

方向通過型音源分離フィルタへのFPGAの応用

鈴木 紀章[†] 中臺 一博^{††} 天野 英晴[†] 奥乃 博^{†††,††} 北野 宏明^{††}

[†] 慶應義塾大学大学院理工学研究科 〒223-8522 神奈川県横浜市港北区日吉3-14-1

^{††} 科学技術振興事業団ERATO北野共生プロジェクト

^{†††} 京都大学大学院情報学研究所

E-mail: [†]{noriaki,hunga}@am.ics.keio.ac.jp, ^{††}{nakadai,kitano}@symbio.jst.go.jp, ^{†††}okuno@nue.org

あらまし ロボットの知覚システムを構成する場合、計算コストが高い問題を低消費電力かつある程度の柔軟性を保持して実現するためにはFPGAによるリコンフィギャラブルシステムのアプローチが効果的である。本稿では、聴覚システムを構成する際に用いる方向通過型フィルタの処理の一部である高速フーリエ変換(FFT)、平方根、ArcTanをFPGA上に実装し、ハードウェア処理による小型で低消費電力のシステムを作成するための検討を行う。評価の結果、FPGAに実装したFFT、平方根、ArcTan演算回路は、12MHz動作において1GHzのPentium IIIの2.9倍、2.9倍、3.3倍の処理速度で演算可能である事が分かった。

キーワード FPGA, アクティブ方向通過型フィルタ, ヒューマノイドロボット, 音源分離, リコンフィギャラブルシステム

Applying FPGA to Sound Separation by Direction-Pass Filter

Noriaki SUZUKI[†], Kazuhiro NAKADAI^{††}, Hideharu AMANO[†], Hiroshi G. OKUNO^{†††,††}, and

Hiroaki KITANO^{††}

[†] Department of Information and Computer Science, Keio University 3-14-1, Hiyoshi, Kohokuku, Yokohama, 223-8522, Japan

^{††} Japan Science and Technology Corporation ERATO Kitano Symbiotic Systems Project

^{†††} Graduate School of Infomatics, Kyoto University

E-mail: [†]{noriaki,hunga}@am.ics.keio.ac.jp, ^{††}{nakadai,kitano}@symbio.jst.go.jp, ^{†††}okuno@nue.org

Abstract Reconfigurable systems are efficient for high performance but low cost/power implementation for intelligent systems for robots. In this paper, a part of processing for the direction-pass filter, such as Fast Fourier Transform(FFT), square root, and arc tangent used in auditory system of humanoid robots are implemented on an FPGA, and their performance is evaluated. Our result shows that FFT, square root and arc tangent implemented on the FPGA of 12MHz are 2.9 times, 2.9 times and 3.3 times faster, respectively, than those in Pentium III of 1GHz.

Key words FPGA, Active Direction-Pass Filter, Humanoid Robot, Sound Source Separation, Reconfigurable System

1. はじめに

プログラマブルなLSIであるFPGA (*Field-Programmable Gate Array*) は、DSP (*Digital Signal Processor*) や汎用プロセッサと比較し、動作周波数が低くても同等の性能が得られるため、画像処理や信号処理用途への応用が盛んである [1]。また、FPGA上への実装は、ASIC (*Application-Specific Integrated Circuit*) 化が比較的容易であり、製品化にあたって、ASICによる大量生産により、低コスト化が図れることも大きな魅力

であり、様々なLSIのプロトタイプに積極的に用いられてきた [2]。

一方、近年注目されているヒューマノイドをはじめとする自律ロボットでは、知覚・制御処理などの計算コストが高い処理を、ロボット内部に搭載する必要があり、チップ化による小型化が不可欠であった。またチップ化では、長時間駆動が可能ないように省電力であること、低発熱であること、人に不快感を与えないように低騒音であることといった要件を満たすデバイスであることが求められる。さらに、開発中は、プログラムのイ

