やさしい解説

サブバンド信号処理*

——実時間動作化の奥の手——

牧野昭二 (NTT コミュニケーション科学基礎研究所)**

43.60.−c

1. はじめに

あっ，間に合わない。実時間動作

音響システムを実際に作って，実時間で動作させようとするとき，シミュレーションどおりには行かないことがある。サンプリング間隔の中に処理が納まり切れず，実時間動作が間に合わないのである。プロセッサのクロックを速くするという単純明快な方法のほかに，パイプライン処理・並列処理を導入する，サブバンド信号処理を使う，等といった演算を効率化する手法によって，実時間動作が可能になる。

サブバンド信号処理は帯域分割処理とも言われ，サンプリング周波数の変換を伴うため，マルチレート信号処理の代表例とも言える。サブバンド信号処理は，上述したように間に合わなかった実時間動作を可能にしたい場合や，ハードウェアを小型化したい場合，安価で遅い CPU を使って装置を経済化したい場合などにも用いられる。

サブバンド信号処理は，信号を幾つかの周波数帯域に分割し，それぞれの周波数帯域で独立に処理を行うため，各周波数帯域における信号やシステムの性質の違いを反映させ，帯域ごとに処理を変えることができる。また，各帯域の信号はフルバンドの信号に比べて周波数特性が平坦になる。つまり，信号が白色化される。これは収束速度が速くなる等，適応フィルタなどにとって大変にありがたいことである。

サブバンド信号処理は音声や音楽の符号化，エコーキャンセラを始め多くのアプリケーションに使われているが，ここでは，エコーキャンセラで使われているサブバンド処理を例にとってできるだけ具体的にやさしく解説する。

2. 最も簡単，帯域 2 分割

サブバンド信号処理で最も簡単なものは，帯域 2 分割である。間引きと補間が本質的に信号の変調操作であることを積極的に利用することによって，信号を二つの帯域に分割し，サンプリング周波数を 1/2 にする。それでは，さっそく具体的な例を見ていこう。

二つのサブバンド信号に分割

帯域 2 分割・合成過程の全体の流れを**図-1** に，スペクトルを**図-2** に示す。フルバンド信号の帯域を [0-8] kHz，サンプリング周波数を 16 kHz として，[0-4] kHz と [4-8] kHz の二つの帯域に分割してみよう。まず，4 kHz の理想ローパスフィルタで，[0-4] kHz 帯域の信号を切り出す [図-1(a) 上，図-2(a) 左]。[0-4] kHz 帯域の信号は，8 kHz サンプリングで表すことができる。そこで間引率 2 で間引きを行う。具体的には，データを 2 サンプルごとにサブサンプリングする。これで，[0-4] kHz 帯域の信号は 8 kHz サンプリングで表された [図-1(b) 上，図-2(b) 左，f_s=8 kHz になっていることに注意]。

同様に，4 kHz の理想ハイパスフィルタで，[4-8] kHz 帯域の信号を切り出す [図-1(a) 下，図-2(a) 右]。さて，ここからが問題である。サンプリング定理に従えば，[4-8] kHz 帯域の信号は，8 kHz サンプリングで表すことができるはずである。ここでは最も大胆に，そのまま間引率 2 で間引きを行ってみよう。すると [4-8] kHz 帯域の信号は 4 kHz で折り返されて (変調されて)，[0-4] kHz 帯域の信号となる。これで，[4-8] kHz 帯域の信号は 8 kHz サンプリングで表現された [図-1(b) 下，図-2(b) 右，f_s=8 kHz になっていることに注意]。

---

* A first step to the subband processing—For real-time implementation—.

** Shoji Makino (NTT Communication Science Laboratories, Kyoto, 619-0237) maki@cslab.kecl.ntt.co.jp

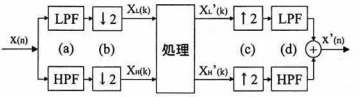


図-1 帯域 2 分割・合成過程の全体の流れ

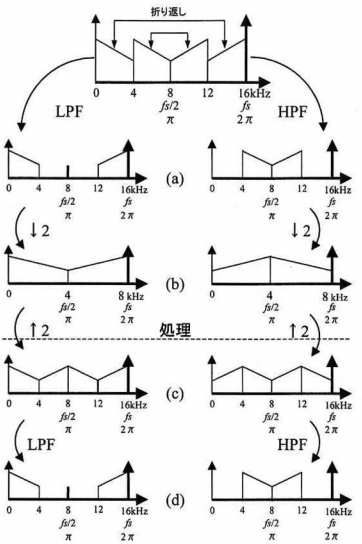


図-2 帯域 2 分割・合成過程のスペクトル

結局，[0-8] kHz 帯域の信号を 16 kHz サンプリングで表現していたものが，[0-4] kHz 帯域の信号を 8 kHz サンプリングで表現したもの二つに分割された。

