MeRL Stands for Microelectronics Research Lab.

図形と音声の変換手法とその応用に関する研究

集積回路工学研究室

岩淵 勇樹

主任指導教員名: 秋田純一教授

第1章

序論

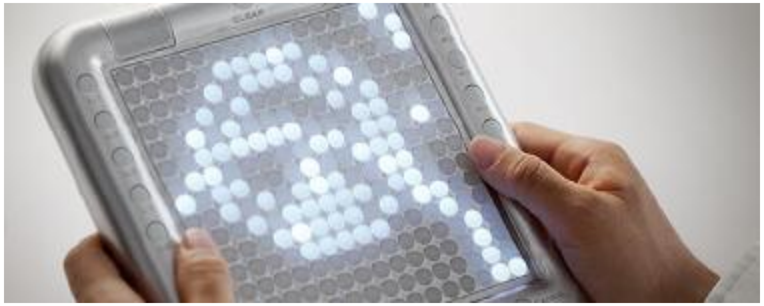
金沢大学 KANAZAWA UNIVERSITY

MeRL Stands for Microelectronics Research Lab.

背景

- 音楽制作における電子楽器の浸透
- 電子楽器市場に新しい波

次世代の電子楽器(1)



TENORI-ON
2008年5月発売
開発元: ヤマハ

- 視覚的・直感的な作曲が可能
- 16×16のボタンで平面的な操作

http://www.yamaha.co.jp/

次世代の電子楽器(2)

KAOSSILATOR
2007年11月発売
開発元: KORG

- タッチパッド操作による演奏
- 横軸・縦軸で独立したパラメータ操作

http://www.korg.co.jp/

ゲーム機で楽しむ音楽

KORG DS-10
2008年7月発売
開発元: AQインタラクティブ

- KORG MS-10をニンテンドーDSで再現
- シンセサイザーに加え、カオスパッド入力なども可能
- ゲーム的要素は含まれない

http://www.aqi.co.jp/

CGMにおける音楽

- VOCALOID（ボーカル音声の合成ソフトウェア）により一般人が制作した音楽作品が急増

http://www.crypton.co.jp/mp/pages/prod/vocaloid/cv01.jsp

電子楽器の現状

- CGM時代における楽器の新たな需要
- 「作曲」「演奏」では数々のイノベーション
- では、音色作りは？

「音色」の入力インタフェース

- 主にツマミやボタン･スライダ
- 自由な音作りには慣れが必要

タッチパネル製品の普及

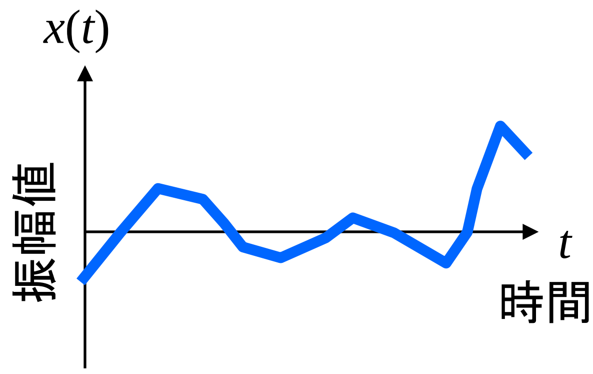
- スマートフォン（iPhone、Android端末）
- タブレット端末（iPad等）
- ゲーム機（ニンテンドーDS、Wii U、PlayStation Vita）
- タンジブルユーザインタフェース

