

オーディオミキサのスマート化と field programmable gate array (FPGA) による実装*

高橋 弘太 (電気通信大学)**

1. はじめに

本解説では、オーディオミキサのスマート化技術について簡潔にまとめた後、field programmable gate array (FPGA) を用いた信号処理システムとしての具体的な実装例を紹介する。ミキシング技術に興味のある方だけでなく、FPGA によるリアルタイムでの音響信号処理に興味のある方の参考になれば幸いである。

2. オーディオミキサのスマート化

オーディオミキサの核は、複数の音信号をレベル調整して同時線形混合する機構である。この核に、パン (左右の定位操作)、イコライザ (周波数依存のゲイン付与)、コンプレッサ (非線形のゲイン制御)、リバーブ (残響付与) などの調整機構を加えたものを本稿ではオーディオミキサと呼ぶことにする。図-1 に典型的な構成を示す。

オーディオミキサのスマート化の研究は、図-1 中の各機構の操作の自動化から始まった。その中でも最初に手がつけられたのは、フェーダ操作 (レベル調整) の自動化である。例えば、Dugan [1] の自動ミキサは、複数のマイクロホンによる会議音声の集音時に、各マイクロホンのゲインを自動的に決定するシステムである。この流れを汲んだ処理系はミキサの拡張ボードや digital audio workstation (DAW) のプラグインとして現在でも使われている。

その後、研究の対象は、パンの自動化 [2]、イコライザの自動化 [3]、コンプレッサの自動化 [4]、リ

バーブの自動化 [5] などに広がっていった [6]。

また、これらのスマート化を実現手段で分類すると、機械学習による方法とアルゴリズムによる方法 (知識ベースに基づく方法) に二分される [7, 8]。後者は、設計者がプロの操作者の意見などを聞いたり独自の仮説を立て、それに基づいて最適化問題を構成するなどのアプローチで操作量決定アルゴリズムを作りあげる方法である。本稿の後半で紹介するスマートミキサも後者に属する。

スマート化は、単なる自動化にとどまらず、人間による操作では到達できなかった高度なミキシングを狙う方向にも広がっている。以下では、ライブやレコーディングスタジオにおける音楽制作に絞って高度なミキシングについて解説する。具体的にはマルチトラック音源を (例えば 2ch に) ミックスダウンする場面である。ミックスダウン時にはトラック間でのマスキング現象が起こり、プロの操作者でさえ対処が困難な場合がある。

マスキングとはある強い音 (マスク) によって別の音 (マスキ) の知覚が妨害されることである。同時マスキングと継時マスキングに分類され、前者においては、ある周波数のマスクがその近傍の周波数のマスキの知覚を妨害することから、ミキシングではこの妨害を避けるようにフェーダやイコライザを操作することが望まれる。

さて、音楽制作の場合、単にマスキングを避けるだけでは十分ではない。マスクはマスクが存在しないときのスペクトルに近いものとして知覚されることが好ましい。このため、マスクの存在下で知覚されるマスキのラウドネスであるパーシャルラウドネス [9–11] の概念を理解しておくことが重要である。

ミキサへのマスキング対策の具体例としては、自律的な調整機構を持つ例 [12] の他、操作者に指針を与える機能にしばった例 [13] もある。どちら

* Smart audio mixers and field programmable gate array (FPGA) implementation.

** Kota Takahashi (The University of Electro-Communications, Chofu, 182-8585) e-mail: kota.t@uec.ac.jp
[doi:10.20697/jasj.79.12.621]

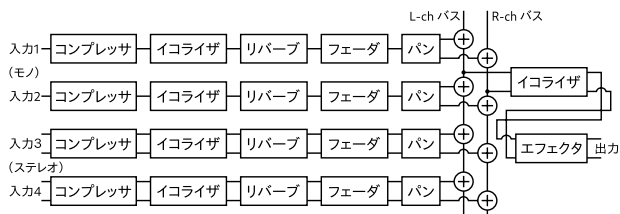


図-1 2ch オーディオミキサの典型的な構成。コンプレッサとイコライザは入れ替えるなど処理の順番はこれに限らない。エフェクタとしてリバーブやマキシマイザやリミッタなどがあり、図の位置以外にも挿入される。

のタイプについても研究論文だけでなく DAW のプラグインとして製品化されているものもある。

3. スマートミキサ

3.1 基本概念とアプリ版及び業務用システム

以下、著者らが開発しているスマートミキサ [14–16] を紹介する。スマートミキサは、ライブや楽曲制作において大音量の伴奏（以下、非優先音と呼び N と略す。前章の観点からはマスカと言ってもよい）が存在してもボーカル（以下、優先音と呼び P と略す。マスキリーと言っても良い）が明瞭に聴き取れることを目標に開発された。

パーシャルラウドネスをきめ細かく制御するためには図-1 の枠組みでは不十分である。なぜなら、パーシャルラウドネスの評価には、周波数ごとに P と N の強度を相互に参照する必要があるが、図-1 においては、周波数分析や相互参照の機構を配置する余地がないからである。そこで我々は、ミキシングの枠組み自体を考え直すことにした。新しい枠組みを図-2 を使って説明しよう。P と N を短時間 FFT (fast Fourier transform) で周波数分解し、これを 1 サンプルシフトで時間軸方向にも展開することで 2 枚の時間周波数平面を構成する。この 2 枚の平面を局所的に相互参照して 1 枚の平面を生成する操作としてミキシングをとらえようというのが新しい枠組みである。

この処理を定式化しよう。 n を離散時間として、サンプリング周波数 F_S で標準化した P を $x_P[n]$ 、N を $x_N[n]$ とする。両信号を 1 サンプルシフトで N_F 点の短時間 FFT して $X_j[n, k]$ (j は P 又は N) を得る。

$$X_j[n, k] = \sum_{i=-N_F/2+1}^{N_F/2-1} h[i] x_j[n+i] \cdot \exp\left(-\frac{j2\pi kn}{N_F}\right) \quad (1)$$