メモリは同じ

簡単な例として，125 ms 長の FIR フィルタを実現することを考えてみよう。この FIR フィルタを 16 kHz サンプリングで表現すると 2,000 タップになるが，帯域を 2 分割すれば，8 kHz サンプリングで表現すれば良いので 1,000 タップで済む。低域と高域を合わせれば合計 2,000 タップになるので，メモリは減っていない。それでは，何がメリットなのだろうか？

サブバンドの利点，時間稼ぎ

FIR フィルタの畳み込みにはタップ数分の積和演算が必要である。しかも，次の入力信号が入って来る前に実行しなければ，実時間動作が間に合わない。16 kHz サンプリングであれば 62.5 μs 以内に 2,000 回の積和演算を終わらせることが，実時間動作のためには必要なのである。

これに対して，帯域を 2 分割すると，8 kHz サンプリングで表現すれば良いので，125 μs 以内に 1,000 回の積和演算を 2 回（合計 2,000 回）終わらせれば，実時間動作が可能となる。フルバンドに比べて時間の余裕が 2 倍（125 μs/62.5 μs）になっていることに気付いて欲しい。これは，単位時間当りの演算量が 1/2 になった，遅い CPU を使っても実時間動作が可能になった，ハードウェアを小型化できる，等と言い替えることができる。

フルバンド信号に戻すには

処理が終わったら補間率 2 で 0 補間して 16 kHz サンプリングに戻す。具体的には，値 0 のサンプルを挿入して 2 倍にアップサンプリングする。すると，[0-4] kHz 帯域は [4-8] kHz 帯域に折り返る［図-1(c) 上下，図-2(c) 左右］。そこで，[0-4] kHz 帯域を 4 kHz の理想ローパスフィルタで切り出し［図-1(d) 上，図-2(d) 左］，[4-8] kHz 帯域を 4 kHz の理想ハイパスフィルタで切り出し［図-1(d) 下，図-2(d) 右］，加え合せればフルバンド信号に戻る。

理想フィルタは実現できないから

ここでは，簡単のため理想フィルタを用いて話を進めてきたが，理想フィルタは実際には実現できない。そこで，実際には折返し歪が許容できる程度にローパスフィルタのカットオフ周波数を 4 kHz より幾分低く設定し，傾きもできるだけ急峻にする。ハイパスフィルタのカットオフ周波数も 4 kHz より幾分高く設定し，傾きもできるだけ急峻にする。4 kHz を挟んで信号の存在しない帯域ができてしまうのが欠点であるが，ローパス/ハイパスフィルタの設計次第で聴感上さほど問題のないサブバンドシステムが実現できる[1),2)]。

なお，帯域分割フィルタと帯域合成フィルタが対になり，低域に生じる高域からの折返しと，高域に生じる低域からの折返しが対称で，合成時に

補い合うQMF（Quadrature Mirror Filter）フィルタバンクという便利なものもあるが，エコーキャンセラのように外部からの信号の影響がある場合には，折返し歪が完全に打ち消し合わなくなるため，使用できない。

3. 更に多く帯域32分割

ポリフェーズフィルタバンクを利用

帯域2分割ができたので，今度はもっと多くの帯域に分けてみよう。更に多くの帯域に分割する例として，帯域32分割について考える。もちろん前節の方法をTree状に使って実現することも可能だが，分割数が多いときには特性と演算量の両面からあまりお薦めできない。その代わり，ポリフェーズフィルタバンク[3]というもっと良い方法がある。

ここで，お詫びと共に，頭の切り換えをお願いしたい。前節と本節では，残念ながら帯域分割数の定義が異なるのである。前節では，帯域 $[0, \pi]$ を2分割し，これを帯域2分割と呼んだ。これは［0-8］kHz帯域の実数信号を，［0-4］kHz帯域と［4-8］kHz帯域の二つの実数信号に分割することを考えれば納得できる。これに対して本節では，帯域 $[-\pi, \pi]$ あるいは帯域 $[0, 2\pi]$ を32分割し，これを帯域32分割と呼ぶ。これは［0-8］kHz帯域の実数信号（実は，［8-16］kHz帯域の複素共役信号を含む）を，帯域幅500 Hzの2実数信号，30複素数信号に分割するもので，短時間スペクトルに対応し，FFTを用いて効率的に実行できる。そのため，FFTのポイント数をもって帯域分割数と広く呼ばれているのである。読者の皆様には，全く違う手法であると思ってご容赦いただきたい。それでは，具体的な例を見ていこう。

