

ソフトウェアシンセサイザーの制作

渡邊 陽平

October 17, 2025

- 制作の背景と目的
- システムの設計と仕様
- 実装のプロセス
- 主な工夫
- 実装予定の機能

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。

しかし、**ブラックボックス化しているという現状。**

本研究ではシンセサイザーの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。

システムの設計と仕様 (1)

音声出力部

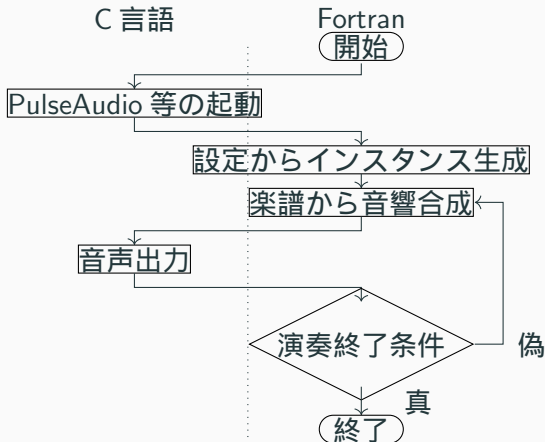
- 規格:C17
- 役割:pcm 情報を音声としてスピーカーから出力する。
- 採用理由: 入出力に関するライブラリが豊富であるため採用。
- 使用ライブラリ:PulseAudio クライアントライブラリ 16.1
(音声出力)

音響合成部

- 使用言語:Fortran 2003
- 役割: 音響合成を行う。
- 採用理由: 信号処理に最適化された言語特性から採用した。
- 使用モジュール:ISO_C_BINDING(C 言語との連携)

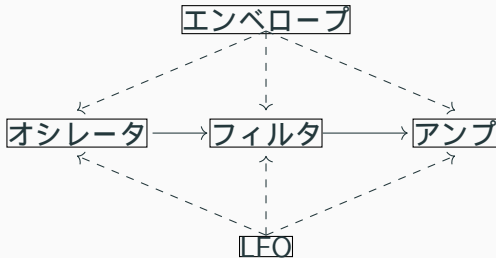
システムの設計と仕様 (2)

システムのフローチャート



システムの設計と仕様 (3)

減算式シンセサイザーの構造



実装のプロセス (1)

オシレータの実装

例: 矩形波生成の Fortran コード

```
IMPLICIT NONE
PURE FUNCTION OSC\_SQUARE(X)
  REAL, INTENT(IN)::X
  REAL::OSC\_SQUARE

  REAL::I
  I = X-AINT(X)
  IF(I < 0.5) THEN
    OSC\_SQUARE=1
  ELSE
    OSC\_SQUARE=-1
  END IF
END FUNCTION OSC\_SQUARE
```

後のコーディング時の便宜を図るため、一周を 2π ではなく 1 としている。

実装のプロセス (2)

C 言語と Fortran の連携

ISO_C_BINDING モジュールを用いることで C 言語で定義された手続きを Fortran から呼び出すことが可能になる。例:

```
int hoge_in_c(int n, int m){  
    return x;  
}  
  
USE ISO_C_BINDING  
FUNCTION HOGE_IN_F(N, M) BIND(C,&  
    &NAME='hoge_in_c')  
    INTEGER(C_INT), INTENT(in)::N,M  
    INTEGER(C_INT), INTENT(out)::HOGE_IN_F  
END FUNCTION HOGE_IN_F
```

このように C の手続きを Fortran で参照することにより、この場合であれば HOGE_IN_F として呼び出すことが出来る。

void 型の関数を参照する場合は、関数としてではなくサブルーチンとして参照する。

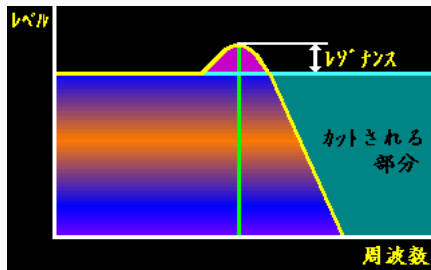
実装予定の機能 (1)

フィルター

減算式シンセサイザーでは、オシレータの音から不要な周波数成分をカットし、音色を変化させる。

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す。

例: ローパスフィルタ

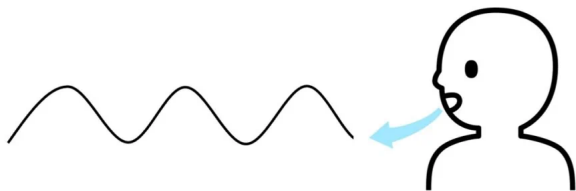


実装予定の機能 (2)

LFO

出力の大きさを周期的に変化させる。

アンプに繋がればトレモロ、オシレータのピッチに繋がればビブラートが再現できる。



音に周期的な揺れを作って心地よさを演出する

©弾き語りすとLABO 2022

(出典:<https://hikigatarisuto-labo.jp/vibrato/>)

実装予定の機能 (3)

サンプリング周波数逓減モジュール

あえて波形をガタガタにすることで、レトロゲームのような効果を演出する。