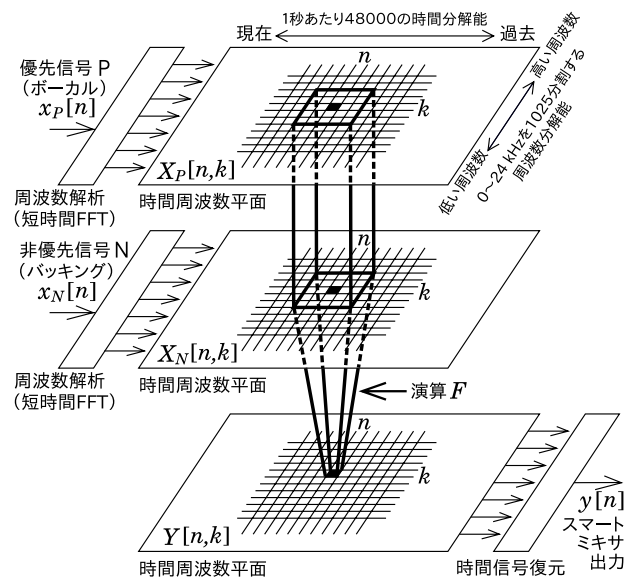


図-2 スマートミキサの出力平面の 1 点の値は 2 枚の入力平面の近傍の値を相互参照する演算 F で決定する。

ここで、 k は周波数ビン番号、 $h[i]$ は窓関数、 j は虚数単位である。スマートミキサは、混合結果 $Y[n, k]$ を、点 (n, k) の近傍における X_P と X_N の値に演算 F を施すことで決定する。例えば、

$$Y[n, k] = \alpha_P[n, k] X_P[n, k] + \alpha_N[n, k] X_N[n, k] \quad (2)$$

を F の具体形とするならば、スマートミキサの良し悪しの鍵は、 $[n, k]$ の近傍における $\{X_P, X_N\}$ から $\{\alpha_P, \alpha_N\}$ を決定するアルゴリズムが握っている。なお、 $Y[n, k]$ から時系列 $y[n]$ への変換は、

$$y[n] = \frac{1}{N_F} \sum_{k=0}^{N_F-1} Y[n, k] \quad (3)$$

$$= \frac{1}{N_F} \left(Y[n, 0] + 2 \sum_{k=1}^{N_F/2-1} Y[n, k] + Y\left[n, \frac{N_F}{2}\right] \right) \quad (4)$$

となり、逆FFTは不要である。1 サンプルシフトの採用は、高音質と並んでこのような利点がある。ちなみに、式 (3) の代わりに式 (4) を用いることで扱うビン数が約半分になる。例えば、 $N_F = 2,048$ の場合は、 $k = 0$ から $k = 1,024$ までの 1,025 ビンのみを扱えばよい。 $y[n]$ の算出部だけでなく、ほぼすべての処理においてこの省略化が可能である。

我々は、スマートミキサの最初の製品として、図-3 に示す iOS アプリ [17] を開発した。このア



図-3 スマートミキサ技術を応用したアプリ omoide

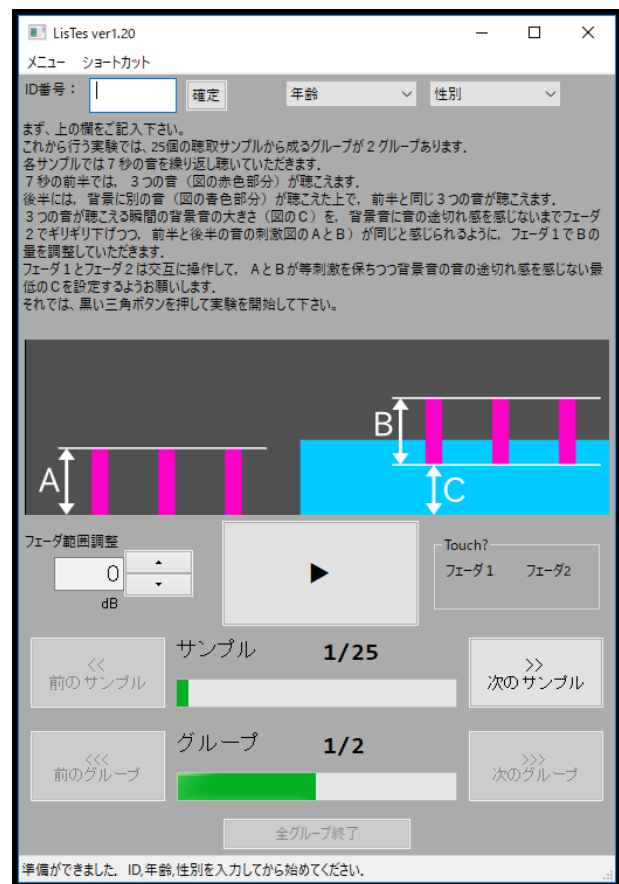
プリでは、ゲイン $\{\alpha_P, \alpha_N\}$ 決定にシンプルな規範を用いた。具体的には、入力対数強度の和を出力対数強度に近づけることと、P の増幅によって生じたパワー増加量を超えない範囲で N のパワーを減少させることの二つの規範である。

次に、我々はスマートミキサ HCU-288N を開発した。HCU-288N は業務用であり、レイテンシ（入出力間の遅延時間）が小さく、かつ良質な音質が求められたため、FPGA による専用システムとして構成した。また、マスキングを少なくするために、ゲイン決定の規範も根本から見直した。

