol{\alpha}_k)$ and $\phi^{\rm sg}(\boldsymbol{\alpha}_k)$.

Step 3. Solve the following linear program to find a lower bound L_k of the optimal value $\phi^{\rm opt}$ and an optimal solution $\boldsymbol{\alpha}_k^{\rm opt}$:

$$L_k = \min\{c^T\xi \,|\, A\xi \leq b,\ \xi_{\min} \leq \xi \leq \xi_{\max}\},$$

where

$$A = \begin{bmatrix} (\phi^{\rm sg}(\boldsymbol{\alpha}_1))^T & -1 \\ \vdots & \vdots \\ (\phi^{\rm sg}(\boldsymbol{\alpha}_k))^T & -1 \end{bmatrix}, \quad \xi = \begin{bmatrix} \boldsymbol{\alpha} \\ L \end{bmatrix},$$

$$b = \begin{bmatrix} (\phi^{\rm sg}(\boldsymbol{\alpha}_1))^T \boldsymbol{\alpha}_1 - \phi(\boldsymbol{\alpha}_1) \\ \vdots \\ (\phi^{\rm sg}(\boldsymbol{\alpha}_k))^T \boldsymbol{\alpha}_k - \phi(\boldsymbol{\alpha}_k) \end{bmatrix}, \quad c = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\xi_{\min} = \begin{bmatrix} \boldsymbol{\alpha}_{\min} \\ L_{\min} \end{bmatrix}, \quad \xi_{\max} = \begin{bmatrix} \boldsymbol{\alpha}_{\max} \\ L_{\max} \end{bmatrix}.$$

Step 4. Compute an upper bound of the optimal value ϕ^{opt}

$$U_k = \min_{1 \le i \le k} \phi(\boldsymbol{\alpha}_i).$$

Step 5. Set $\boldsymbol{\alpha}_{k+1} = \boldsymbol{\alpha}_k^{\text{opt}}$ and replace k by $k+1$.

Remark 1 In Step 2, the H^∞ norm $\phi(\boldsymbol{\alpha}_k)$ and the frequency achieving it need to be computed. These values can be efficiently computed by the method based on [2] which is implemented as the command `dnorminf` in MATLAB.

4 Numerical Examples

We present design examples for

$$\begin{aligned} F(s) &= \frac{1}{(Ts+1)(10Ts+1)}, \quad T := \frac{2.205}{\pi}, \\ P(s) &= 1 \end{aligned}$$

$h = 1$, $m = 2$, $l = 2$, $L = 2$. The low-pass filter $F(s)$ has cutoff frequencies of $\pi/22.05$ rad/sec $= 1/44.1$ Hz and $\pi/2.205$ rad/sec $= 1/4.41$ Hz. For the sampling frequency 44.1 KHz of audio CDs, these cutoff frequencies of this filter correspond to $1/44.1 \times 44.1$ KHz $= 1$ KHz and $1/4.41 \times 44.1$ KHz $= 10$ KHz. This frequency-domain characteristic simulates that of a typical orchestral music. All computation has been done with MATLAB on a PC with Intel Pentium III 500 MHz CPU.

Example 1
To compare the computational efficiency of the proposed method and the conventional LMI approach [3][2], we design interpolators for various upsampling factors M by these methods. Set $\epsilon = 0.05$, $\boldsymbol{\alpha}_1 = [1, \ldots, 1]$, and a search region $\{\boldsymbol{\alpha} \in \Re^{Ml} : -10 \le \alpha_i \le 10, i = 1, \ldots, Ml\}$. Figure 4 shows CPU times for solving the design problems. For $M = 32$, the conventional LMI approach does not terminate within 10 hours. Table 1 shows the sizes and the numbers of variables of the LMIs corresponding to various values of M. It is seen from Figure 4 that the problems for $M = 32, 64$ can be solved within reasonable CPU times.

Example 2
For upsampling factor $M = 32$, we next compare performance of the interpolator designed by the proposed method with that of two conventional interpolators; (i) one is an interpolator designed by decomposing the design problem for $M = 32$ into two problems for $M = 2, 16$ and concatenating them [6] (the upsampling factor of this interpolator corresponds to $M = 2 \times 16 = 32$); (ii) the other is an equiripple filter of order 500. Figures 5, 6 and 7 show the time responses against a rectangular wave. The interpolator designed via decomposition with $M = 2, 16$ appears to be quite good but has a slightly slow rising and falling response in comparison with the interpolator designed by the proposed method. The equiripple filter show a large amount of ringing, whereas the one designed by the proposed method has much less peak around the edge.

[2]The FSFH method is used here, while it is not used in [3].

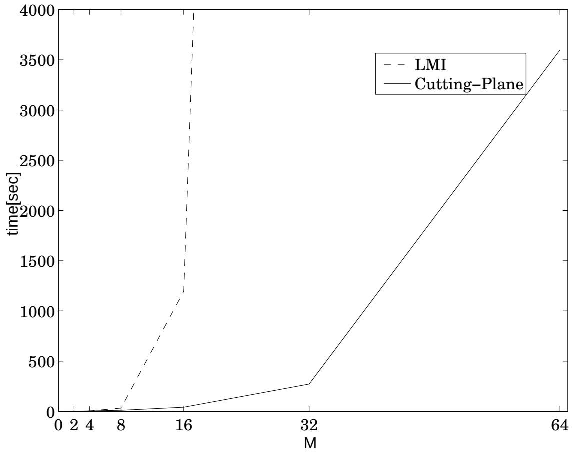

Figure 4: Upsampling factors and CPU times.

