【解説】

インターネットのための音声処理技術

大室　仲　　金子孝夫
NTTサイバースペース研究所

ここ数年の間に，インターネット環境が飛躍的に発展した．かつて，インターネットはテキストの電子メールをやりとりする目的がほとんどであったが，近年では，インターネット上で音声や音楽が扱われることはごくあたりまえの時代になった．その要因としては，モデムの性能向上やISDNの普及による回線の高速化や，パーソナルコンピュータの処理能力の大幅な向上によって専用のハードウェアなしに複雑な信号処理が可能になったことなどが挙げられる．インターネット上での音声アプリケーションは，ダウンロード（蓄積）タイプ，実時間配信（放送）タイプ，実時間通信（電話）タイプに大きく分けられるが，それぞれのタイプや使用する環境によって必要とされる音声処理技術もさまざまである．本稿では音声符号化技術を中心に，音声処理技術の基本と最近の動向について述べる．

■音声のデータ形式

音声データの基本的な形式は，PCM（Pulse Code Modulation）である．PCMはコンパクトディスク（CD）や衛星放送のBモードなどに用いられる一般的なデジタルフォーマットで，サンプリング周波数と量子化ビット数によって規定される．サンプリング周波数はCDが44.1kHzでBモードが48kHzである．量子化ビット数はどちらも16ビットである．量子化ビット数が16ビットあればダイナミックレンジが90dB以上あり，一般に人間の耳では劣化を検知できないとされている．16ビットで量子化されたPCMフォーマット（16ビットPCM）を音声符号化の世界では「原音」として扱う（厳密にいえばPCMもアナログ信号を符号化する方法の1つである）．16ビットPCMの1サンプル値をshort intで表現して時刻順に書き込んだものがRAW形式であるが，コンピュータのアーキテクチャ（ビッグエンディアンかリトルエンディアンか）によってバイト順序が異なることに注意が必要である．WindowsのWAVファイルやMacintoshのAIFF形式は，RAW形式のバイト順序を定義するとともに，サンプリング周波数や時間長などの補助情報を付加したファイルフォーマットである．

16ビットPCMのほか8ビットの（直線）PCMも存在するが，最近ではあまり用いられない．かつてパソコンのCPUでは複雑な信号処理ができなかったり，8ビットのAD/DA変換チップが主流であった時代には，ファイルサイズの節約も兼ねて8ビットPCMが用いられることもあった．しかし，8ビットPCMは量子化レベルが高々256レベルであり，十分なSN比が得られないため，最近では高度な音声符号化技術による圧縮が主流となっている．ときどき，「○×方式の音声符号化の入力として，16ビットPCMの代わりに8ビットPCMを入力すると，出力ファイルのサイズは半分になりますか？」という質問をいただくことがある．ここがzip等によるユニバーサルなデータ圧縮と音声符号化との根本的に異なる点で，データ圧縮の場合は入力ファイルサイズが小さければ出力ファイルサイズも小さくなる．一方，音声符号化では，後述するロスレス符号化は別として，入力PCM信号の量子化ビット数は圧縮後のサイズには基本的に関係がない．

本稿では対象を音声信号に限定し，音楽に関しては別の記事に任せるが，音声と音楽の違いは，必要とされるサンプリング周波数と信号に含まれる冗長性の違いである（音声でもコーラスは冗長性の点から音楽に含める）．サンプリング周波数は「標本化定理」によって必要な周波数帯域の2倍のサンプリングレートで標本化すればよいことが決まっている．音声に含まれる信号の周波数帯域はせいぜい7kHzまでであり，サンプリング周波数は16kHzもあれば十分である．サンプリングレートが高いほど情報量（と処理量）が増えるため，アプリケーションごとに必要とされる周波数帯域に応じたサンプリング周波数が使われる．サンプリング周波数はオーディオデバイスのハードウェア仕様にもかかわるが，一般には，用件を伝えることが目的の電話用途には8kHz（3.4kHz帯域），声の質など豊かな情報伝達が必要な場合には11.025kHz（5kHz帯域），テレビ会議など拡声系で臨場感を必要とする用途には16kHz（7kHz帯域）が用いられる．なお，サンプリングレートは容易に変換できるため，ハードウェ