高いレベルでのマスキング対策のためには、[9, 10] など多くの研究者が認めるパーシャルラウドネスのモデルを採用することが最良と考えるかもしれない。しかし、これらのモデルを正確に再現しようとすればレイテンシが大きくなる。また、我々は、N に対して欠落間を知覚させない範囲内で必要時間減衰させることとした。この減衰量に対しては既存のモデルは無力である。それだけでなく、難波 [18] が述べているように、創作的意図を持った作品の聴取に際しては「芸術的、認知的要因など物理量以外の要因」も関与する可能性があり、完全なモデル化は困難であると思われた。

以上の理由から、我々は、新たな聴取実験を行った。それに加えて、ライブ会場現場において操作者が芸術的センスに基づいてゲイン決定の規範の再調整を行う操作パネルも用意することとした。

図-4 に聴取実験用に作ったソフトウェア LisTes の画面と実験の様子を示す。被験者には、前半で呈示される三つのバーストの P が、N が重畳した後半でも等刺激と感じ、かつ N が途切れて知覚されないように、P と N の音量をフェーダで調整し



(a) LisTes の画面

実験が間違いなく進むように監視。聴取サンプルを聴きフェーダを操作。
(b) オペレータ席 (c) 被験者席

図-4 LisTes の画面と心理実験の様子

てもらいその調整量を記録した。この実験を周波数（3 種類）、音の種類（6 種類）、強度の組み合わせ（25 種類）のすべての組み合わせで 20 人の被験者を対象に実施した。100 時間超の実験で、 $2 \times 3 \times 6 \times 25 \times 20 = 18,000$ 個の調整量が得られ、これを基礎データとして多項式近似でモデル化して $\{\alpha_P, \alpha_N\}$ を決定する規範を定めた。

3.2 FPGA での音響信号処理システムの実装

FPGA として、AMD (Xilinx) の Zynq Ultra-scale+ シリーズの XCZU9EG を選んだ。この FPGA は、PL 部（ロジック部）と PS 部（ARM Cortex-A53 によるプロセッサ部）から構成される。PL 部では信号処理を行い、PS 部では操作パネルによるパラメータ設定や時間周波数平面にお

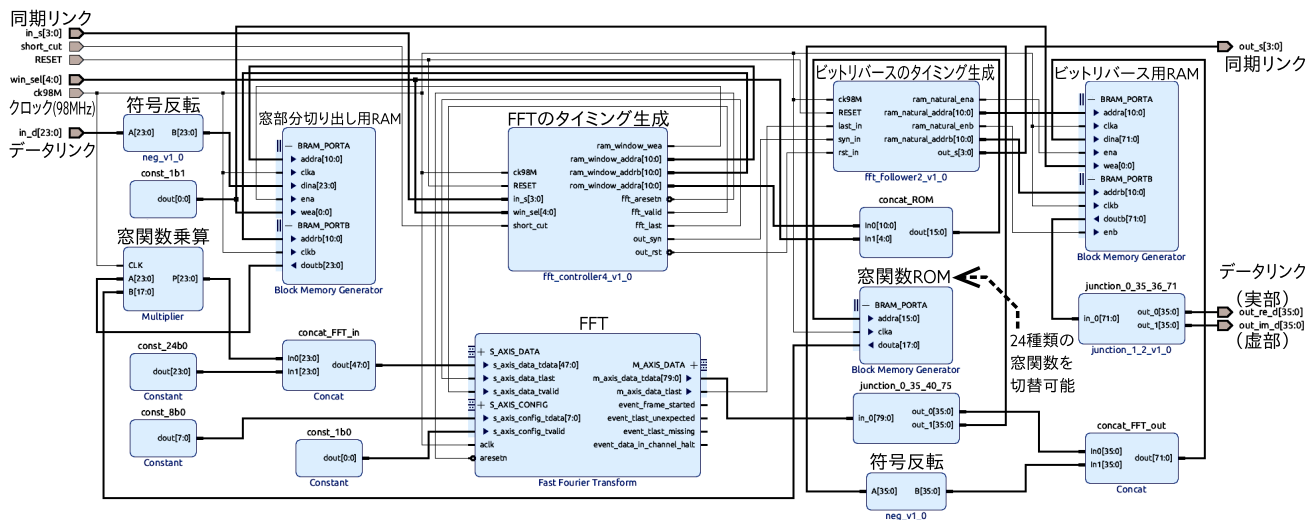


図-5 1 サンプルシフトのスライディング窓での短時間 FFT

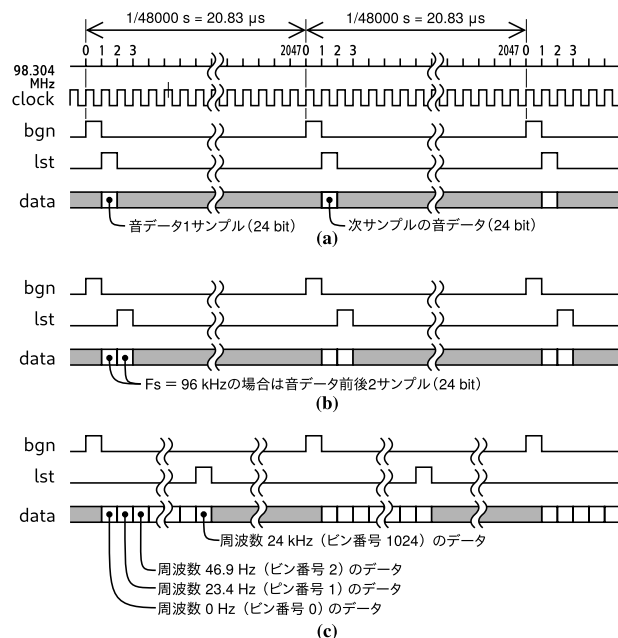


図-6 同期リンクとデータリンクによる IP コア間結線

けるスペクトルやゲインの描画を行う。実時間信号処理を PL 部で完結させることにより、ソフトウェアのバグや予期しない操作で一瞬のストールやハングアップがあったとしても、出力をそのまま出し続けることができるというメリットがある。