32のサブバンド信号に分割

ポリフェーズフィルタバンクによる帯域分割・合成過程の全体の流れを**図-3**に，スペクトルを**図-4**に示す。ポリフェーズフィルタバンクを用いる方法では，まず，サブバンド m の信号を，$W_N[=\exp(j2\pi/N)]$（本節の例では $N=32$）を用いて W_N^{nm} で変調することにより，ベースバンドに周波数シフトする。次に，帯域 $[-\pi/32, \pi/32]$ のローパスフィルタ $f(n)$（すべての帯域で共通なため，プロトタイプフィルタと呼ばれる）

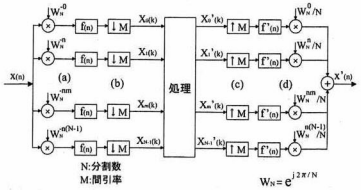


図-3　ポリフェーズフィルタバンクによる帯域分割・合成過程の全体の流れ

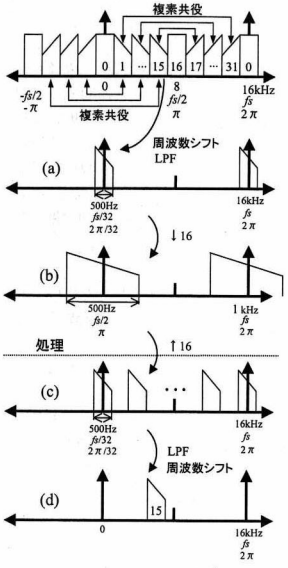


図-4　帯域32分割・合成過程のスペクトル

により帯域制限する［図-3(a)，4(a)］。更に，折返し歪の影響を避けるために帯域分割数32より小さな数で間引きを行う。ここでは間引率16で間引く（データを16サンプルごとにサブサンプリングする）。これにより，サンプリング周波数を1/16の1 kHzに下げる。これは信号の帯域幅（250 Hz実数，500 Hz複素数）に対して2倍のオーバサンプリングに相当していることに気付

いて欲しい。これにより理想フィルタを持ち出すことなく，折返し歪を十分に低減できる。以上により，$x(n)$は32個のサブバンド信号$X_0(k)$～$X_{31}(k)$に分割される［図-3(b)，4(b)，f_s=1 kHzになっていることに注意］。サブバンド信号$X_0(k)$～$X_{31}(k)$は短時間スペクトルに対応し，32サブバンド信号のうち0と16が実数，残りは複素数になる。サブバンド16に対して対称なもの（例えば15と17，等）は複素共役の関係にあり，全部で17（2実数，15複素数）のサブバンド信号があれば，全帯域信号を合成することができる。

フルバンド信号に戻す

各サブバンドで処理された信号$X_0'(k)$～$X_{31}'(k)$を，補間率16で0補間する（値0のサンプルを挿入して16倍にアップサンプリングする）ことによりもとのサンプリング周波数に戻すと，たくさんの折返し信号が現れる［図-3(c)，4(c)］。間引率16を帯域分割数32より小さく設定したため，折返し信号間に隙間が得られていることに注意して欲しい。そこで，この隙間を利用して帯域$[-\pi/16, \pi/16]$のローパスフィルタ$f'(n)$により帯域制限し，W_N^{nm}を用いて変調することにより，元の周波数まで周波数シフトする［図-3(d)，4(d)］。

最後に，各サブバンドの信号を加算することにより，全周波数帯域の信号を合成する。以上のように，間引率16を帯域分割数32より小さく設定することにより，帯域間のバンドギャップなく，ローパスフィルタ$f(n)$，$f'(n)$のタップ数に応じて，折返し歪を含めて−60～−40 dB程度の精度で帯域分割・合成が実現できる[4]。この帯域分割・合成過程は，32点FFTを用いて効率的に実行できる[5]。