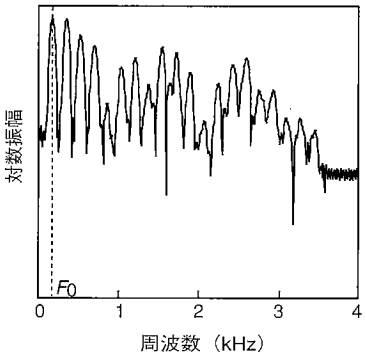

図-1　3.4kHz帯域音声信号のスペクトル

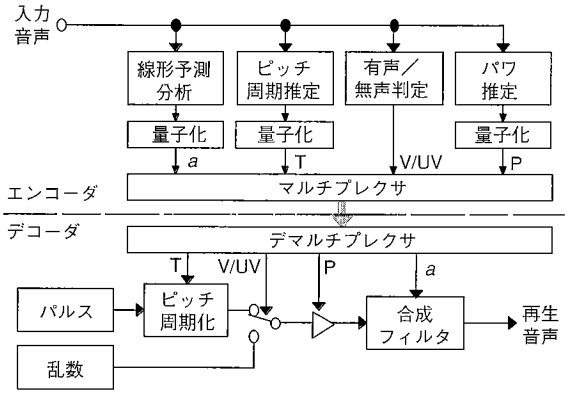

図-2　ボコーダ

アの制約がすべてではない．整数比（例8kHzと16kHz）のアップサンプリングやダウンサンプリングは，低域通過フィルタ（LPF）をソフトウェア的に実現するだけで簡単に変換できる．8kHzと11.025kHzのような整数比でない場合は少々複雑になるが，LPFとサンプリング定理の式を組み合わせれば容易に変換式を導出できる．CPUの処理能力が低かった時代には，サンプリングレートの変換を実時間で高品質に実行することは難しかったが，最近では1割程度のCPU占有率で，十分に高品質なサンプリングレートの変換ができるようになった．音声符号化方式の中には，符号化アルゴリズム自身にサンプリング変換を含んでいるものがある．見かけ上の入出力サンプリング周波数はたとえば11kHzであるが，実は内部で8kHzにダウンサンプリングされてから符号化されている場合がある．音声符号化においてサンプリングレートは重要なファクタであるが，ユーザ（ハードウェア）側からはブラックボックス化することも可能である．

音声符号化に求められる条件

音声符号化に求められる要求条件は，アプリケーションの利用形態による．一般に音声符号化のスペックとしては以下の項目が挙げられる．これらすべてに優れているに越したことはないが，アプリケーションごとの重要な項目に特に注目し，重要でない項目はある程度まで許容することで，用途に合った符号化方式を設計することができる．

(1) ビットレート
(2) 再生周波数帯域
(3) 人間の音声か非音声（音楽等）か
(4) 再生音の品質（歪み）
(5) エンコード，デコードのための処理量
(6) バッファリングに伴う遅延

まず，ビットレートと再生音の品質は直接的な関係にある．16ビットPCMを「原音」とすると，8kHzサンプリングの場合でビットレートは128kbit/sとなる（11.025kHzの場合は176.4kbit/s）．この「原音」のビットレートをいくらまで下げるかは，回線速度と利用目的による．インターネット（イントラネット）電話の場合は，End-to-Endのスループット以下であればよい．一方，ダウンロードタイプではビットレートが低いほど高速にダウンロードできるし，逆に時間をかければいくらでも高品質化が可能なため，再生音品質の最低ラインを決めて，その中でビットレートの低いものを選択することになる．次に対象とする入力信号の種類と周波数帯域を決める．会話音声なのか音楽や環境音なのかは特に重要で，送りたい情報が会話音声には音声用の，音楽には音楽用のアプローチがある（BGMや背景音のある音声では，背景音の重要度による）．遅延と処理量に関しては，ダウンロードタイプや放送タイプの場合，エンコードにかかる処理量やバッファリングに伴う遅延はあまり重要ではない．エンコードはセンタ側でデータをアップロードする際の1回限りであるし，片方向なので遅延は許容される．インターネット電話のように双方向で実時間通信の場合には，エンコードの処理量は特に重要であるし，遅延が大きくなると話しづらいという問題が生じる．

音声符号化の原理

音声符号化とは，音声信号の「冗長性」を利用して，できるだけ音質を低下させることなく低いビットレートで音声信号を表現することである．音声符号化は，もともと通信回線を有効に（低コストで）利用するために研究されてきた．現在では携帯電話をはじめとする無線通信の用途か，専用線を利用した電話多重化装置としての利用が多い．一般に，情報圧縮にはzip等に代表されるロスレス符号化と，ある程度の劣化を許す非可逆符号化がある．音声符号化にも主にデータ圧縮手法に基づくロスレ