PL 部はサンプリング周波数 $F_S = 48 \text{ kHz}$ の 2,048 倍である 98.304 MHz のマスタクロックで動作させる。短時間 FFT の IP コアを図-5 に示す。1,025 ビンの出力は、 $1/F_S$ 秒の区間にある 2,048 クロックのうちの前半の 1,025 クロックで出力される。出力のタイミングを図-6(c) に示す。

短時間 FFT 以降の処理は、処理要素を IP コアとして実装し互いに連結して全体システムを構

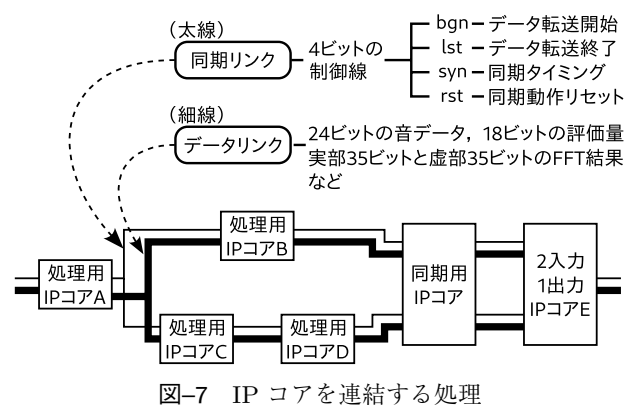


図-7 IP コアを連結する処理

成する方針をとった。コア間の接続には統一したプロトコルを定めた。これらの方針により大規模な処理系も容易に構成できた。図-7 に示すように、コア間の接続は同期リンクとデータリンクで行う。同期リンクは bgn, lst, syn, rst の 4 本の制御線で構成される。bgn と lst で、時分割転送されるデータの最初と最後のピンを指し示す。時系列データの場合は、図-6(a) のように bgn と lst のパルスを隣接させて 1 サンプルだけ送信する。 $F_S = 96 \text{ kHz}$ の場合など、2 サンプル送信する場合には、図-6(b) の手順で送信できる。このように、時系列と時間周波数系列を統一して扱うことができる。

残り 2 本の制御線 syn, rst について図-7 で説明する。IP コアの入出力間にはレイテンシが存在するため、同期用 IP コアなしでは IP コア E の二つの入力間には時間ズレが存在する。このため、IP コア B の出力と IP コア D の出力を比較して、レイテンシの遅いほうに合わせて IP コア E に入力してやる同期用 IP コアを設計し挿入した。syn と rst

は同期用 IP コアの動作に必要な制御線である。

図-8で動作を説明する。まずrstが1クロック分立ち上がり、IP コア間を伝搬し同期用 IP コアはリセットされる。42.7ms後にsynが立ち上がるが、経路によってレイテンシに差があるのでsynの到着時間に差があらわれる。同期用 IP コアはこの時間差を計測し、先に到着したデータをデュ

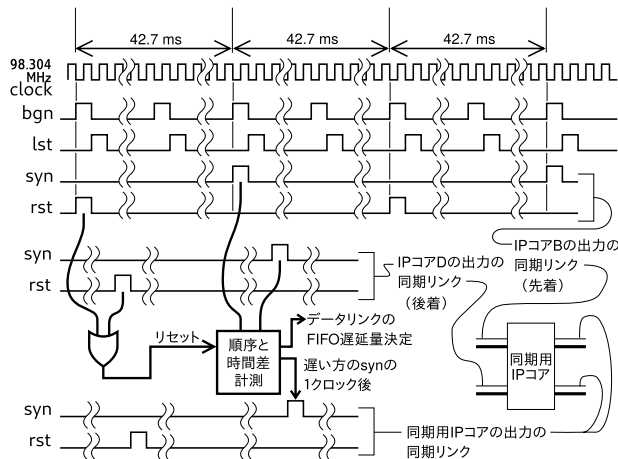


図-8 同期の説明

アルポート RAM に書き込みこの時間差に合わせて読み出すことでレイテンシ差を吸収する。同期用 IP コアの設計図を図-9に示す。この IP コアを必要箇所に挿入しておくことで、動作中のパラメータ設定の変更でレイテンシが変化する経路があったとしても自動的に調整が行われる。

以上の発想に基づき、IP を連結して構成した PL 部の全体図を図-10に示す。入力を入力マトリクスで P、N の L-ch, R-ch に分配された後、解析用と変更用の短時間 FFT で 1,025 ビンの信号と