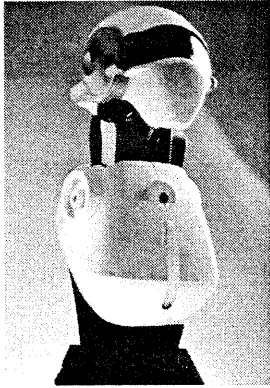


図1 ヒューノイドロボットSIG

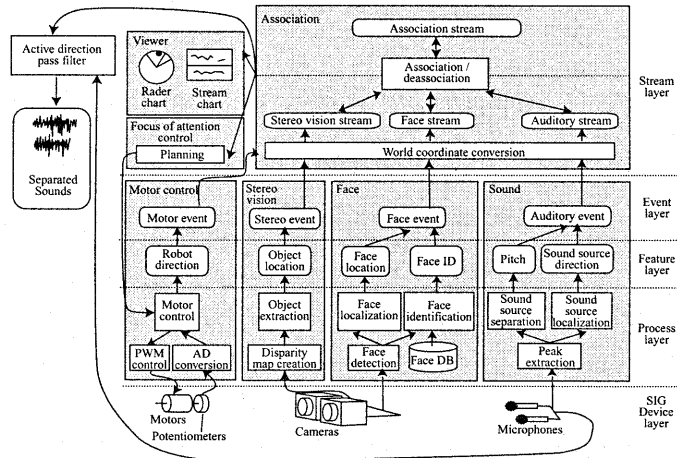


図2 ロボット知覚システム

ンクリメンタルな変更が容易であることも求められるため、将来的に ASIC を見据えた FPGA による実装が有効である。

そこで本研究では、科学技術振興事業団 ERATO 北野共生システムプロジェクトで開発されたロボット知覚システムの中で最も実時間性が要求される聴覚処理の音源分離処理に着目し、FPGA ボードを用いたハードウェア化により、より小型で低動作周波数、低消費電力システムの実現を目指す。FPGA ボードには、慶應義塾大学天野研究室で FPGA と PC の協調動作を目的として開発された、PCIバス接続の汎用 FPGA 搭載ボードを用いた。このボードでは、PCIバス経由でコンフィグレーションが可能であり、様々なアプリケーションに用いることができる。

本稿では、ロボット聴覚システムのハードウェア化の第一ステップとして、音源分離処理に必要な演算処理のうち、FFT 演算回路、平方根演算回路、ArcTan 演算回路の実装評価を通じて、ASIC化する際のソフトウェアとハードウェアの分担方法や、回路の構成等の検討を行う。

以降、2節でロボットシステムについて紹介し、3節で FFT 演算回路、平方根演算回路、ArcTan 演算回路の FPGA への実装について述べる。また、4節でこれらの回路の評価を、5節で今後の検討課題について述べ、6節でまとめる。

2. ロボット知覚システム

科学技術振興事業団 ERATO 北野共生システムプロジェクトでは、人間とソーシャルインタラクションを行うロボットでは、聴覚は本質的な機能の一つであるという考えの下、図1に示す上半身のヒューノイドロボット SIG を用いて、ロボット聴覚の実現に向けた研究を行っている [3], [4]。研究の主な課題は、アクティブな動作を行い聴覚機能を向上させるアクティブオーディション、ロバスト性向上のための視聴覚情報の統合、および、一般的な音の理解であり、実際に、動作時ノイズのキャンセルを行い、実時間で複数人物を追跡し、また複数人が同時発話する場合でも、音源を定位・分離・認識するシステムを実現している。

このシステムは、視聴覚情報、モータ情報、情報統合などを扱う8つのモジュールから構成されているが、各センサ情報の抽出の計算コストが高いことに加え、ロバスト性を高めるために複数のセンサ情報を用いる必要がある。このため高性能の CPU を用いても、1 プロセッサでは実時間処理を行うことが難しい。