⇓

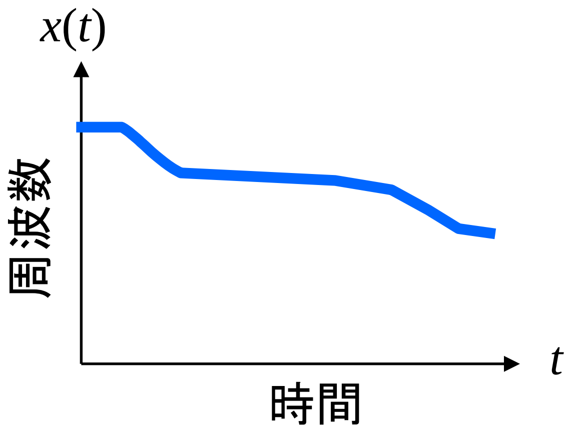
タッチ操作に最適な新しいインタフェースが必要

画像を用いた変換音楽

例）グラフ



「波形編集」
Audacityなど



「スペクトログラム」
UPIC、MetaSynth、MIDIアニメなど

- ほとんどは各軸の次元が違う
- 図形的特徴に忠実なシステムは皆無

アウトライン

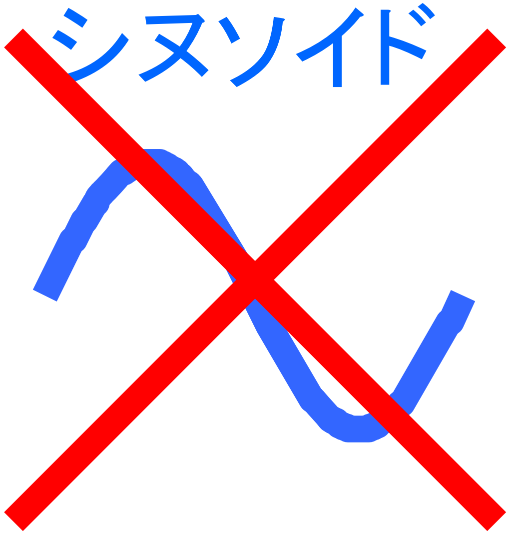
- 音声信号を可視化
- 閉曲線図形から音色を生成
- 平面的な入力インタフェースを生かした音色入力の方法を提案

第2章

回転によって
音色が不変である信号

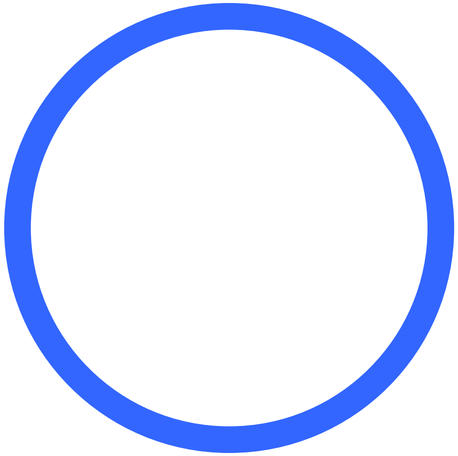
音のかたちとは？
（発想の原点）

正弦波＝



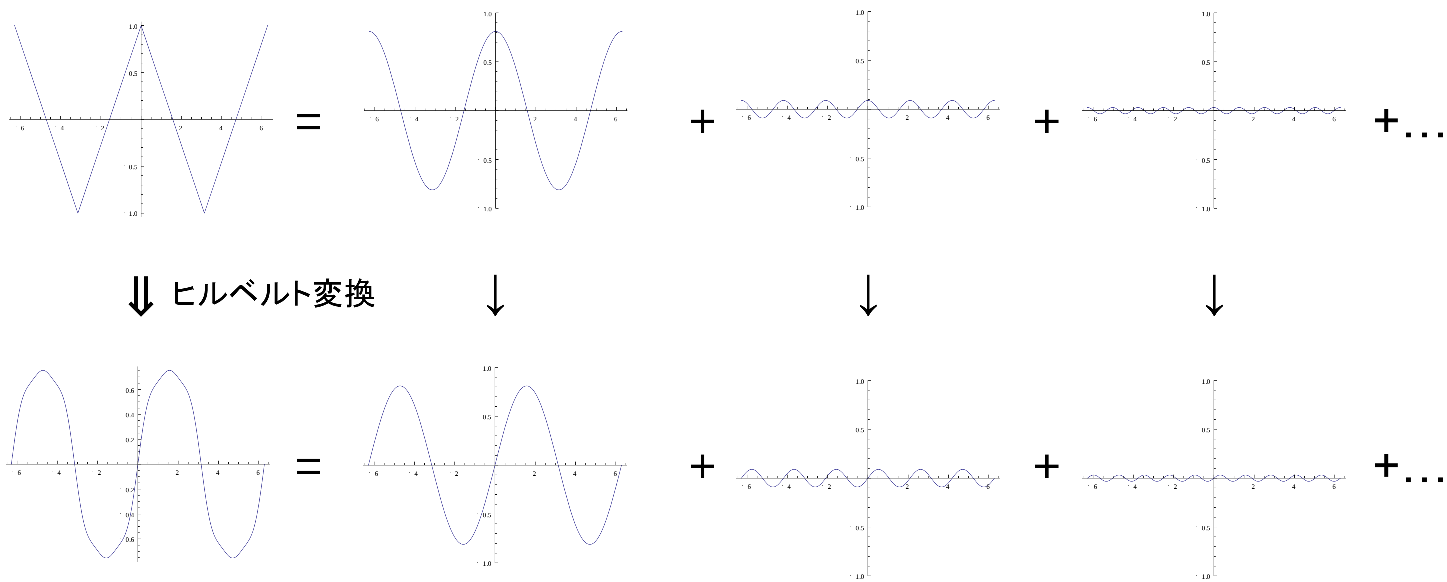
音のかたちとは？（発想の原点）

真円

正弦波＝ 

ヒルベルト変換

音声信号
（例: 三角波）



1. 音声信号を三角関数（正弦波）に分解
2. 位相を1/4周期（90度）ずらす（cosをsinに）
3. 分解した正弦波を足し合わせる

⇒ヒルベルト変換

ヒルベルト変換

$$H(\omega)=\begin{cases} i & (\omega<0) \\ 0 & (\omega=0) \\ -i & (\omega>0) \end{cases}$$