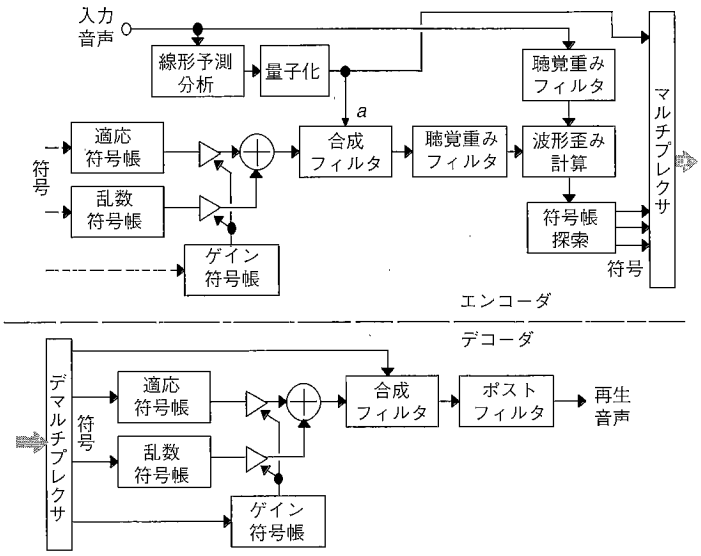

図-3　CELP系符号化

ス符号化と信号処理手法に基づく非可逆符号化があるが，現在主流の符号化方式はほとんどが非可逆方式である．その理由は，ロスレス方式ではあまり圧縮率を上げることができないことと，音声符号化の最終的な利用形態はスピーカまたはヘッドセットから再生して人間が聞くことであり，聴覚的に問題がなければ完全な可逆性は必要とされないためである（ただし，圧縮と伸長を繰り返す「タンデム接続」をすると，本来の性能よりも品質が低下する）．

音声符号化手法は，大きく分けて，波形を忠実に再現することを目的とする波形符号化と，音声の生成モデルに基づいて生成モデルのパラメータから音声を合成するボコーダと呼ばれる方法がある．また，量子化の方法によってスカラ量子化とベクトル量子化に属するものがあるが，最近の符号化方式は量子化効率を上げるために，ほとんどがベクトル量子化を利用している．波形符号化は対数PCMやADPCMに代表される方式で，入力信号が音声であるか否かにかかわらず，すべての信号に（音楽にも）適用できるというメリットがある．反面，あまり圧縮率を上げることができず，ビットレートを下げるとノイズっぽくなるというデメリットがある．ITU-T G.726（ADPCM）が32kbit/sであるように，圧縮率はせいぜい「原音」の2～4分の1である．一方，ボコーダは音声に特化した方式であるため音声以外にはまったく適用できないが，1～2kbit/s（「原音」の64～128分の1）といった極端に低いビットレートを実現できる．

図-1に8kHzサンプリング（3.4kHz帯域）音声の母音区間におけるスペクトルの例を示す．音声のスペクトルは，基本周波数（ピッチ，F_0）とその高調波からなる調波構造のスペクトルを，ゆるやかなスペクトル包絡で変調した形をしている．したがって，スペクトル包絡の形状と基本周波数，パワが分かれば，音声信号に含まれる情