メモリは減らない

前節と同様，125 ms長のFIRフィルタを実現することを考えてみよう。このFIRフィルタを16 kHzサンプリングで表現すると2,000タップになるが，帯域を32分割し，間引率16で間引けば，各帯域のサンプリング間隔は16倍に広がり，1 kHzサンプリングとなる。しかも，各帯域のFIRフィルタのタップ数は，1/16に間引かれて125タップで済む。このようなFIRフィルタが全部で17帯域（2実数，15複素数）あり，これらを合わせれば合計で実数250タップ，複素数1,875タップになる。複素数は2倍のメモリが必要であると数えると，全体で4,000のメモリが必要となる。2倍のオーバサンプリングのため，メモリは2倍（4,000/2,000）に増えている。

帯域32分割，間引率16の演算量は1/4

帯域分割・合成に要する演算量は，FIRフィルタの演算量に比べて少ないため無視できる。フルバンド構成のとき，16 kHzサンプリングであれば62.5 μs以内に2,000回（250 μsで8,000回に相当）の積和演算を終わらせることが必要であった。これに対して，帯域を32分割すると，2倍のオーバサンプリングをしても，各帯域のサンプリング間隔は間引きによって16倍の1 kHzサンプリングに広がり，各帯域のFIRフィルタのタップ数は1/16の125タップに間引ける。従って，各帯域における演算量は，フルバンド構成に比べて$1/16^2$になる。全体では，1 ms以内に125回の積和演算を2実数帯域，15複素数帯域（複素数は4倍の積和演算が必要であると数えると，合計約8,000回）終わらせれば，実時間動作が可能となる。フルバンドに比べて時間の余裕が4倍（1 ms/250 μs）になっている。サブバンド信号処理の以上の利点を生かせば，高速RLSアルゴリズムの実時間動作も可能である[6]。

4. 実数で演算するならSSBサブバンド

前節のポリフェーズフィルタバンクを使ったサブバンドでは，各帯域の信号は複素数になった。これを実数信号に変換する方法として，Single Side Band（SSB）変調法が知られている[7]。

複素数のサブバンド信号を実数化

前節のポリフェーズフィルタバンク図-4(a)で得られた信号から始める［**図-5**(a) 右］。まず，この信号を帯域幅分右へ周波数シフトする［図-5(b) 右］。同様に，図-5(a) 右と複素共役な信号［図-5(a) 左］を帯域幅分左へ周波数シフトする［図-5(b) 左］。これら複素共役な2つの信号を加え合わせると左右対称な信号，つまり実数信号が得られる［図-5(c)］。次に，折返し歪の影響を避けるために帯域分割数の半分16より小さな数で間引きを行う。ここでは間引率8で間引く（データを8サンプルごとにサブサンプリングする）。これにより，サンプリング周波数を1/8の

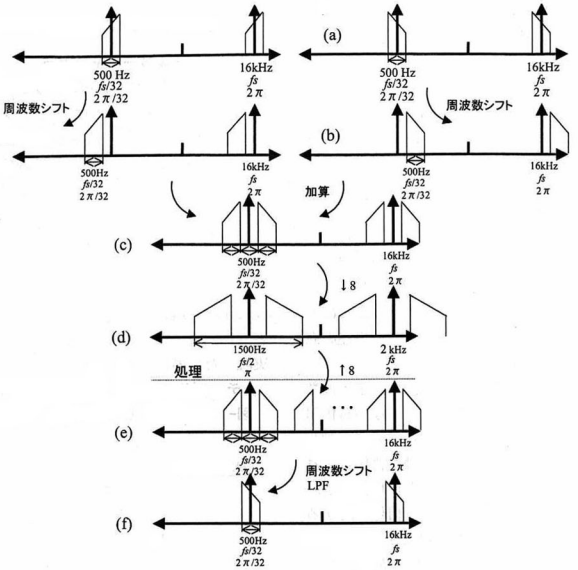


図-5 SSB変調による実数化

2 kHzに下げる［図-5(d)，f_s=2 kHzになっていることに注意］。これは信号の帯域幅（500 Hz実数）に対して2倍のオーバサンプリングに相当していることに気付いて欲しい。これにより理想フィルタを持ち出すことなく，折返し歪を十分に低減できる。

以上により，0と16が帯域幅250 Hzの実数信号，1から15は帯域幅500 Hzの実数信号になる。全部で17実数のサブバンド信号があれば，全帯域信号を合成することができる。