そこで、実装上は、ロボットの外部に、5 台の Pentium III 1 GHz の Linux PC を設置し、これらを gigabit イーサで接続した分散アーキテクチャとして構成されている。その結果、全体として約 200 ms のレイテンシを持つシステムを実現している。

本稿では、このうち聴覚モジュールの音源分離処理を扱う。このモジュールでは、処理が間に合わずにバッファオーバーランが発生すると、波形が歪んでしまい結果に大きな影響を与えるため、最も実時間性が要求される。データは周波数領域で処理するため、扱うデータ量や演算量も視覚処理と比較しても決して少ないわけではない。次節では、音源分離を行う聴覚処理について述べる。

2.1 アクティブ方向通過型フィルタによる音源分離処理

音源分離処理では、アクティブ方向通過型フィルタを用いている [4]。システムの構成を図3に示す。

人間とロボットの音声によるインタラクションを実現する場合、話者の口元にマイクを設置する方法をとることもできるが、ロボットのマイクを使って集音するほうがより自然である。その場合、精度のよい認識を行うためにはインタラクションの対象とする話者以外が発生する音声はなるべく排除する必要がある。アクティブ方向通過型フィルタは、その音源分離処理のための一手法である。

アクティブ方向通過型フィルタは、人間の両耳に相当する2本のマイクを使用し、両耳間位相差 (*Interaural Phase Difference, IPD*) と両耳間強度差 (*Interaural Intensity Difference, IID*) から特定方向の音響信号を抽出するフィルタである。このフィルタは、理論上、マイク数と音源数の関係に制約はなく、2本のマイクで2音源以上の分離に対応することができる。また、

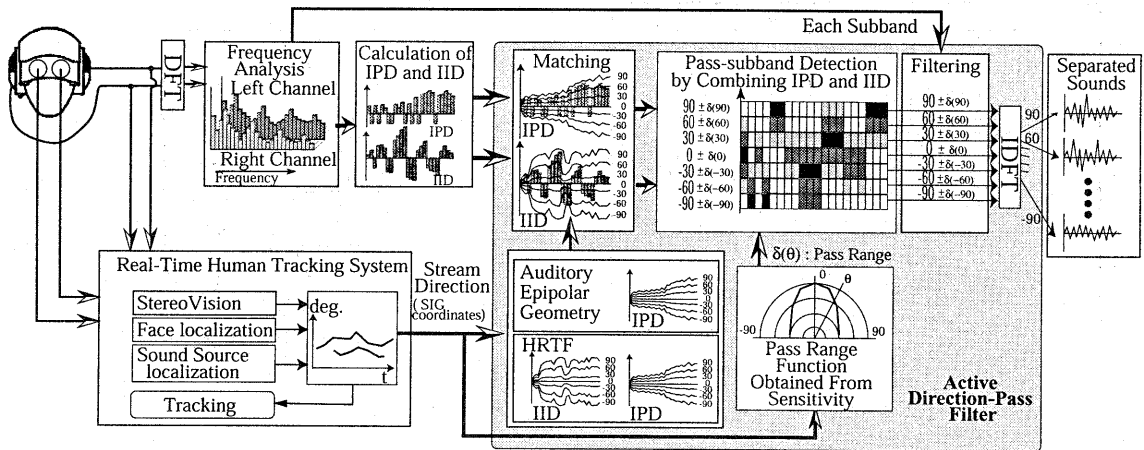


図3 アクティブ方向通過型フィルタによる音源分離

マイクや音源が移動する場合でも特別なチューニングをすることなく対応することができ、ロボットへの搭載に適した手法である。

アクティブ方向通過型フィルタは、スペクトルの各サブバンドで、IPDとIIDに関する仮説推論を行い、抽出方向に対応するIPD、IIDをもつサブバンドのみを通過させることにより、特定方向の音を抽出する。また、方向に対する抽出精度を考慮して、正面方向の音源については、通過させる方向の範囲(帯域)を狭く、正面から離れた音源では、通過させる帯域を広く取るように通過帯域を制御することにより、音源分離精度を向上させている。このような通過帯域制御を行って、正面方向ではS/N比の高い音響信号を抽出し、正面方向から離れた音源に対しては帯域を広く取り、背景雑音の混入によりS/N比は多少落ちるものの、必要な情報をできるだけ抑制せずに、特定の音源を強調することができる。正面方向から離れた音源を精度よく抽出する必要がある場合は、音源方向を向くような制御を行う。