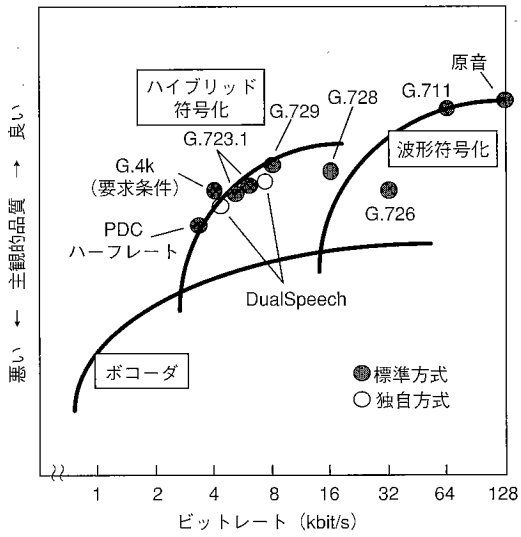

図-4　音声入力時のビットレートと品質

報の大半を表現することができる．ボコーダでは入力信号を分析してこれらのパラメータの値を計算し，受信側でパラメータをもとに音声を再合成する．**図-2**にボコーダのブロック図を示す．ボコーダにおける合成フィルタの駆動音源として，母音区間ではピッチ周期で並べられたパルス列が，子音区間では乱数系列が用いられる．図-1に対応させると，ピッチ周期（$1/F_0$）で並べられたパルスは，F_0の整数倍の高調波を持つ調波構造スペクトルに相当する．一方，子音区間は声帯振動を伴わない白色の駆動音源に相当する．しかし，実際の音声はボコーダで表現されるような単純構造の信号だけではないため，ボコーダによる再生音は「機械的」「ロボットのようだ」といわれる（ロボットや機械にもともと声があるわけはないので，昔のロボットがボコーダを使っていて，自然性が十

	G.728	G.729	G.723.1		PDC	DualSpeech	
ビットレート（kbit/s）	16.0	8.0	6.3	5.3	3.45	7.4	4.6
フレーム長（ms）	0.625	10	30		40	20	
サブフレーム長（ms）	0.625	5	7.5		10	5	10
線形予測係数（kbit/s）	0.0	1.8	0.8		0.775	1.3	1.1
適応符号帳（kbit/s）	0.0	1.3	0.6		0.8	1.3	0.8
乱数符号帳（kbit/s）	11.2	3.4	3.3	2.266	1.0	3.4	2.0
ゲイン，その他（kbit/s）	4.8	1.5	1.6		0.875	1.4	0.7

表-1　各符号化方式の仕様

分でなかったというのが正しい）．

これに対して最近の主流は，波形符号化とボコーダのそれぞれの長所を生かした「ハイブリッド符号化」に分類されるものが多い．中でも「CELP（Code Excited Linear Prediction）系」と呼ばれる方式が多い．**図-3**に典型的なCELP系符号化方式のブロック図を示す．図-2と図-3を比較して分かるように，ボコーダとCELP系符号化の構造はよく似ている．異なる点を挙げると，CELP系符号化ではピッチ周期性を再現するための適応符号帳（櫛形フィルタに似た構造）から出力される時系列ベクトルと，ランダム性を表現するための乱数符号帳から出力される時系列ベクトルの加重和で駆動音源を表現しており，調波構造しか表現できないボコーダに比べるとより多様な駆動音源を表現できるモデルとなっている．駆動音源の決定（探索）方法の点では，ボコーダは入力音声を分析して各パラメータを計算するシンプルな構造であるのに対して，CELP系エンコーダは内部に「ローカルデコーダ」を持ち，再生音をいったん生成したうえで波形歪みの小さいものを選択している（実際のインプリメントではそれと等価な処理に置き換えることが多い）．このようなボコーダの決め方をオープンループ探索，CELP系の決め方をクローズドループ探索と呼ぶ．ただし，完全なクローズドループ探索には膨大な処理量がかかるため，実際にはCELP系符号化でもオープンループ探索とクローズドループ探索を併用している．

■さまざまな符号化方式

CELP系符号化が「ハイブリッド符号化」と呼ばれる所以は，「ボコーダ＋αの構造で駆動音源を多数生成し，その中から波形符号化の尺度で最適なものを選択している」点といえるが，ビットレートが高く波形符号化に近いものほど＋αの占める割合が多く，ビットレートが低くてボコーダに近くなるほど＋αが少なくなる．**図-4**に，波形符号化，ボコーダ，ハイブリッド符号化を用いたときのビットレートと主観品質の関係を模式的に示した．また，代表的な符号化方式を対応する位置にプロットした．G.4kは現在進められている標準化作業の要求条件である．なお，各方式の品質はおよその相対的関係を示したもので，厳密な評価値を表すほどのものではない．図中のG.728からPDCハーフレートまでの各方式（G.4kは未定）は，CELP系のハイブリッド符号化に属する．