- オールパスフィルタ
- 負周波数で90°、正周波数で–90°位相変化

解析信号

- $\tilde{s}(t) = s_x(t) + i\, s_y(t)$

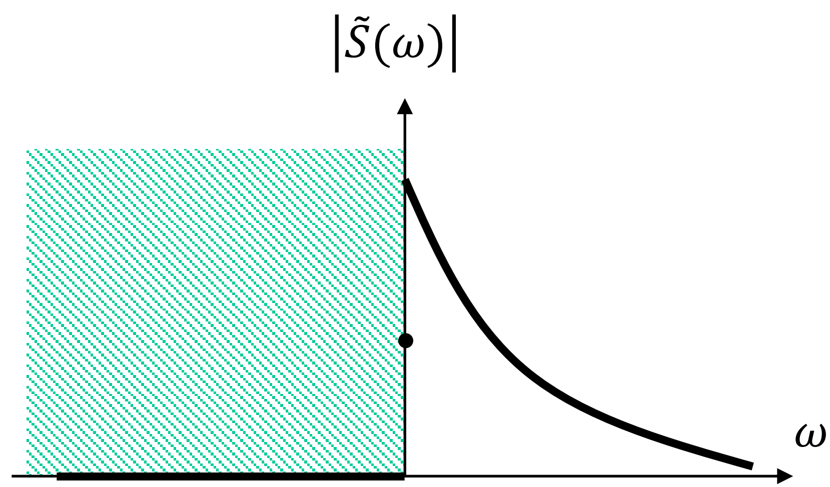
$s_y(t) = \mathcal{H}[s_x(t)]$（ヒルベルト変換）のとき

$$\tilde{S}(\omega) = 2U(\omega) \cdot S_x(\omega)$$

→<u>一般に「解析信号」とよばれる</u>

- $\Re[\tilde{s}(t)] = s_x(t)$
- $\tilde{s}(t) = \left(\delta(t) + \frac{i}{\pi t}\right) * s_x(t)$
- $= \underbrace{s_x(t)}_{\text{実部}} + i \underbrace{\left(\frac{1}{\pi t} * s_x(t)\right)}_{\text{虚部}}$