実数のサブバンド信号を複素数に戻す

各サブバンドで処理された信号を，補間率8で0補間する（値0のサンプルを挿入して8倍にアップサンプリングする）ことによりもとのサンプリング周波数に戻すと，たくさんの折返し信号が現れる［図-5(e)］。間引率8を帯域分割数の半分16より小さく設定したため，折返し信号間に隙間が得られていることに注意して欲しい。そこで，帯域幅分左へ周波数シフトすることによりベースバンド信号に戻し，この隙間を利用して帯域$[-\pi/16, \pi/16]$のローパスフィルタ$f(n)$により帯域制限する［図-5(f)］。更にW_N^{nm}を用いて変調し，元の周波数まで周波数シフトすることにより図-4(d)が得られる。

なお，次のように前節のポリフェーズサブバンドと本節のSSBサブバンドの演算量は等しい。ポリフェーズサブバンドでは大部分の信号が複素数になるため，実数での演算に比べて4倍の演算量がかかる。これに対して，SSBサブバンドでは，間引率が1/2になるため，演算量が$4(=2^2)$倍かかる。これらが相殺し合うためである。

5. サブバンド信号処理の利点と欠点

サブバンドへの分割・合成方法が分かったので，ここで改めてサブバンド信号処理の利点と欠点をまとめてみよう。

利点は多数

サブバンド信号処理の第1の利点は，これまで

説明してきたように，間引きによってサンプリング間隔が広がり，各帯域のタップ数が減ることにより，演算が効率化されることにある。この効果は，演算量がタップ数に比例する場合（FIR フィルタ，LMS 適応フィルタ，等）はもとより，演算量がタップ数の 2 乗に比例する場合（逆フィルタ処理，RLS 適応フィルタ，等）には極めて顕著である。これにより，ハードウェアによる実時間動作化が促進できる。

サブバンド信号処理の第 2 の利点は，時間-周波数にわたる解析が行えるため，各周波数帯域における信号やシステムの性質の違いを反映させ，帯域ごとに処理を変えることができることにある。これにより，更なる処理の効率化や性能の向上が図れる。

サブバンド信号処理の第 3 の利点は，全周波数帯域でみるとスペクトル形状が平坦でない有色信号を，スペクトルが平坦に近い帯域信号に分割して処理していることにある。これは，言い替えれば，入力信号を白色化していることに相当する。このため，音声のように相関のある信号に対する適応アルゴリズムの収束速度を向上させることができる[8),9)]。また，エコーキャンセラをステレオ化する際にも有効である[10)]。

第 4 の利点は，各帯域の信号は間引きによって $[0, 2\pi]$ に拡大されることにある。これにより，極や零点を取り扱う場合に，それらの分解能が上がり処理し易くなる[11)]。

第 5 の利点は，他の手法（センタクリッパ，音声スイッチ，等）を簡単に埋め込むことができ，装置全体の性能向上を図れることにある。

第 6 の利点は，各帯域から複数の制御信号が得られるため，ダブルトーク制御等の様々な制御が高度化できることにある。

欠点は遅延とメモリ増加

これらに対して，サブバンド信号処理の最大の欠点は遅延である。帯域 32 分割，間引率 16 の場合 20〜40 ms の遅延が生じる。これらは主にローパスフィルタ処理において発生し，分割・合成の精度とトレードオフの関係にある。遅延を減らす検討やなくす検討[12)]が盛んに行われているが，根本的な解決には至っていない。また，ローパスフィルタや FIR/適応フィルタのメモリ増加も欠点である。

6. サブバンド信号処理の適用例

サブバンドエコーキャンセラ

サブバンドエコーキャンセラの構成の一例を図-6 に示す。受話入力 $x(n)$ とエコー信号 $y(n)$ は，帯域分割フィルタにより N 個のサブバンドに分割される。各サブバンドには適応フィルタがあり，それぞれ独立にエコーを消去する。最後に，帯域合成フィルタにより，各サブバンドの誤差信号から全周波数帯域の誤差信号 $e(n)$ を合成する。

サブバンドエコーキャンセラは，フルバンド構成に比べて演算が効率的であるため，次に述べるように，大規模なエコーキャンセラを 1 チップで実現したい場合に適している。