アクティブ方向通過型フィルタのアルゴリズムは以下の6ステップで構成される。

(1) 入力音のスペクトルから、各サブバンドのIPD $\Delta\varphi'$ とIID $\Delta\rho'$ を計算する。ここで、 S_{pl} 、 S_{pr} は、それぞれある時刻に左右のマイク入力信号から得られたスペクトルである。

$$\Delta\varphi' = \arctan\left(\frac{\Im\{S_{pl}\}}{\Re\{S_{pl}\}}\right) - \arctan\left(\frac{\Im\{S_{pr}\}}{\Re\{S_{pr}\}}\right) \quad (1)$$

$$\Delta\rho' = 20 \log_{10}\left(\frac{|S_{pl}|}{|S_{pr}|}\right) \quad (2)$$

(2) 抽出すべき音源の方向を θ_s とする。 θ_s は実時間人物追跡システム [3] から、ロボット座標系^(注1)での水平角として得られる。

(3) 通過帯域関数に従って、 θ_s に対応するアクティブ方向通過型フィルタの通過帯域 $\delta(\theta_s)$ を選択する。選択された通過帯域 $\delta(\theta_s)$ を用いて、 $\theta_l = \theta_s - \delta(\theta_s)$ 、 $\theta_h = \theta_s + \delta(\theta_s)$ と定義し、 θ_l から θ_h の範囲にある音響信号を抽出する。

(注1)：現在ロボットが向いている方向を基準とした座標系

(4) θ_l と θ_h に対するIPD、IIDを推定する。これらの推定には、ロボットの伝達関数を利用する。ロボットの伝達関数は、無響室で水平方向について5度刻みでインパルス応答を計測することによって得られる。

(5) 音源方向 θ に対して、ロボットの伝達関数を利用して、入力スペクトルから以下の条件式を満たすサブバンドを選択する。

$$f < f_{th} : \Delta\varphi_H(\theta_l) \leq \Delta\varphi' \leq \Delta\varphi_H(\theta_h),$$

$$f \geq f_{th} : \Delta\rho_H(\theta_l) \leq \Delta\rho' \leq \Delta\rho_H(\theta_h)$$

$\Delta\varphi_H(\theta)$ 、 $\Delta\rho_H(\theta)$ は、それぞれロボットの伝達関数から推定されるIPD、IIDである。 f_{th} は、フィルタリングの判断基準にIPDとIIDのどちらを用いるかを定める閾値である。一般に、低周波数域ではIPD、高周波数域ではIIDの影響が大きく、この閾値はマイク間距離に依存する。SIGでは、 f_{th} は1500 Hzとしている [5]。

(6) 選択されたサブバンドから、音響信号を再合成し、該当範囲にある音響信号を抽出する。

本稿では、アクティブ方向通過型フィルタに必要な処理のうち、入力音のスペクトルを生成するFFT演算、(1)のIPD、IIDの計算に必要な平方根、ArcTan演算について実装を行った。

