

リアルタイム時間圧縮再生システムの開発

荒井 幹哉† 幹 康†

拓殖大学工学部‡

1. はじめに

音は私たちの身の回りにある様々なものから発生する。私たちは音を聞いて周囲の状況を知る事ができるため、音は生活をする上で欠かせない存在という事ができる。しかし、音の中には人間の耳に聞こえない音(人間の耳に聞こえる音は可聴音と呼ばれ、その周波数帯域は 20Hz ~ 20000Hz である)も同時に存在する。

これら非可聴音帯域のうち、可聴音の帯域よりも低い周波数の物は超低周波音と呼ばれている。こうした音を耳で聞こうとする時、対象信号の周波数を可聴域になるように圧縮する必要がある。このとき、再生時に早送り再生する方法が存在するが、再生時間が短くなってしまい、音を聞き取る事が困難になってしまう。

そこで、この様な音を聞く方法として時間圧縮法が提案されている。この手法を用いる事で、元のデータと同じ再生時間を保持したまま音の周波数を高め、超低周波音を耳で聞くことが可能となる。

この手法は、補聴器や建築物の非破壊検査等への応用が考えられる。その様な場面において、リアルタイムに時間圧縮処理を行う事ができれば有効な道具として活用すると考えられる。

そこで本研究では、Windows OS 上で超低周波を観測し、リアルタイムで時間圧縮法を処理して再生するシステムの開発を行った。

2. システムの構成

2. 1. 時間圧縮法

時間圧縮法はあるデータに対し、一定長でデータを切り出し、切り出したデータに対して高速再生を行う。次に切り出す時に少しずつデータをずらしていく。データをずらしながら切り出し、連続的に再生することによって、経過時間を保持したまま周波数を高める事が可能になる。

例えばデータ数 100 点のデータがあるとする。このデータを 10 点ずつ切り出しをして高速再生させる。この時に再生させる区間を数点ずつ移動させ再生する。その移動する点数が 1 点分ならば 10 点に対して 1 点更新するため、圧縮率が 10 倍になっているという。すなわちこの元データが 10Hz の信号だったとした場合、圧縮後の信号は 10 倍の 100Hz として出力されることになる。

1 ~ 100 点のデータ

元のデータ

1~10	2~11	3~12	...	90~100
------	------	------	-----	--------

切り出したデータ

図 1 時間圧縮法の原理

2. 2. AD変換機

音声信号を PC に取り込む際には主にサウンドカードを使用するが、多くのサウンドカードは入力特性が可聴帯域以下の周波数に対応していない、もしくはデータ量を少なくするためにフィルタを通すため、目標としている超低周波そのものを検出する事ができない。

そこで、本研究ではハイパスフィルタの無い AD 変換機を用いて対象信号の取得を行う。変換機には National Instruments USB-6009 を使用した。この AD 変換機は USB ポートで接続し、データは USB インタフェースを通じて PC とやり取りするため、多くの PC への接続が容易な所も特徴である。

この AD 変換機は、C 言語において動作命令が用意されており、変換機の制御はこの動作命令で行う。この変換機は任意の点数データを取得してから、配列の形でプログラムに転送される。

2. 3. Win32API

処理を行ったデータは Wave 形式の音声信号としてスピーカーより再生する。この再生処理には Windows の一般的な API である Win32API を使用する。

Win32API は Windows が管理するオーディオ再生デバイスのハンドルを取得し、その再生キューにデータを書き込むことでプログラム内から Wave ファイルとして音を再生する事ができる。

2. 3. マルチバッファリング

再生されるデータ(Wave バッファ)は再生キューに蓄えられた後に再生される。一つのバッファの再生が終了すると、次のバッファを再生キューに書き込むまでの間に無音状態が発生する事になる。

この無音状態を回避するため、マルチバッファリングと呼ばれる手法を用いる。これは Wave バッファを 2 つ以上用意して再生キューに書き込む。片方のバッファの再生が終了すると、そのまま次のバッファが再生されるため、2 つ目のバッファの再生が終了する前に 1 つ目のバッファを更新し、再度書き込みを行う。この作業を