各方式の違いを**表-1**にまとめた．大雑把にいえば，フレーム長，乱数符号帳の構成，各符号帳に割り当てるビット数，符号帳探索手法の違いであるが，これらの違いがさまざまな特徴（設計思想）に反映する（もちろんこのほかにもそれぞれにさまざまな工夫がなされている）．フレーム長とは音声を入出力（送受信）する最小の時間単位であり，サブフレームとはフレームの中でさらに短い時間に分割したベクトル量子化の処理単位である．フレーム長はその符号化方式が必要とする最小限のバッファリングであり，フレームの長短は遅延の長短に関係する．一般にビットレートが低いほどフレームやサブフレームが長い．また乱数符号帳に割り当てられるビットが相対的に少なく，適応符号帳への依存が高い．これはビットレートの低いものほど「＋α」が少なくなり，ボコーダに近い性質を示すことに対応している．一方，G.728（LD-CELP）は線形予測や適応符号帳という手法を利用しているが，それらにビット割り当てをせずに「バックワード予測」と呼ばれる手法を使っている．またフレーム長が0.625ms（5サンプル）ときわめて短く，CELP系のハイブリッド符号化の中でも限りなく波形符号化に近いといえる．

図-5，**図-6**，**図-7**にPDCハーフレート（PSI-CELP），G.729（CS-ACELP），DualSpeech[1]における乱数符号帳の構造を示す．乱数符号帳の構成は，いかに低ビットで自然性を表現するかに関係している．いずれの構造にもピッチ周期化部があるが，母音性の強い区間ではピッチ周期化を強く，子音区間などではピッチ周期化を弱くかける（またはかけない）可変構造になっている．図-5のタイプはサブフレーム長の複数のベクトルから最適なものを1つ選択する（ないし複数選択してそれらのベクトル和で表す）構造で，PDCのほかG.728もこのタイプに属する．このタイプのメリットは，ベクトル形状を大量の音声データを使った「学習」と呼ばれる手法で設計することができるため，より低ビットレートで効率の高い符号化を実現できる点である．反面，ベクトルを記憶するための大きなメモリ領域が必要であったり，符号帳の探索のための処理量が多くなるというデメリットもある（もっとも，パソコンでの処理を前提とすれば，メモリ量は大