ヒルベルト変換



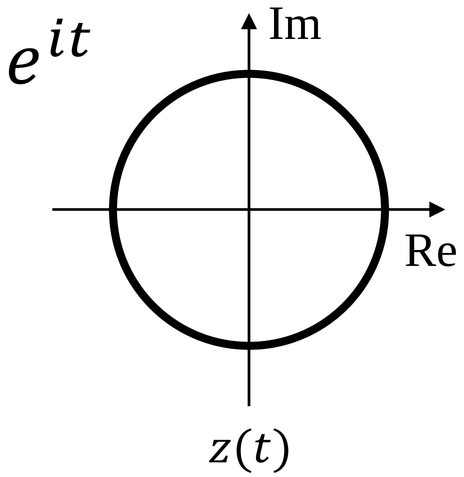
解析信号の例

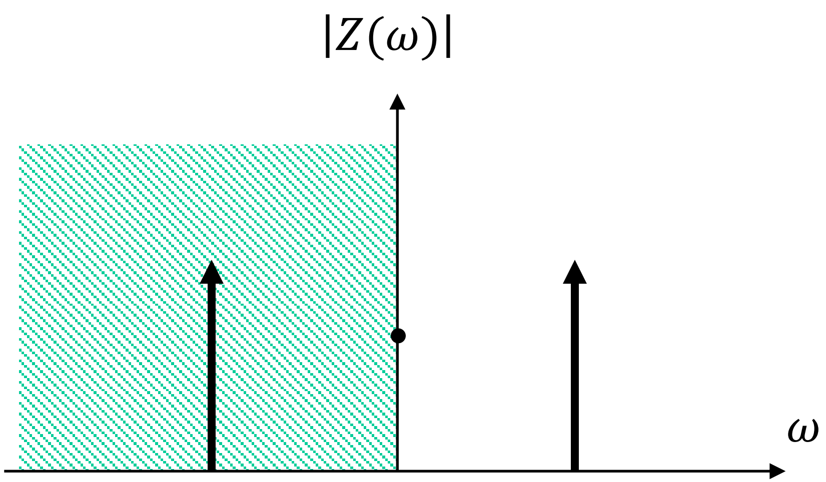
例）

$$e^{i\,t} = \cos(t) + i\sin(t)$$

$$\sin(t) = \mathcal{H}[\cos(t)]$$

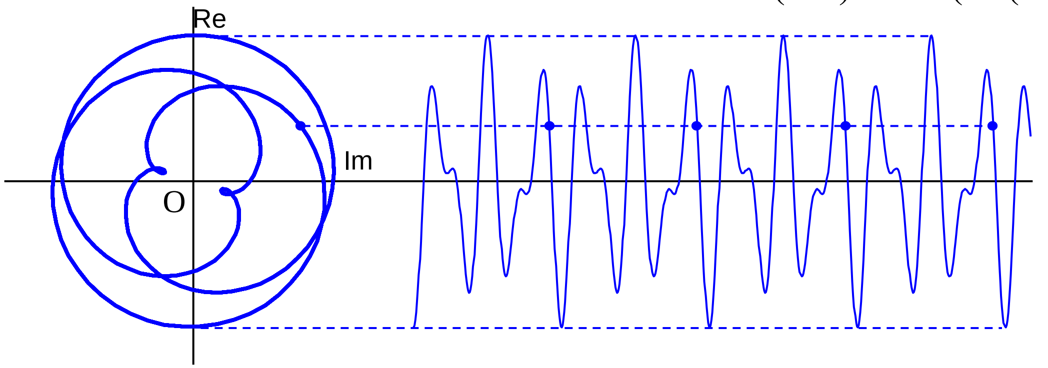
$$\delta(\omega - \omega_0) = 2U(\omega)\,\frac{\delta(\omega - \omega_0) + \delta(\omega + \omega_0)}{2}$$