3. FPGA への実装

今回のシステムでは、PCIバス接続のFPGA搭載ボードを

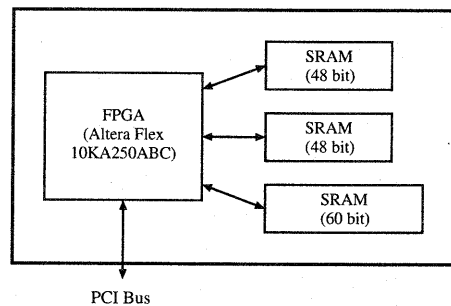


図4 FPGA ボードの構成

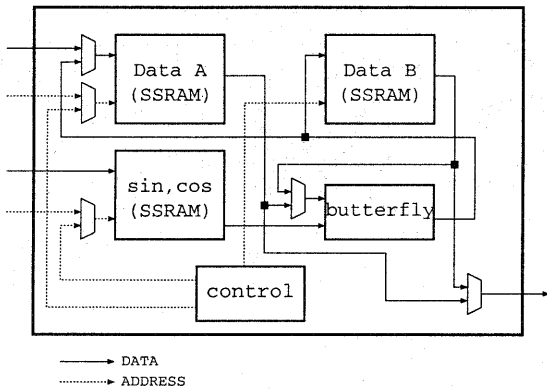


図5 FFT演算回路

PCに搭載し、そのFPGA上に処理回路を実現する事により実装を行った。

ボードには、Altera社のCPLDであるFlex 10KA250ABCが実装されており、3セットのSRAM(2組が32 bit幅、1組が40 bit幅)と接続されている。

ボード上のメモリやレジスタはホストPCのメモリ空間上にマップされ、PC上のプログラムからボードをアクセスする際には、ポインタ等を用いて該当アドレスの読み書きを行う事によってボード内のリソースにアクセスする事ができる。以降、実装したFFT、平方根、ArcTan回路について説明する。

3.1 FFT演算回路

図5は、FFT演算回路の中心部のブロック図である。FFT演算回路のデータ形式は、ホストPCとのデータの受渡しを考慮し、IEEE754単精度実数形式に準拠したフォーマットを用いている。ただし、非数(NaN)、極めて大きい(Inf)または小さい数(-Inf)、丸め誤差等の処理は省略してある。

この回路は256エントリ分の内部メモリを持ち、256ポイントのFFT演算を行う事ができる。アクティブ方向通過型フィルタでは2048ポイントのFFTを行う必要があるため、256ポイントの演算を複数段繰り返し行うことによって2048ポイントのFFTを実現している。

内部は、2組みの256ポイント分のデータメモリ、sin、cos値の格納用メモリ、バタフライ演算ユニット、制御ユニットから構成されている。

演算を行う際には、まずデータメモリAに演算データを、sin、cos用メモリにバタフライ演算に必要なsin、cosの値を格納し、演算開始信号を入力することによって演算を開始する。

信号が入力されると、データメモリAから取り出されたデータは、sin、cos用メモリの値を元にバタフライ演算が施され、データメモリBに格納される。複数段のFFT演算を行う際には、データメモリBに格納されたデータが再びバタフライ演算器を通してデータメモリAに移る。この処理を必要段数繰り返す。このユニットでは、1段の演算から256ポイントのFFT演算段数である7段までの間で、自由に段数を設定することができる。256ポイント以上の演算を行う際には、データメモリに

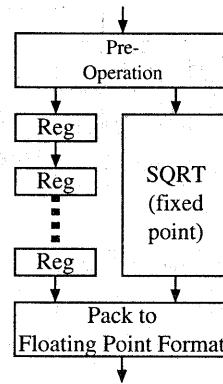


図6 平方根演算器

セットするデータの組み合わせ、sin、cosの値、演算段数をコントロールして繰り返し256ポイントの演算を行うことにより演算を実行する。

3.2 平方根演算器

図6は、浮動小数点平方根演算器のブロック図である。この回路のデータ形式は、FFT演算回路同様、IEEE754単精度実数形式に準拠したフォーマットを用い、FFT同様非数等の処理は省略している。

この演算器はパイプライン構成をとり、1クロック毎に1つの浮動小数点数の投入が可能である。この演算器の中心は、25 bit入力24 bit出力の固定小数点平方根演算器である。