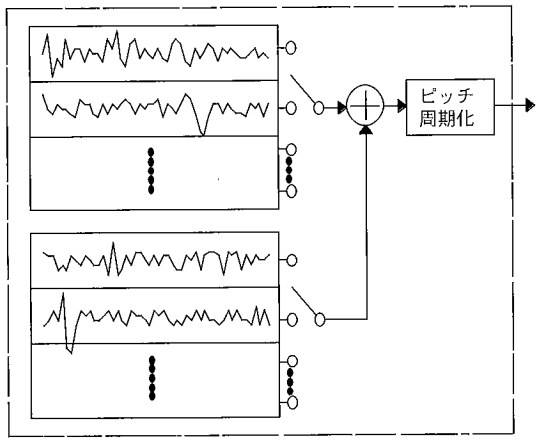

図-5　PSI-CELPの乱数符号帳

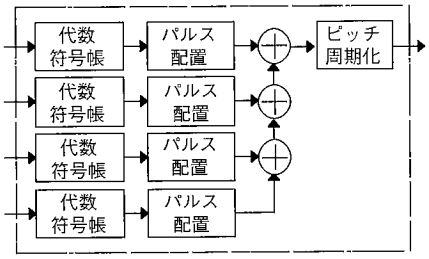

図-6　CS-ACELPの乱数符号帳

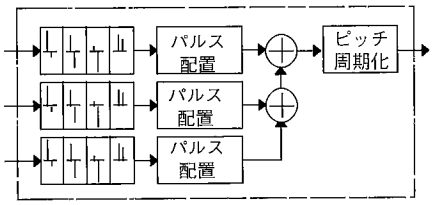

図-7　DualSpeechの乱数符号帳

きな問題ではない）．図-6のタイプはAlgebraic（代数）符号帳と呼ばれる．この構造の特徴は，駆動音源をベクトルとしてメモリに保持するのではなく，代数的なルールにしたがって振幅1のパルスを（この例では4本）サブフレーム内に配置してベクトルを生成するもので，パルスの位置情報と正負符号を送信する．G.729（CS-ACELP）のほか，G.723.1の5.3kモード（ACELP）もこれに属する．6.3kモード（MP-MLQ）はマルチパルスに分類されているが，振幅1のパルスを配置する点ではACELPに似ている．この手法のメリットは，構造が簡単で高速にパルス位置を探索できる点である．ただし，位置情報と符号を送ることはベクトル量子化の観点から効率はいまひとつで，どちらかといえば中ビットレートの方式に向いている．これらの特徴を踏まえ，DualSpeechでは図-7に示すような学習符号帳と代数符号帳の中間的な構造を採用している．基本的にはパルスを配置する手法をとっているけれども，振幅1のシングルパルスの代わりに隣接する2サンプルを1組とした「デュアルパルス」を定義し，このパルスの形状を音声データを用いた学習によって設計している．エンコーダではデュアルパルスの形状（コード）と配置する位置を選択して送る．DualSpeechの設計思想は高能率符号化と徹底した処理量の削減の両立を目指しており，表-1の中では最も演算量が少ない（処理量はチップのアーキテクチャやインプリメントの巧拙，固定小数点演算か浮動小数点演算かなどに依存するため単純に比較することは難しいが，もし同じ条件で比較すると仮定すれば，G.729とG.723.1がほぼ同じで公称20MIPS，PDCはその2倍弱，DualSpeechは半分程度と推定される．なお，G.729にはAnnex Aと呼ばれる低演算量版があり，Annex Aの公称演算量はG.729の半分の10MIPSとされている）．

背景雑音の問題

CELP系の音声符号化はビットレートが低くなるほど構造がボコーダに近くなるため，音声以外の信号が入力されたり，音声でも背景雑音が重畳している場合には再生音声の品質（自然性）が劣化する．図-4の各方式はビットレートの高い順に，G.711，G.726，G.728まではほとんど背景雑音の影響を受けないのに対して，G.729以下は（背景雑音のための特別な処理を併用しなければ）ビットレートが低くなるにつれて背景雑音の影響が徐々に大きくなる．電話用途で普通の（口元にマイクのある）ハンドセットが使用される場合には，背景雑音が問題となることは少ないが，インターネット用途では，オーディオ用のマイクロフォンを使って，ある程度の騒音下で集音されることが多い．ITU-TにおけるG.4k（4kbit/s）標準化の要求条件では，SN比が15dBの自動車雑音付加音声と同30dBのオフィス雑音付加音声を入力した場合に，G.729（8kbit/s）と同等の品質が求められており，音声符号化の利用形態の多様化に伴って，音声が圧縮できればよいという時代から背景雑音のある音声に対する性能が重視される時代なってきたといえる．

背景雑音の問題に対する代表的なアプローチは，ノイズリダクションの利用である．入力信号の段階でノイズリダクションをかければ，コーデックに影響を与えるノイズ成分は低減される．スペクトルサブトラクション法はノイズリダクション手法としてポピュラーな手法の1つである．原理的には雑音区間において推定した雑音スペクトルを，周波数領域において差し引く（対数領域で減算，線形領域では除算する）ものである．EVRC（IS-127）[2]では，入力音声を128点（48点はオーバーラップ）のFFTで周波数領域に展開し，16のバンドに分割して，バンドごとにノイズの推定から決定した-13～0dBのゲインを乗算する．これを再度128点の逆FFTで時間領域に戻して，48点ずつオーバーラップさせながら雑音抑圧した信

号を生成している．

ノイズリダクションはコーデックの相互接続性に影響しない簡便な手法として利用価値は高いが，背景雑音が完全になくなってしまうわけではないため，符号化による不快音をなくしてしまうことはできない．ノイズリダクションを強くかけすぎると，それだけで背景雑音の性質が不自然に変化することもある．また，背景音も「ユーザが伝達したい情報の一部」だとすれば，音声符号化の都合で一方的にカットしてしまうのも最善とはいえない．この問題を根本的に解決するには，符号化アルゴリズムそのものを音声＋雑音という形でモデル化して設計する必要がある．これまでに，入力音声をノイズリダクションの手法を使って雑音成分と音声成分に分離して別々に符号化して送る方法[3]や，背景雑音を短時間変動成分と長時間定常成分としてとらえ，デコーダ側で各背景音成分を推定して加算する方法[4]などが提案されており，G.4kの要求条件にかなり近づく試験結果が得られている．