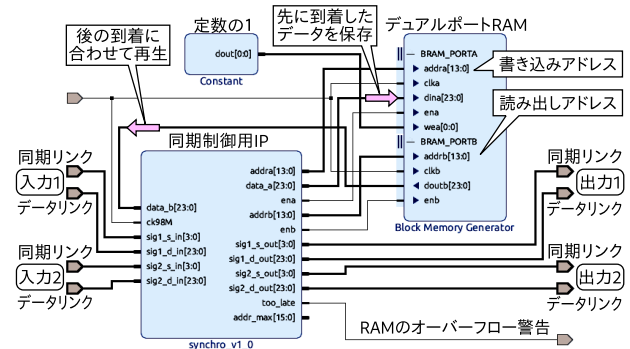


図-9 同期用 IP コアの設計図

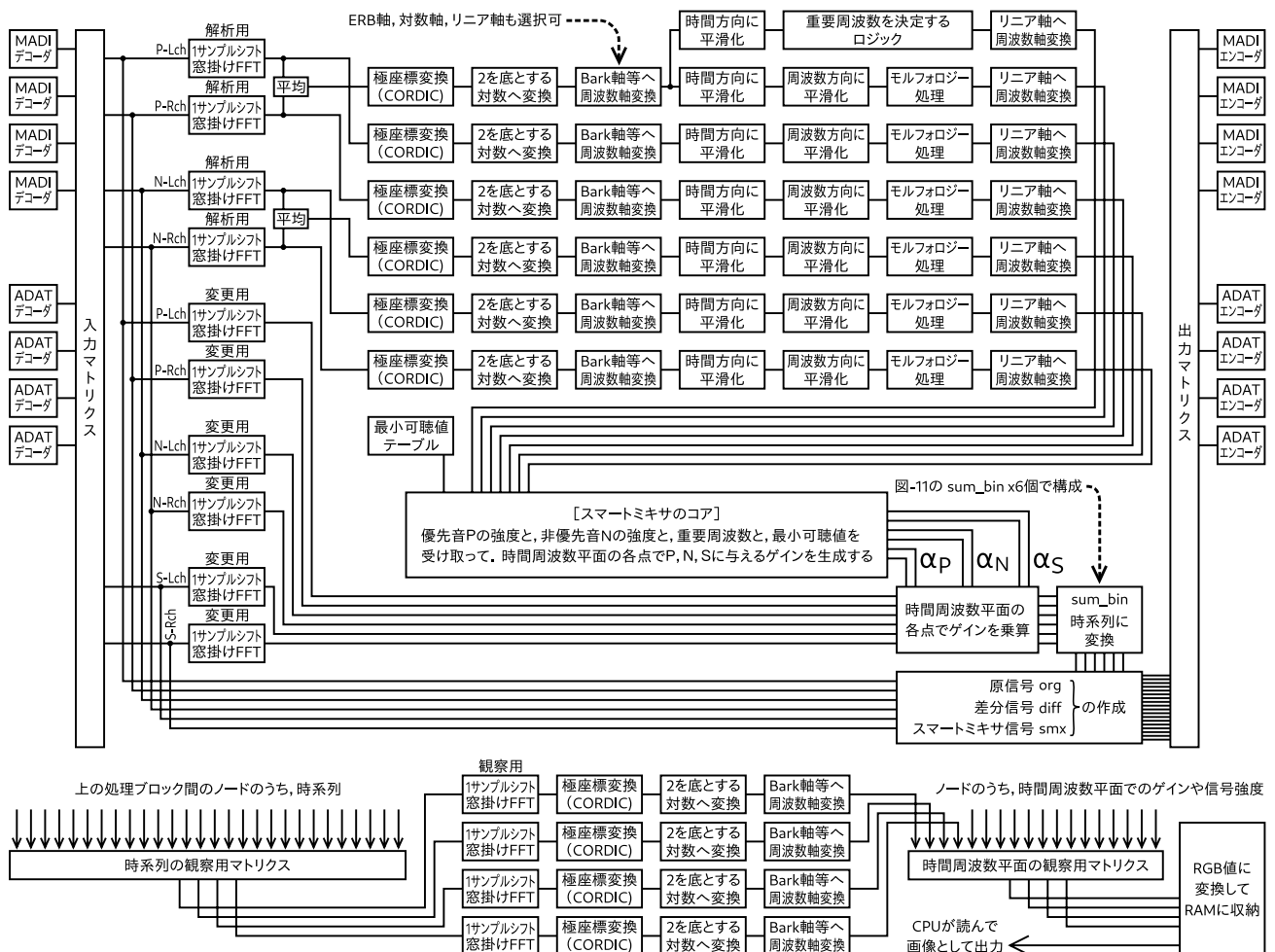


図-10 FPGA の PL 部 (ロジック部) 上に構成した信号処理のブロック図

```

module sum_bin
(
  input wire ck98M, // 98.304 MHz クロック (48000 x 2048)
  input wire [3:0] sig_s_in, // 入力同期リンク (bgn, lst, syn, rst)
  input wire [35:0] sig_d_in, // 入力データリンク (FFT出力35bit)
  output wire [3:0] sig_s_out, // 出力同期リンク (bgn, lst, syn, rst)
  output reg [23:0] sig_d_out // 出力データリンク (音波形24bit)
);
  reg bgn_reg; // レジスタ受け・遅延 1
  reg bgn_reg_2; // レジスタ受け・遅延 2
  reg lst_reg; // レジスタ受け・遅延 1
  reg lst_reg_2; // レジスタ受け・遅延 2
  reg lst_reg_3; // レジスタ受け・遅延 3
  reg [35:0] sig_reg; // レジスタ受け・遅延 1
  reg [45:0] acc; // アキュムレータ
  reg enable; // bgnからlstで決まる加算区間を表示
  assign sig_s_out[0] = lst_reg_2; // 出力bgnは入力lstの2clock遅れ
  assign sig_s_out[1] = lst_reg_3; // 出力lstは入力lstを3clock遅れ
  assign sig_s_out[2] = syn_reg; // syn_regを生成するコードは省略
  assign sig_s_out[3] = rst_reg; // rst_regを生成するコードは省略
  //-----
  // 入力をレジスタで受け止める
  always @(posedge ck98M) begin
    bgn_reg <= sig_s_in[0];
    bgn_reg_2 <= bgn_reg;
    lst_reg <= sig_s_in[1];
    lst_reg_2 <= lst_reg;
    lst_reg_3 <= lst_reg_2;
    sig_reg <= sig_d_in;
  end
  //-----
  // enableは、bgn～lstの区間で1にする。加算の範囲を示す変数である
  always @(posedge ck98M) begin
    if (bgn_reg == 1'b1) begin // bgn到着でenableを立てる
      enable <= 1'b1;
    end else if (lst_reg == 1'b1) begin // lst到着でenableを降ろす
      enable <= 1'b0;
    end
  end
  //-----
  // 0番目のbinから1024番目のbinまで値を順に加算していく。
  always @(posedge ck98M) begin
    if (bgn_reg_2 == 1'b1) begin // 0 binの値は、
      acc <= {{10{sig_reg[35]}}, sig_reg[35:0]}; // accを上書き
    end else if (lst_reg == 1'b1) begin // 1024 binの値は
      acc <= acc + {{10{sig_reg[35]}}, sig_reg[35:0]}; // accに加算
    end else if (enable == 1'b1) begin // 1～1023bin
      acc <= acc + {{9{sig_reg[35]}}, sig_reg[35:0], 1'b0}; // 1024 bin
    end
  end
  //-----
  // 加算の結果24ビットで表現できなくなった場合は飽和させる
  always @(posedge ck98M) begin
    if (lst_reg_2 == 1'b1) begin // 出力タイミングに出力し省電力
      if (acc[45] == 1'b0) begin // 和の最終値が正のとき
        if ( ( | acc[44:34] ) != 1'b0 ) begin // 上位が000以外で
          sig_d_out <= 24'h7FFFFF; // 正の最大値に置き換え
        end else begin
          sig_d_out <= acc[34:11]; // 切り取って大丈夫
        end
      end else begin // 和の最終値が負のとき
        if ( ( & acc[44:34] ) != 1'b1) begin // 上位が111以外で
          sig_d_out <= 24'h800000; // 負の最小値に置き換え
        end else begin
          sig_d_out <= acc[34:11]; // 切り取って大丈夫
        end
      end
    end
  end
endmodule