浮動小数点平方根演算器では、前処理ユニットで仮数部を固定小数点平方根演算器に投入するためのシフト、指数部の計算を行う。その後仮数部の演算を固定小数点平方根演算器で行っている間、指数部の値を保存するために、指数部の値は固定小数点平方根演算器のパイプライン段数分レジスタを通過する。固定小数点平方根演算器では、引き放し法[6][7]を用いて平方根演算を行っている。仮数部の演算終了後、後処理として浮動小数点フォーマットに整形して出力を行う。

3.3 ArcTan演算回路

正確なArcTan演算回路をハードウェアにより実現する場合、論理が複雑で、実装した場合には大きな回路規模を要する。そこで、アプリケーションで求められる演算精度を考慮して、浮

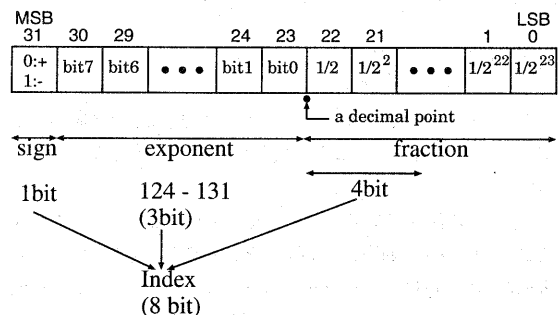


図7 テーブル引きインデックス

表1 2048 pointのFFTの実行時間

実装方法	測定条件	実行時間
CPU(Table)	実測	7352 μ s
FPGA(Table)	実測(PIO転送時間含む)	6938 μ s
FPGA(Table)	Sim.(本体の動作時間のみ)	2535 μ s
CPU(Base)	実測	12144 μ s
CPU(FFT)	実測	677 μ s

(表中の sim. はシミュレーションを指す)

動小数点データの一部を用いてインデックスとし、それを元にテーブル参照を行う疑似的な ArcTan 演算回路を実装した。

音源定位では、5度程度の定位誤差を許容している。そのためこの必要条件を満たすために、0度から90度までの範囲で128点(180度で256点)のデータを持つテーブルを採用した^(注2)。

また、データ形式はFFT、平方根演算と同様である。

図7は、テーブル引きのインデックスの取り方を図に示したものである。インデックスは8ビットあるため、テーブル用メモリは256エントリ分用意している。このインデックスは、入力値が $2^{-3} \sim 2^4$ までの値を入力がとった場合に用いる。この範囲を越えた場合は、値に応じて出力を0度、90度、180度にする。

4. 評価

評価には、Pentium III 1GHz、512MB SDRAM搭載のPCにFPGAボードを実装したシステムを使用する。このシステムを用い、各処理をPC側で実行した場合、FPGAボード側で行った場合の実行時間を測定する。なお、FPGAの動作周波数は12MHzに設定している。

4.1 FFT

ここでは、2048ポイントのFFT演算の評価結果を示す。

FPGAボード側で処理した場合の評価は2通り行い、実機を用いてPC側からデータをFPGAボードに転送し、その処理が終了するまでの時間を測定したもの(実測)、シミュレーションで、FFT演算モジュールに演算開始信号を送ってから全ての演算が終了するまでのサイクル数を測定し、12MHzで動作させた場合の実行時間を測定したもの(Sim.)について行った。なお、シミュレーションによる結果は純粋な演算時間を表し、実測値にはデータをPCとボード間でPIOで転送するオーバーヘッドが含まれる。転送データサイズは1回の演算あたり32KB^(注3)である。

PC側のCPUで処理した場合の結果については、テーブル引きを行わずにFFTアルゴリズムをそのまま実装した方法(Base)、FPGAの実装と同様 sin、cosの値をあらかじめ計算してテーブル引きで行う方法(Table)を測定した。また、x86アーキテクチャ用の最適化や、その他の高速化手法を使用したライブラリである fftw ライブラリ^(注4)を用いた方法(FFT)につ