■フレーム（パケット）ロスの問題

ほとんどの音声符号化方式では時間方向の「予測」手法が用いられているため，ビットエラーやフレームロスによって過去の情報（予測のもと）が失われると，予測処理そのものが正常に動作できなくなる．ダウンロードタイプの場合は一般にTCP/IPプロトコルで再送制御が可能なため，パケットロスの問題が生じることはない．一方，リアルタイム配信系やインターネット電話などのリアルタイム通信用途の場合には，TCPの代わりに再送制御を行わないUDP等のプロトコルが使われることが多く，回線が混雑するとパケットロスが発生する．ITU-T標準をはじめとする各符号化方式では，3％程度のフレームロスには耐えられる対策を講じている．具体的には，あるフレームの情報が届かない場合には，直前のコードと直前の（予測のための）内部状態を使って適当に連続する信号をデコーダ側で生成する．このとき「パシッ」といった極端なノイズが発生しないように，該当フレームとその周辺フレームでパワをうまくコントロールしている．1フレーム単位で情報が落ちる程度ならこの程度の処理で実用上大きな問題はない．しかし，G.729の1フレームは高々10バイト，DualSpeechでも19バイトであるため，インターネットで利用する場合には，オーバーヘッドを少なくするために，数フレームで1パケットを構成することが多い．この場合には，1パケットが抜け落ちると数フレームが一度に落ちることになり，コーデックの想定するフレームロスの許容範囲を大きく超えてしまう．これを防ぐためには，遅延が増えることを許容して図-8のようなインタリーブ手法を用いてパケットを構成するとよい．コーデックにはそれぞれフレームロスが生じたときに，正常な状態に復帰するための時間があり，1パケットが落ちても，実際のフレーム順序に戻したときに，この復帰時間だけ間があいた状態になることが望ましい．

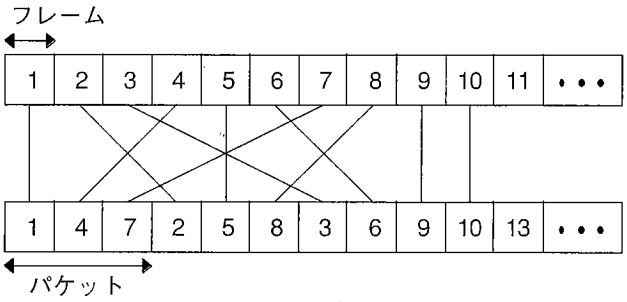

図-8　インタリーブによるパケットロス対策

■音声処理の新しい流れ

これまでの音声符号化研究は，電話用途を前提として進められてきた．一方，ユーザニーズは「安ければよい」という時代からより高品質で付加価値のあるサービスへと移りつつある．現在の代表的な符号化方式は8kHzサンプリング（3.4kHz帯域）を前提としている．3.4kHz帯域の音は電話機のハンドセットで用件を伝えるには十分であるが，スピーカーで再生すると臨場感の乏しい音質になってしまう．パソコンをはじめとする民生機器では11.025kHzサンプリング（5kHz帯域）が普及していることもあり，最近では5kHz帯域の音声を対象とした音声符号化の研究も進められている．かつては，サンプリングレートを上げるとサンプリングレートに比例してビットレートが増加したものだが，CELP系符号化モデルを使えば，サンプリングレートとは独立に特定のビットレートの方式を設計でき，7.8kbit/sで設計した5kHz帯域の方式が，8kbit/sで3.4kHz帯域のG.729よりも主観品質が高いという報告もなされている[5]．また，音声を圧縮するというコストダウン的な発想から，信号処理技術を使って付加機能をつける動きもある．音程を変えない発声速度の変換や音声の編集加工など，従来のオーディオ機器ではできない機能の実現はその代表例である．IPネットワーク上で音声の通信を行うアプリケーションは今後ますます増えていくものと考えられるが，音声処理技術を使ってより豊かなコミュニケーションが実現することを期待している．

参考文献

1）大室他: DualSpeech音声符号化ソフトウェア, NTT R&D, Vol.47, No.5, pp.37-42 (1998).

2）Ramabadran, T.V. et al.: Background Noise Suppression for Speech Enhancement and Coding, 1997 IEEE Workshop on Speech Coding for Telecommunications Proceedings, pp.43-44 (1997).

3）三関他: 音声・背景雑音分離に基づく低レート音声符号化, 音講論, 2-7-1, pp.235-236 (1998-03).

4）大室他: 低ビットレート音声符号化における背景雑音付加音声の品質向上, 信学技報, SP98-145, pp.69-74 (1999).

5）原田他: 5kHz帯域低ビットレート音声符号化－帯域感と主観品質の関係－, 信学技報, SP98-144, pp.63-68 (1999).

（平成11年3月15日受付）