```

図-11 時間周波数データから時系列データに戻す IP コア sum_bin の verilog コード
syn と rst の生成部は省略。

なる。解析用の信号は、その後、極座標へ座標変換され得られたノルム成分は対数変換される。更に、周波数軸をリニアから Bark 軸に座標変換された後、時間方向の平滑化と周波数方向の平滑化とモルフォロジー処理によって時間周波数平面での近傍への影響が加味される。Bark 軸への変換を行う理由は、周波数方向の平滑化の際の近傍の定義を Bark 軸の尺度で行うことができるからである。リニア軸に逆変換されたデータはスマートミキサのコアに導かれ、 $\{\alpha_P, \alpha_N\}$ が導出される。解析用とは別の変更用の窓関数で短時間 FFT

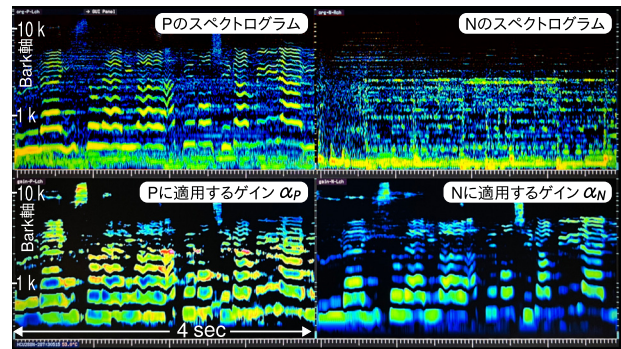


図-12 時間周波数データの動画像

4 種の動画像が左へ滑らかにスクロールする。スペクトログラムについては黒で値が低く、ゲインについては黒が 0 dB を示している。

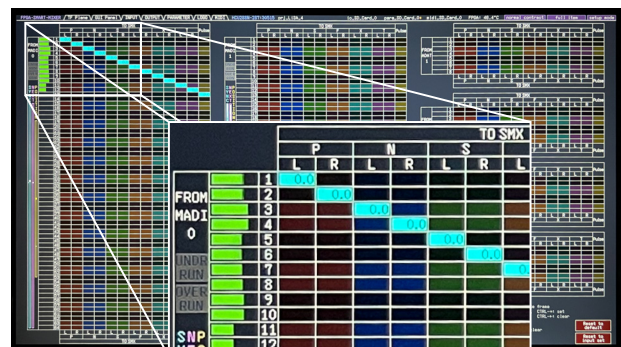


図-13 入力マトリクス

MADI や ADAT 入力を P や N に任意のゲインで結線できる。左上部分を拡大表示した。

された信号にこの $\{\alpha_P, \alpha_N\}$ のゲイン操作を施し、時系列に戻し、出力マトリクスを経て出力が得られる。別の窓関数を用いる利点は、解析用には高い周波数分解能が要求される一方で、変更用には低レイテンシが要求されるため、それぞれに適した窓関数を選ぶようにするためである。また、紙面の都合で詳細は省略するが、HCU-288N にはコーラス専用の入力 S があり、 α_P に必要な修正を施した α_S によるゲイン操作が施されている。

IP コアは、Verilog 言語で作成したオリジナル IP コアや AMD (Xilinx) 提供の IP コアを用いた。必要に応じてこれらを相互結線して上位階層の IP コアも作った。例えば、図-5 中に二つあるタイミング生成 IP は Verilog 記述である。また、式 (4) を計算する IP コアのコードを図-11 に示す。簡単な記述で実用処理が実装可能であることが分かる。

また、HCU-288N には動画像出力機能があり、時間周波数平面のスペクトルやゲインを横スクロールしながらリアルタイムで観察できる (図-12)。動画像データは PL 部の RAM にいったん蓄えら