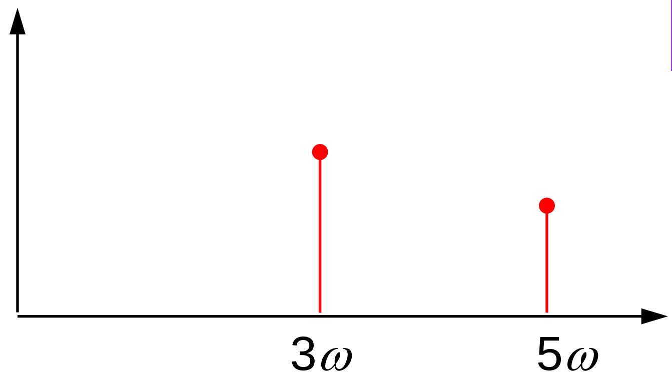




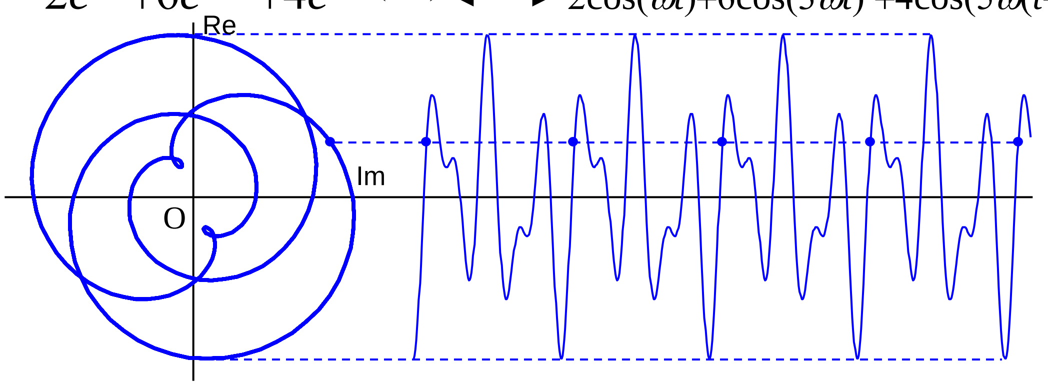
解析信号の例

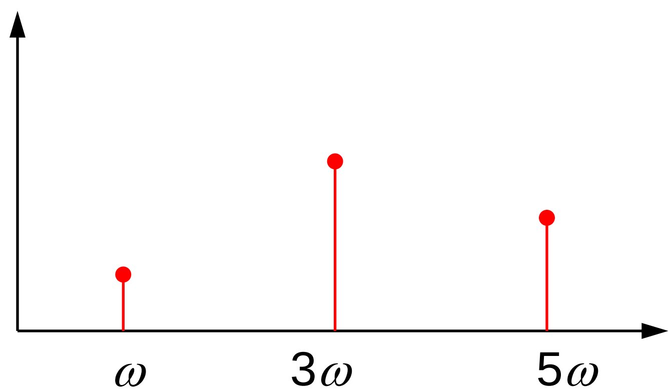
$$6e^{i3\omega t}+4e^{i5\omega(t-\theta)} \longleftrightarrow 6\cos(3\omega t)+4\cos(5\omega(t-\theta))$$