(注2)：浮動小数点フォーマットを元にしたインデックスのため、精度が粗い角度領域と密な角度領域が存在する。そのため最も粗い角度領域でこの精度を満たすようにする

(注3)：2048個の複素数データのため、平方根や ArcTan の2倍

(注4)：http://www.fftw.org/

表2 平方根演算の実行時間

実装方法	測定条件	実行時間
CPU	実測	497 μ s
FPGA	実測(PIO転送時間含む)	2589 μ s
FPGA	Sim.(本体の動作時間のみ)	172 μ s

(表中の sim. はシミュレーションを指す)

表3 ArcTanの実行時間

実装方法	測定条件	実行時間
CPU(Table)	実測	567 μ s
FPGA(Table)	実測(PIO転送時間含む)	2584 μ s
FPGA(Table)	Sim.(本体の動作時間のみ)	171 μ s
CPU(Base)	実測	736 μ s

(表中の sim. はシミュレーションを指す)

いても評価を行った。

CPU処理の時間測定、FPGA処理の実測値の測定には、gettimeofday関数を用いて処理前と後の時刻情報を取得し、その差を求める事により行った。

評価結果を表1に示す。この結果から、PC側でのテーブル引きFFTと比較して6%の高速化を、データ転送オーバーヘッドを含む12MHz動作のFPGA処理で達成可能である事が分かった。また、FFTモジュール本体の実行時間は、CPU処理に比べて2.9倍程度高速である。

ただし、最適化されたFFTライブラリを用いた場合、FPGAの実測値の1/10の時間で処理が完了している。今回実装したFFT演算回路は複素FFT演算をそのまま実装しているため、無駄な演算の削減等の最適化を行うことにより、更に高速な回路の実現を期待できる。

4.2 平方根

表2は、2048個の連続データの平方根演算を行った結果である。FFT同様、CPUによる処理時間、FPGAの実測値の測定にはgettimeofday関数を利用し、シミュレーションによる方法は演算に必要なサイクル数から時間を算出している。

この結果から、FPGAによる処理はCPUによる演算の2.9倍ほど高速である事が分かる。ただし、PC側とのPIOによるデータ転送(平方根演算では16KBの転送が必要)のオーバーヘッドが大きく、実測値ではCPUの5倍程度の時間を要する。

4.3 ArcTan

表3は連続する2048個のデータのArcTan演算を実行した結果である。CPUやFPGAの実測値、シミュレーションの評価方法等はこれまでと同様である。CPUによる処理は、FPGA同様テーブル引きを行うもの(Table)と、mathライブラリの級数展開による方法(Base)の2種類について行っている。

この結果、ArcTan演算をCPUで処理した場合、テーブル引きの方法でも平方根演算より13%程度処理時間が長くなっている。一方FPGAによる方法では、パイプライン構成をとるために実行時間の差はパイプライン段数分の差しか生じない。そのため、FPGAによる方法はCPUによる方法より3.3倍程度高速に処理を行うことができる。

ただし、こちらも全体の演算量は多くないため、データ(16K-

B)の転送オーバーヘッドは大きく、実測値ではCPUの4.5倍程度の時間を要する。

5. 検討及び今後の課題

FFT演算回路はCPU処理の2.9倍の速度を達成し、PIOによるPCIバス経由のデータ転送オーバーヘッドを含めて6%の高速化を達成しているが、fftwライブラリを使用したCPUによる処理は今回の回路の実測値より10倍程度高速であるため、あまり回路規模が大きくなならない範囲で無駄な演算の削減等の最適化を施し、より高速な回路の検討を行う必要がある。

平方根演算、ArcTan演算では、それぞれ2.9倍、3.3倍の高速化を達成しているが、PIOによるデータ転送のオーバーヘッドは、演算処理時間に比較して大きなものである事が分かった。