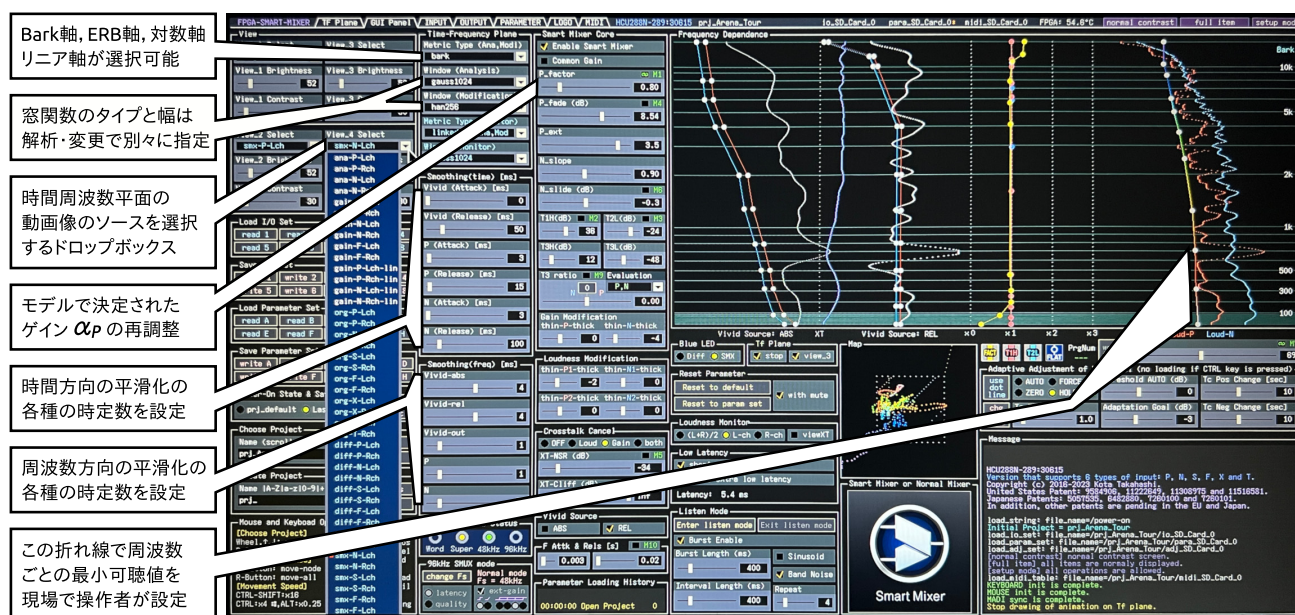


図-14 操作パネル

れた後、DMA によって PS 部の DRAM 上に展開され、 $1,920 \times 1,080$ で 60 fps の動画像として送り出される。これらの処理部は、図-10 の下部に描かれている。切替えにより、同じディスプレイ上に、入力マトリクス (図-13) や出力マトリクスや操作パネル (図-14) などとも表示できる。

なお、本稿では説明を省略するが、HCU-288N には、P と N と S 以外にも、エフェクト用の F、かぶりキャンセル用の X、スルーの T のカテゴリがあり、それぞれ音源の性質にあわせたミキシングが行われるなど、様々な拡張が施されている [19]。表-1 にそれらに対する機能も含めた PL 部のリソースの使用数を示す。FPGA の消費電力は 12 W である。ラックマウント型のスマートミキサ装置として完成した HCU-288N を図-15 に示す。

3.3 ライブ会場での運用

本稿で紹介した HCU-288N と、その前代機の HCU-3200R を用いて、2019 年よりスマートミキサの実証実験を行い、現場からのフィードバックを受けて現在でも両機種種の改良を継続している。現在までに、小さな会場としてはライブハウス、大きな会場としては、横浜アリーナ、埼玉スーパーアリーナ、大阪城ホールなどでの運用実績がある。

会場においては、図-16 に示すようにメインのミキシング卓で P, N, S などのチャンネルにミックスダウンし、それをスマートミキサに送ってミキシングを実施する形での運用が行われている。

表-1 FPGA 内のリソース使用数

Zynq Ultrascale+ におけるリソース種別	XCZU9EG 使用可能量	図-5 単独	HCU-288N での使用量
ルックアップテーブル	274,080	4,146	222,638
フリップ・フロップ	548,160	731	312,158
RAM (36 K × 1 bit)	912	36	906
DSP (積和演算器)	2,520	32	1,042

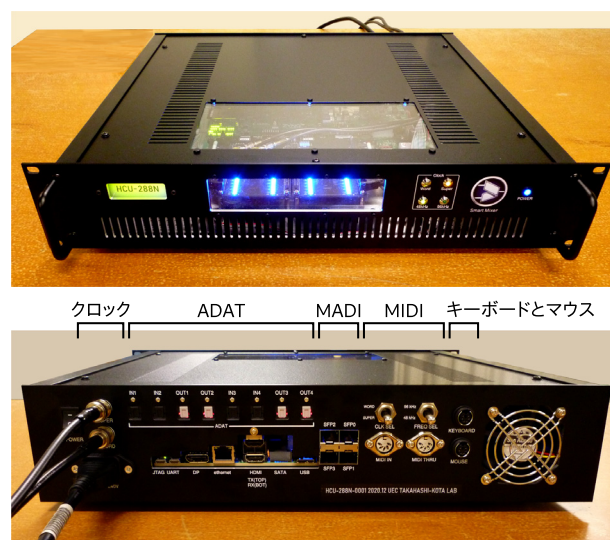


図-15 HCU-288N の外観
フロント側とリア側からの写真

現場からは、ボーカルの細かなニュアンスまで会場全体に伝えることができるという評価をいただいております。気に入っていただいたオペレータの方々には継続して使用していただけている。また、運用中に当初は想定してなかった効用があること



図-16 アリーナ施設で運用中のスマートミキサ

も明らかになった。それは、ハウリング抑制である。スマートミキサは、ある時間のある周波数について、優先音の強度が相対的に弱い領域を限定的に強調する。このため、ハウリングが起りかけて強度が増加した瞬間に該当する周波数での優先音の強調が弱まり、結果的にハウリングが抑止される。これにより、声量の大きくない歌手であっても十分な音量で観客に声を届けることができる。このような副次的効果も歓迎されている。