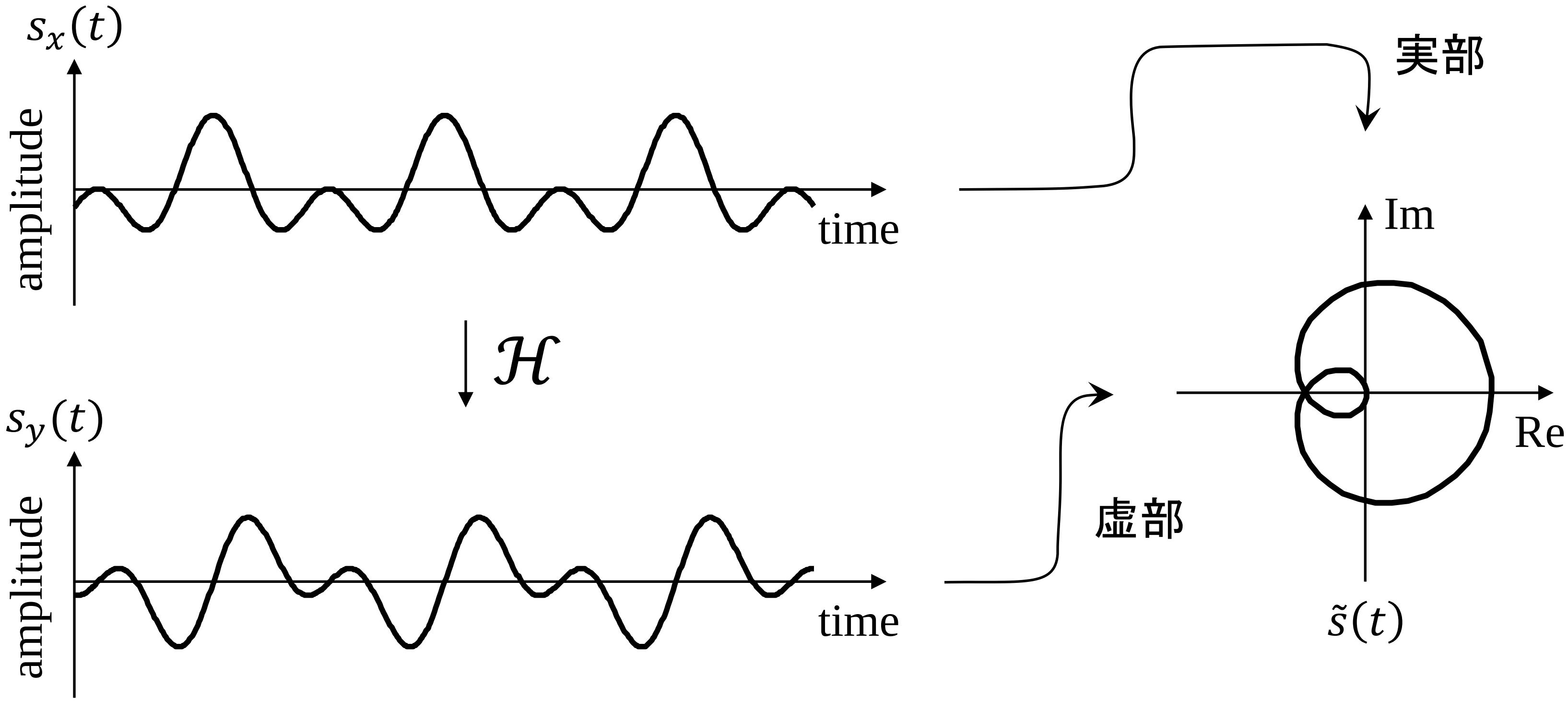


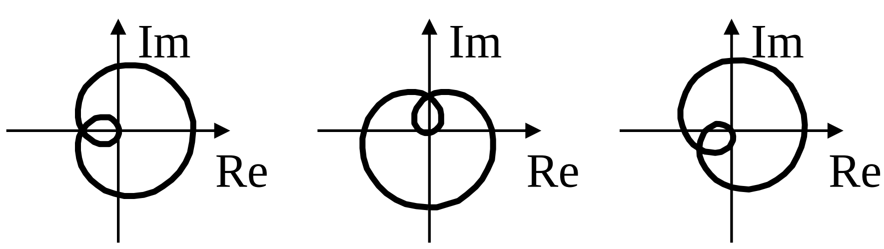
$$2e^{i\omega t}+6e^{i3\omega t}+4e^{i5\omega(t-\theta)} \longleftrightarrow 2\cos(\omega t)+6\cos(3\omega t)+4\cos(5\omega(t-\theta))$$





解析信号の特徴





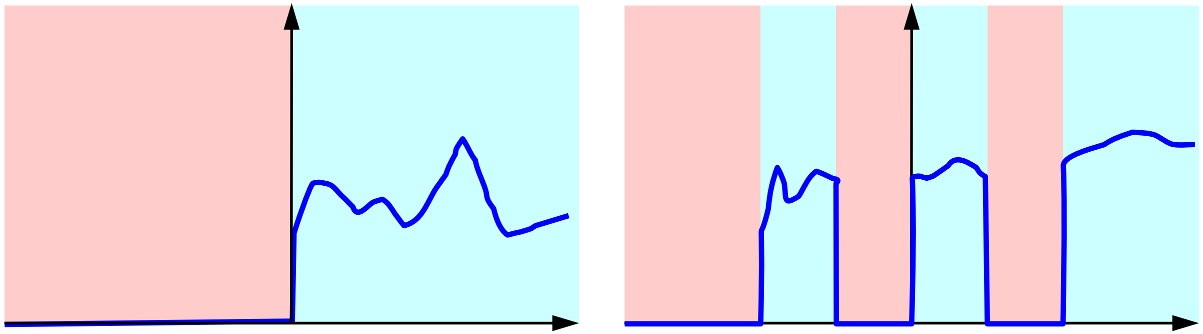
回転図形に対して
実数部のパワースペクトルが
常に等しい

解析信号からの拡張

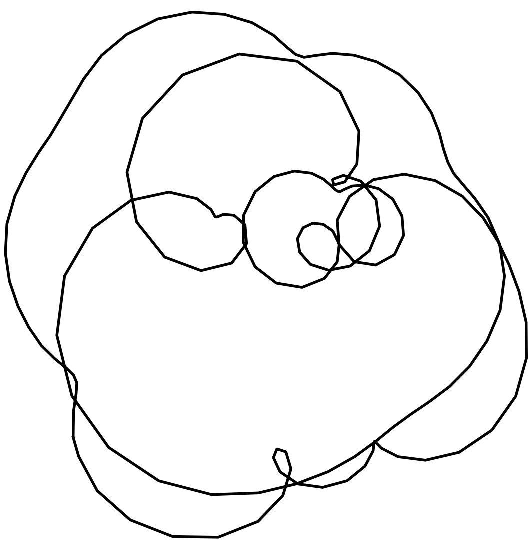
- 解析信号は任意の形状を表現できるか⇒？
- 「回転しても実部のパワースペクトルが常に等しい」という条件を拡張

⇒周波数毎に正か負どちらかのみの周波数成分をもつ

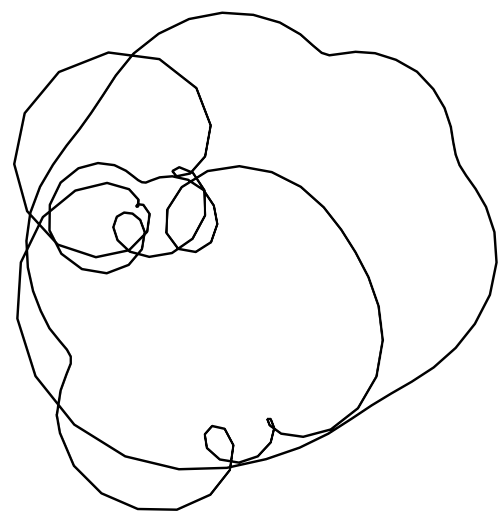
- 周期信号なら級数$\sum_{n=1}^{\infty} a_n e^{s_n i n \omega_0 t}$で表現可
  （$a_n \in \mathbf{Z}, s_n \in \{-1,1\}$）