現在天野研究室では、Xilinx社のVirtex-IIを用いたボード[8]を開発しており、こちらのボードではDMA転送も行うことができるようになるため、オーバーヘッドの削減が期待される。ただし、今回実装した、一部機能のみのハードウェア化ではなく、より大きな単位をハードウェア化し、演算時間に比べて転送オーバーヘッドの割合が小さくなる規模での比較検討を行う必要がある。

また、今回用いたボードのFlex 10Kは現在では決して高速なデバイスとは言えないが、新しいボードのVirtex-IIではより高い動作周波数による動作が期待できるため、同じ実装であってもより高速な処理を行うことができると考えられる。

更に、今回の各回路の実装では内部でもIEEE754単精度浮動小数点フォーマットを基に処理を行っているが、必要精度が確保できれば固定小数点演算で代用できる可能性がある。その場合、より小型で高速な回路を構成できる可能性があるため、内部処理を変更した回路での比較検討も行う必要がある。

今回の実装は、比較的ハードウェア化が容易であると思われる処理の実装を行った。音源分離処理では、他にも様々な処理があり、ハードウェア化することによって高速化できる部分、ソフトウェア処理に適している部分があると思われる。今後はシステム全体での負荷分布の評価や、各処理のハードウェア化の可能性を含めて検討していきたいと考えている。

6. おわりに

本稿では、ロボット聴覚に用いる方向通過型フィルタに必要な処理のいくつかをFPGA上に実装し、CPUによる処理との比較を行った。

CPUによる処理速度と比較して、FFT演算回路では2.9倍、平方根演算で2.9倍、ArcTan演算で3.3倍の性能を達成した。この結果から、ハードウェア処理により低い動作周波数で十分な性能を確保できる可能性を確認することができたが、より効率のよい回路構成等、検討項目も多い。

今後は、その他の処理のハードウェアの実装や、異なる回路構成の検討、ロボット聴覚システムの各処理の負荷の評価やそのハードウェア化の可能性等を含めて検討する予定である。

- [1] 梶崎浩嗣, 黒川恭一, “FPGAを用いた暗号処理ボードの設計と試作”, 電子情報通信学会技術研究報告, VLD2001-126, CPSY2001-85, pp.31-38, 2002年1月
- [2] 中島雅美, 近藤弘都他, “FPGAプロトタイピングを活用した高性能マイクロコンピュータ”M32R”の開発” 電子情報通信学会技術研究報告, VLD2002, 2002年6月
- [3] Nakadai, K. Hidai, K. and Mizoguchi, H. and Okuno, H. G. and Kitano, H., “Real-Time Auditory and Visual Multiple-Object Tracking for Robots”, Proc. of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI-2001), pp.1424-1432, 2001
- [4] Nakadai, K. and Okuno, H. G. and Kitano, H., “Exploiting Auditory Fovea in Humanoid-Human Interaction”, Proc. of the Eighteenth National Conference on Artificial Intelligence (AAAI-2002), pp.431-438, 2002
- [5] Nakadai, K. and Okuno, H. G. and Kitano, H., “Epipolar Geometry Based Sound Localization and Extraction for Humanoid Audition”, Proc. of IEEE/RAS International conference on Intelligent Robots and Systems (IROS-2001), 2001
- [6] 高木直史, “算術演算回路のアルゴリズム - 3. 除算回路のアルゴリズム”, 情報処理, Vol. 37, No. 3, pp.280-286, 1996年3月
- [7] 高木直史, “算術演算回路のアルゴリズム - 4. 初等関数計算回路のアルゴリズム”, 情報処理, Vol. 37, No. 4, pp.362-368, 1996年4月
- [8] 長名 保範, 安生 健一朗, 天野 英晴, “Reconfigurable Hardwareによる細胞シミュレータの高速化手法” 情報処理学会 情報研報 ARC2002-148-8, 2002年5月