

## 1 タイトル

千葉県立千葉工業高等学校から参りました、渡邊陽平です。

これから、私の研究「ソフトウェアシンセサイザーの制作」について発表します。よろしくおねがいします。

## 2 目次

今回の発表では、本研究の概要、システムの設計と仕様、実装のプロセス、実装予定の機能の順で話したいと思います。

## 3 制作の背景と目的

最初に、なぜ私がこの研究内容を選んだかと言うと、現代の音楽制作において、シンセサイザーは不可欠なツールとなっているものの、その構造は謎に包まれている部分が大きく、ブラックボックス化しているという現状があります。

本研究ではシンセサイザー制作を通してその構造や音響合成の仕組みを理解することを目的とします。

## 4 システムの設計と仕様 (1)

本プロジェクトに使用したソフトウェアは次の通りです。

音声の出力には C 言語を使用しました。音声出力に限らず、IO に関するライブラリが豊富であるため、採用しました。

音声の出力を担うサウンドサーバは PulseAudio を採用しました。

音響合成には Fortran 2003 を採用しました。FORTRAN は 1954 年に IBM のジョン・バックス氏が考案した世界最初の高水準言語であり、科学技術計算で使われてきた歴史があり、コンパイル時の最適化技術自体はこの言語から始まりました。

Fortran は何度もバージョンアップを繰り返していますが、今回採用した Fortran 2003 では今回の制作の肝になる ISO\_C\_BINDING モジュールが追加されました。

## 5 システムの設計と仕様 (2)

処理のフローチャートは図の通りです。C 言語による pulseaudio の起動と音声出力を除き、殆どの処理を Fortran が担っています。

## 6 システムの設計と仕様 (3)

減算式シンセサイザーの基本的な構造は図の通りです。オシレータで生成した波形をフィルターで音色を変え、アンプで全体の音量を調節します。エンベロープは鍵が押されてから、離されてからの時間で出力が変化します。LFO は連続的に出力が変化し、演奏技法の再現などに用います。

## 7 実装のプロセス (1)

オシレータの実装の例として、矩形波を出力する関数です。

引数の少数部分を  $I$  とし、これが 0.5 よりも小さいならば 1、大きいならば -1 が出力されるという非常にシンプルな構造となっております。

こういった出力が周期的な関数は三角関数に習って一周を  $2\pi$  とするのが礼儀というものですが、後のコーディング時の便宜を図るため一周を 1 としています。

## 8 実装のプロセス (2)

C 言語と Fortran の連携には ISO\_C\_BINDING を使用しました。これにより、C 言語で定義された関数を Fortran で関数やサブルーチンと呼び出すことが可能になります。

## 9 実装予定の機能 (1)

ここからはまだ実装していない機能について解説していきます。

フィルターはオシレータで作られた音波から不要な周波数成分をカットし、音色を変化させる機能です。

本研究では、アナログ回路のシミュレーションを利用して実装を目指します。

## 10 実装予定の機能 (2)

LFO は出力の大きさを周期的に変化させるモジュールです。

日本語に訳すと低周波発振器となりますが、音のもとになる波を作り出すオシレータよりも低い周波数で出力を変化させます。

これをアンプにつなぐことによりトレモロ、オシレータのピッチにつなぐことによりピブラートが再現できます。

## 11 実装予定の機能 (3)

サンプリング周波数通減モジュールは、私が独自に考案したモジュールです。

ファミリーコンピュータから出力される波形は、左のような滑らかな波ではなく、右のようにガタガタ下波形であることが知られており、これは特に三角波において顕著です。

このモジュールはそれを擬似的に再現することにより、レトロゲームのようなサウンドを再現します。

まだまだ実装できていない機能はありますが、それら全ての実装に向けて努力を続けたいと思います。

ここからはご質問を受け付けたいと思います。

ご清聴ありがとうございました。