Development of system that reproduces by in real time compressing

† Mikiya Arai, Yasushi Miki

‡ Faculty of Engineering, Takushoku University

繰り返す事により、図2の様に音声途切れることなく再生し続ける事ができる。

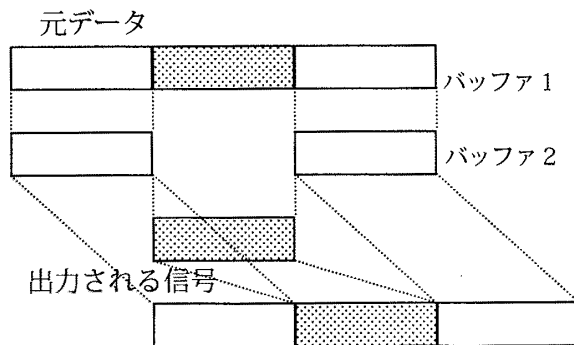


図2 マルチバッファリングの概念図

2. 4. 全体の構成

システム全体は以下のモジュールで構成する。

- ① タイマー
データ取得命令を出す際の間隔を任意時間のタイマーで処理タイミングを管理する。
- ② 取得
A/D変換機で信号データの取得を実行する。このA/D変換機は、1回のデータ取得で任意のサンプル数を取得できるが、処理の高速性を重視して1回の取得命令につき1点のデータを取得させる。
また、処理の高速化を測るため、サンプルが10点ずつ圧縮再生モジュールに転送される。
- ③ 圧縮再生
取得モジュールから渡された取得データを元に、信号に圧縮処理を行う。また、圧縮したデータを Wave バッファとして再生キューに書き込むことも行う。
- ④ A/D変換機
取得部から渡された取得時のデバイス設定を元に、アナログ信号からサンプルデータを取得する。
- ⑤ 出力デバイス
圧縮された入力信号を出力する。PCに付属のオーディオ出力デバイスを使用する。
- ⑥ 更新時間変更
ユーザからの入力を受け取り、データを取得する間隔を任意に変更させることができる。

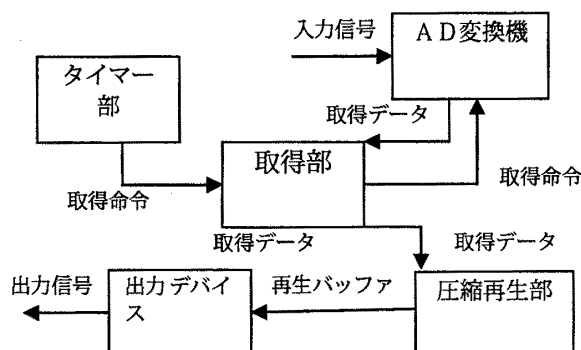


図3 時間圧縮再生システム全体の構成図

2. 5. 動作の手順

動作はタイマー部から任意の間隔で送られてくる取得

命令によって開始される。

取得部は取得命令が送られると、A/D変換機にデータ取得を行わせる。取得したデータが一定の個数取得すると、それを配列として圧縮再生部に渡す。圧縮再生部はデータを更新し圧縮処理を行った後、バッファを出力デバイスへ転送する。この時、バッファの更新時に無音を出さないようにマルチバッファリングを用いて音響信号として再生させる。以上の動作を繰り返し実行する事により圧縮再生を継続する。

また、取得間隔は実行中でも変更でき、バッファの再生速度はこの取得間隔と連動して変更され、データ更新と再生が等間隔で実行される。

3. システムの実行

作成したシステムの動作を確認するために動作実験を行った。切り出しサンプル数200点、サンプリング周期50Hz、出力時のサンプリング周波数400Hz、入力信号に5Hzの正弦波を入力し、20倍に圧縮して出力を行った。結果を図4、図5に示す。

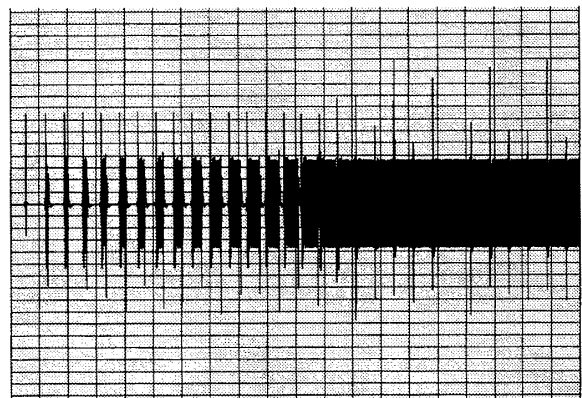


図4 圧縮結果 (横軸0~20s, 縦軸位相)

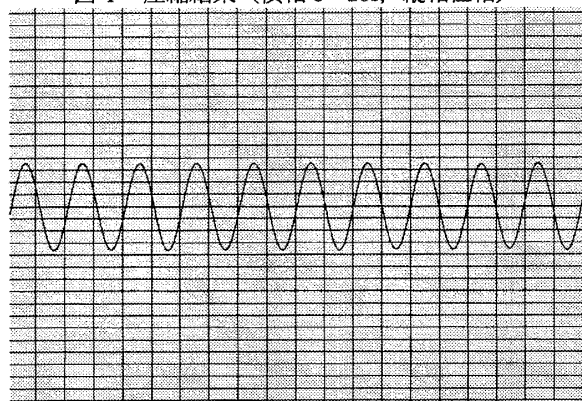


図5 圧縮結果の一部 (横軸0~0.1s, 縦軸位相)

以上の結果から、入力時5Hzだった信号は20倍の1000Hzに圧縮されている事が確認できた。

4. おわりに

本論では WindowsOS上でのリアルタイム時間圧縮再生システムを開発した。また、A/D変換機を取得部分に採用することにより、直流成分の電圧も入力信号として検出できるようになった。