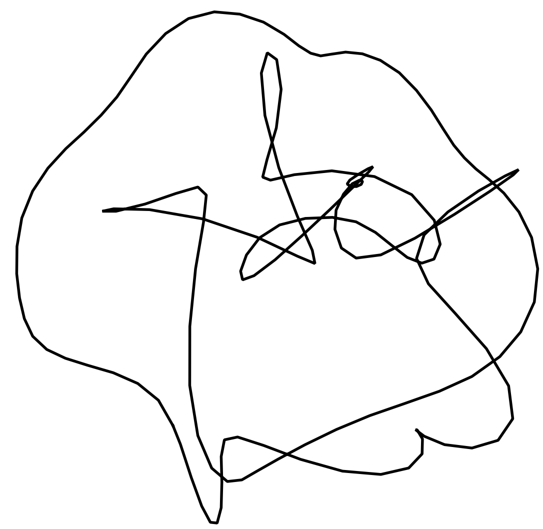
解析信号からの拡張



解析信号



拡張した解析信号の例1



拡張した解析信号の例2

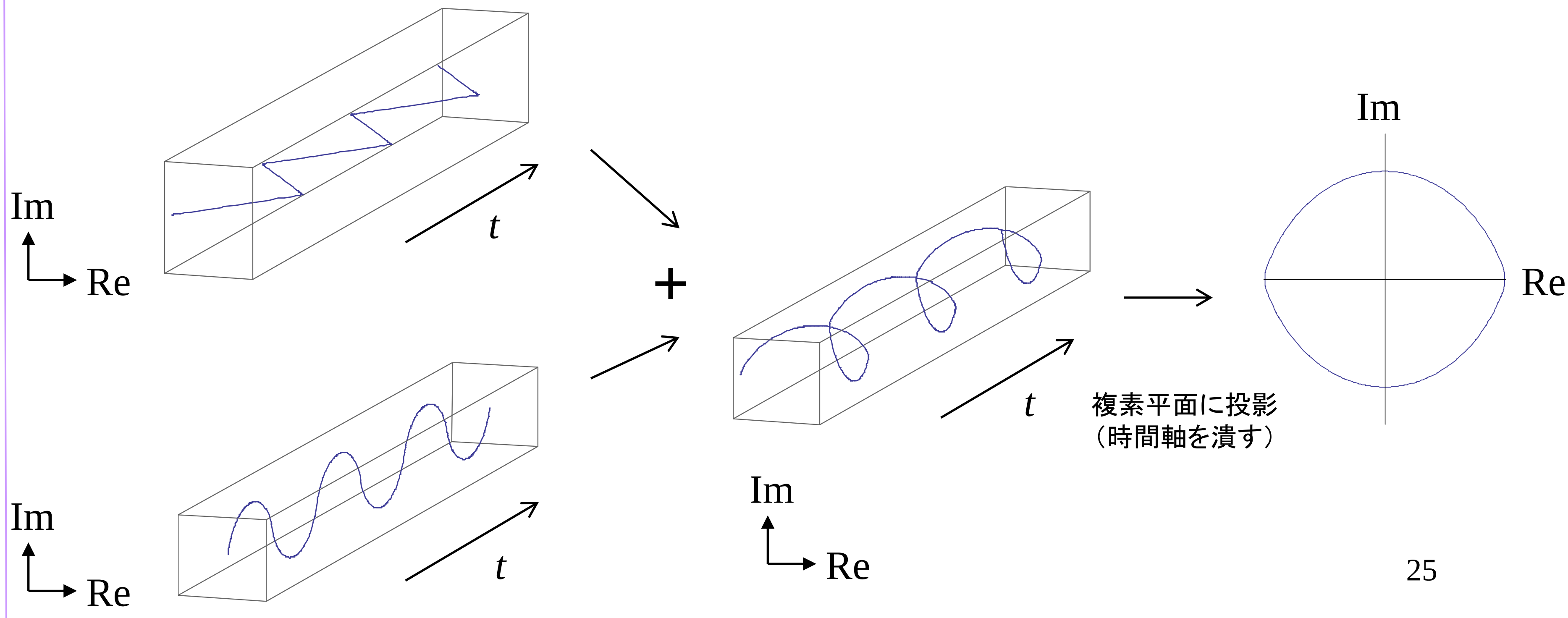
第3章

音声信号から図形への変換

音声信号の複素化

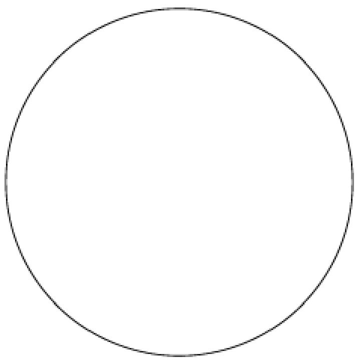
実数部: 元の音声信号

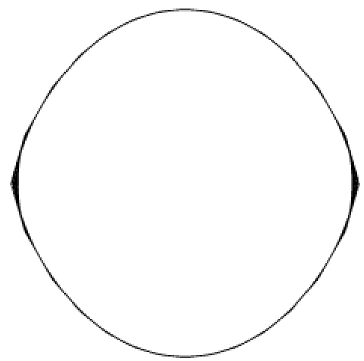
虚数部: ヒルベルト変換した音声信号

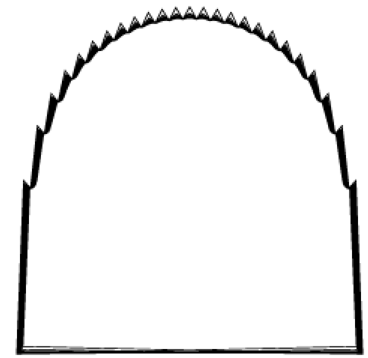


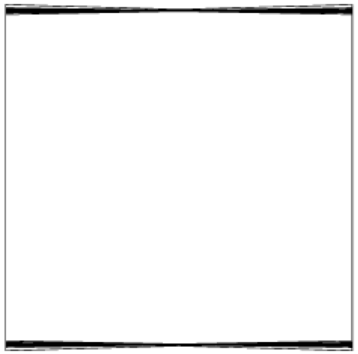
音声信号⇒図形

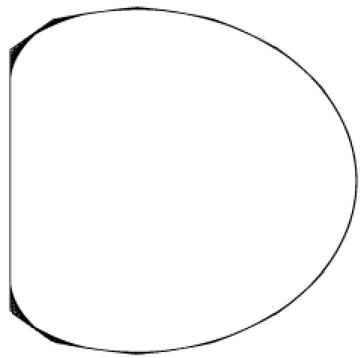