1 チップサブバンドエコーキャンセラ

図-7 は，NTT と AT & T が共同開発した商

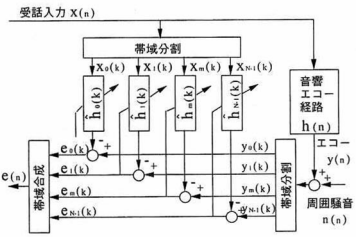


図-6 サブバンドエコーキャンセラ

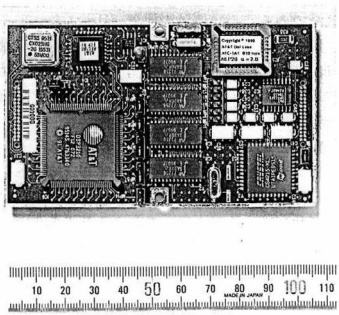


図-7 1 チップサブバンドエコーキャンセラ

用の1チップサブバンドエコーキャンセラであり，3.4 kHz 帯域でエコー消去時間長 250 ms，7 kHz 帯域でエコー消去時間長 150 ms を実現している。各帯域に必要なタップ数を聴覚特性を考慮して最適配置し，各帯域の指数減衰特性を反映させて収束速度を高速化し[13]，ダブルトーク制御用に専用の帯域を使用してダブルトーク時の安定動作を確保するなど，複数の帯域の特性の違いを様々な形で処理に反映させることにより，性能の向上を図っている。

7. おわりに

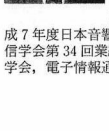
サブバンド信号処理は実時間動作を可能にするための，重要な実現手段であり，高性能で小型経済的なハードウェアを実現するための奥の手である。サブバンド信号処理は周波数変調，サンプリング周波数の変換，FFT の活用など，ディジタル信号処理技術の宝庫であり，今後とも発展し広く応用されることを期待する。

謝　辞

最後に，サブバンド信号処理にともに取り組み，理論と実践に関して日々討論を重ね，本稿に対しても数々の貴重なご意見をいただいた NTT 羽田陽一博士と中川朗氏に謝意を表す。

文　献

1) H. Oikawa, S. Minami and T. Saeki, "A new echo canceller realized by high performance digital signal processor," Proc. ISCAS 88, 1329-1332 (1988).
2) H. Yasukawa, I. Furukawa and Y. Ishiyama, "Acoustic echo control for high quality audio teleconferencing," Proc. ICASSP 89, 2041-2044 (1989).
3) P.P. Vaidyanathan, "Multirate digital filters, filter banks, polyphase networks, and applications: a tutorial," Proc. IEEE **78**, 56-93 (1990).
4) A. Nakagawa, Y. Haneda and S. Makino, "Subband acoustic echo canceller using two different analysis filters and 8 th order projection algorithm," Proc. IWAENC 97, 140-143 (1987).
5) M. Portnoff, "Implementation of the digital phase vocoder using the fast Fourier transform," IEEE Trans. Acoust. Speech Signal Process. **ASSP-24**, 243-248 (1976).
6) B. Hatty, "Recursive least squares algorithms using multirate systems for cancellation of acoustic echoes," Proc. ICASSP 90, 1145-1148 (1990).
7) R. Crochiere and L. Rabiner, *Multirate Digital Signal Processing* (Prentice-Hall, Englewood Cliffs, NJ, 1983).
8) W. Kellermann, "Analysis and design of multirate systems for cancellation of acoustical echoes," Proc. ICASSP 88, 2570-2573 (1988).
9) S. Makino, J. Noebauer, Y. Haneda and A. Nakagawa, "SSB subband echo canceller using low-order projection algorithm," Proc. ICASSP 96, 945-948 (1996).
10) S. Makino, K. Strauss, S. Shimauchi, Y. Haneda and A. Nakagawa, "Subband stereo echo canceller using the projection algorithm with fast convergence to the true echo path," Proc. ICASSP 97, 299-302 (1997).
11) H. Yamada, H. Wang and F. Itakura, "Recovering of broad band reverberant speech signal by sub-band MINT method," Proc. ICASSP 91, 969-972 (1991).
12) D. Morgan and J. Thi, "A delayless subband adaptive filter architecture," IEEE Trans. Signal Process. **43**, 1819-1830 (1995).
13) 牧野昭二，羽田陽一，"周波数帯域における音響エコー経路の変動特性を反映させたサブバンド ES アルゴリズム，" 信学論 **J79-A**, 1138-1146 (1996).



牧野　昭二

昭 54 東北大・工・機械卒。昭 56 同大大学院修士課程了。同年日本電信電話公社（現 NTT）入社。以来，NTT 研究所において，電気音響変換器，拡声電話機，及び，音響エコーキャンセラなどの音響信号処理の研究に従事。現在，NTT コミュニケーション科学基礎研究所知能情報研究部主幹研究員。工博。平成 7 年度日本音響学会技術開発賞。平成 9 年度電子情報通信学会第 34 回業績賞。IEEE Senior Member。日本音響学会，電子情報通信学会各会員。