現在までに、図-15のようなスマートミキサを19台製作した。上で述べたような高い評価をいただいたこともあって数台が現場で活躍している。

4. おわりに

本稿では、まず、ミキサのスマート化について取り上げた。自動化の意味でのスマート化では、機械学習による方法とアルゴリズムによる方法があり、現時点までの研究発表の累積ではアルゴリズムによる方法のほうが多い。しかし、今後は機械学習による方法が大きく伸びてくるだろう。

ところで、ミキシングという行為は創作活動であり、正解の存在しない行為である。特に、音楽制作に関して言えば、芸術的活動でさえある。これは、(ミキシングとは逆の操作の) 信号分離には正解が存在することとは対照的である。

そこで、本稿の後半で紹介したスマートミキサにおいては、操作パネルを設けて、操作者（オペレータ）が、会場やアーティストに合わせて、モデルとしてあらかじめ設定されたパラメータを再調整できるようにしてある。例えば、操作パネル（図-14）には、「ゲイン α_P 」の再調整と書かれた

スライダがある。このスライダにより LisTes による実験（図-4）で作ったモデルで決まるゲイン α_P を再調整できるようになっている。このように人間が介入できる仕組みは必ず必要だと考えている。

最近の FPGA の能力は凄まじい。膨大な処理が簡単に実現できてしまう。FPGA による音響信号処理系を創作するワクワク感をぜひ伝えたいとの思いもあって本稿の執筆をお引き受けした。解説記事としてはやや詳しくすぎるとも見える図-5から図-11をあえて掲載したのも設計の雰囲気を感じていただくためである。本稿をきっかけに一人でも多くの方にこの楽しさを味わっていただくことができたならば著者として望外の喜びである。

謝 辞

約8年間にわたって共同研究を一緒に進めて下さっているヒビノ株式会社の皆様に感謝します。また、ライブ現場のPA席での豊かな経験から、貴重なアドバイスを下さっている同社のオペレータの方々に厚く御礼申し上げます。そして、アプリ omoide を共作して下さいました西方夏子氏に感謝の意を表します。また、博士後期課程在学中に研究に協力して下さいました大脇渉博士に感謝します。

文 献

- [1] D. Dugan, "Automatic microphone mixing," *J. Audio Eng. Soc.*, 23, 442-449 (1975).
- [2] E. P. Gonzalez and J. D. Reiss, "Automatic mixing: Live downmixing stereo panner," *Proc. Int. Conf. Digital Audio Effects (DAFx-07)*, pp. 63-68 (2007).
- [3] E. P. Gonzalez and J. D. Reiss, "Automatic equalization of multichannel audio using cross-adaptive methods," *AES Conv.*, 127, 6 pages (2009).
- [4] J. A. Maddams, S. Finn and J. D. Reiss, "An autonomous method for multi-track dynamic range compression," *Proc. Int. Conf. Digital Audio Effects (DAFx-12)*, 8 pages (2012).
- [5] E. T. Chourdakis and J. D. Reiss, "A machine learning approach to application of intelligent artificial reverberation," *J. Audio Eng. Soc.*, 65, 56-65 (2017).
- [6] B. De Man, J. D. Reiss and R. Stables, "Ten years of automatic mixing," *Proc. 3rd Workshop Intelligent Music Production*, 5 pages (2017).
- [7] Automatic mixing research, <https://csteinmetz1.github.io/AutomaticMixingPapers/index.html> (参照 2023-06-04).
- [8] D. Moffat and M. B. Sandler, "Approaches in intelligent music production," *Arts*, 8(4), 125, 14 pages (2019).
- [9] B. C. J. Moore, B. R. Glasberg and T. Baer, "A model for the prediction of thresholds, loudness and partial loudness," *J. Audio Eng. Soc.*, 45, 224-240 (1997).

- [10] B. R. Glasberg and B. C. J. Moore, "Development and evaluation of a model for predicting the audibility of time-varying sounds in the presence of background sounds," *J. Audio Eng. Soc.*, **53**, 906–918 (2005).
- [11] 鵜木祐史, "新しいラウドネス計算法・ISO 532-2:2017 Moore-Glasberg methodn," 音響学会誌, **77**, 790–797 (2021).
- [12] S. Hafezi and J. D. Reiss, "Autonomous multi-track equalization based on masking reduction," *J. Audio Eng. Soc.*, **63**, 312–323 (2015).
- [13] J. Ford, M. Cartwright and B. Pardo, "Mixviz: A tool to visualize masking in audio mixes," *Audio Eng. Soc. Conv.*, **139**, 10 pages (2015).
- [14] 高橋弘太, 大脇 渉, "スマートミキサ～基本原理と有効性～," 信学技報/応用音響研究会, **111**(333), (EA2011-103), pp. 43–48 (2011).
- [15] W. Owaki and K. Takahashi, "Novel sound mixing method for voice and background music," *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP) 2015*, pp. 290–294 (2015).
- [16] 大脇 渉, 高橋弘太, "時間周波数平面を用いた音声の埋もれない音信号混合法の評価," 信学論, **J99-A**, 168–176 (2016).
- [17] "omoide—かんたんムービー作成," 2015 年リリース <https://apps.apple.com/jp/app/id1023546227>
- [18] 難波精一郎, "知っているようで知らないラウドネス," 音響学会誌, **73**, 765–773 (2017).
- [19] 日本国特許 5057535 (2012), 6482880 (2019), 7260100 (2023), 7260101 (2023), 7292650 (2023). 米国特許 9584906 (2017), 11222649 (2022), 11308975 (2022), 11516581 (2022).



高橋 弘太

1986 東京大学大学院計数工学修士課程修了。博士 (工学)。統計的信号処理, 実時間システムの DSP や FPGA での実装, スマートミキサ, 合唱のミキシングの研究に従事, 独自の話速バリエーション型音声データベース SRV-DB, セパレート音源型音楽データベース SIS-DB を公開し, 研究者のお役に立てることを願っている。