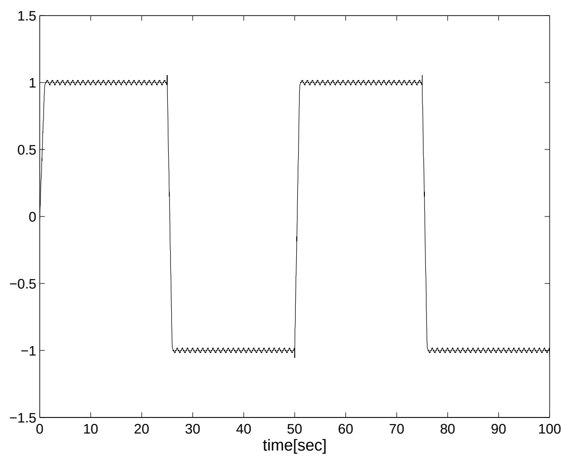

Figure 5: Time response against a rectangular wave (sampled-data design with $M = 32$).

5 Conclusions

We have presented a sampled-data H^∞ design method of interpolators by using the cutting-plane method. By the proposed method, it has been shown that the design problems with practical upsampling factors $M = 32, 64$ can be solved within reasonable CPU times. While

Table 1: Upsampling factors M and LMIs.

M	4	8	16	32	64
Size	42×42	74×74	138×138	266×266	522×522
Number of variables	100	248	736	2480	9040

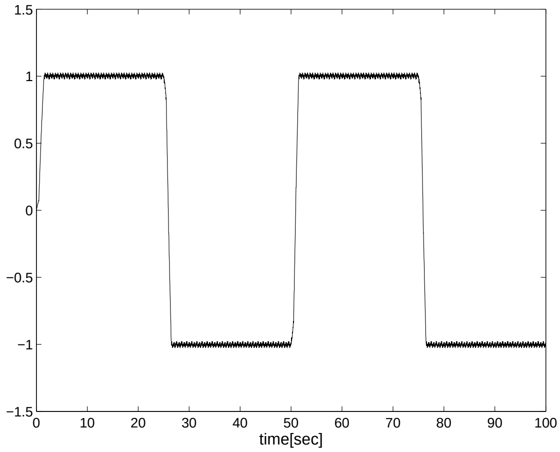

Figure 6: Time response against a rectangular wave (sampled-data design with $M = 2, 16$).

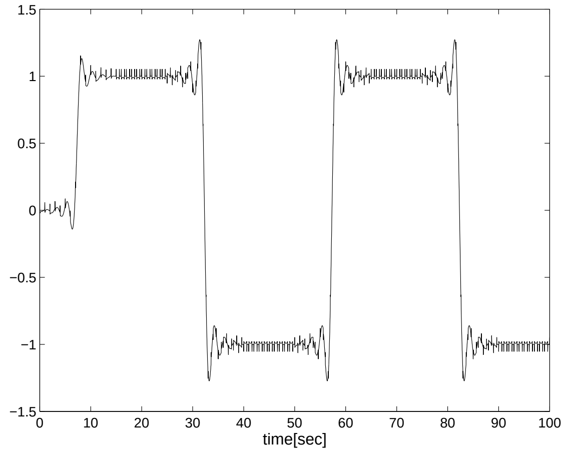

Figure 7: Time response against a rectangular wave (equiripple filter).

this paper deals with only the H^∞ specification, the proposed method can deal with other specifications introduced in [1] (e.g., the H^2 specification). Also, we can immediately extend the proposed method to design problems of decimators and sample-rate converters [6]. In this way, the proposed method is effective for practical sampled-data design in multirate signal processing.

Acknowledgments

The authors would like to thank Professor Stephan Boyd for his valuable suggestions.

References

[1] S. P. Boyd and C. H. Barratt, *Linear Controller Design, Limits of Performance*, Prentice Hall, 1991.

[2] N. A. Bruinsma and M. Steinbuch, "A fast algorithm to compute the H_∞-norm of a transfer function matrix," *Systems Control Lett.*, vol. 14, pp. 287–293, 1990.

[3] H. Ishii, Y. Yamamoto, "Sampled-data H^∞ and H^2/H^∞ design of multirate D/A converter," *Trans. ISCIE*, vol. 11, no. 10, pp. 585–592, 1998 (in Japanese).

[4] J. E. Kelley, "The cutting-plane method for solving convex programs," *J. Indust. Appl. Math.*, vol. 8, no. 4, pp. 703–712, 1960.

[5] P. P. Khargonekar and Y. Yamamoto, "Delayed signal reconstruction using sampled-data control," *Proc. 35th CDC*, pp. 1259–1263, 1996.

[6] M. Nagahara and Y. Yamamoto, "A new design for sample-rate converters," *Proc. 39th CDC*, pp. 4296–4301, 2000.

[7] M. Nagahara, Y. Yamamoto, "Sampled-data H^∞ design of interpolators," *Trans. ISCIE*, vol. 14, no. 10, 2001 (in Japanese).

[8] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice Hall, 1993.

[9] Y. Yamamoto, H. Fujioka, S. Hara, "Sampled-data control systems — VI, application to digital signal processing," *Systems, Control and Information*, vol. 44, no. 6, pp. 336–343, 2000 (in Japanese).

[10] Y. Yamamoto, A. G. Madievski and B. D. O. Anderson, "Approximation of frequency response for sampled-data control systems," *Automatica*, vol. 35, no. 4, pp. 729–734, 1999.

[11] Y. Yamamoto, M. Nagahara and H. Fujioka, "Multirate signal reconstruction and filter design via sampled-data H^∞ control," *Proc. MTNS 2000*, 2000.

[12] Y. Yamamoto and M. Nagahara, "Digital signal processing via sampled-data control theory," *Systems, Control and Information*, vol. 45, no. 4, pp. 162–167, 2001 (in Japanese).