プリミティブな音

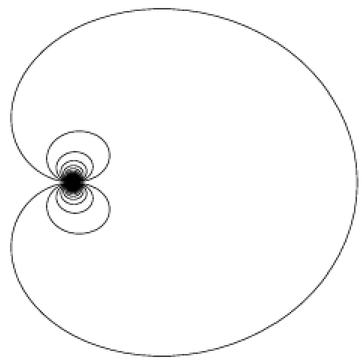

正弦波


三角波


のこぎり波

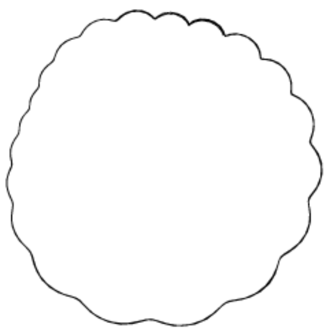

矩形波


正弦波半波整流波

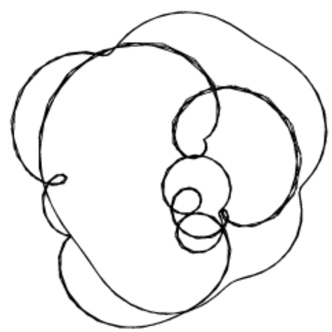

Sinc関数

音声信号⇒図形

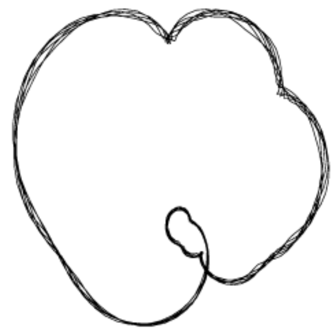
MIDI音（Microsoft GS Wavetable SW Synth）



電子ピアノ

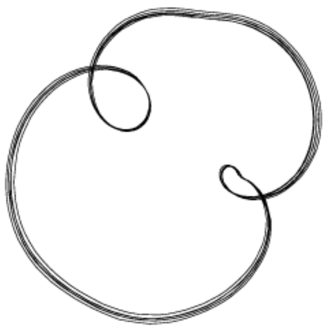


ハーモニカ



ナイロンギター

ヴァイオリン



オーケストラハープ

ストリングス

コーラス

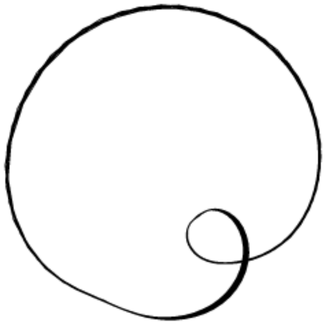
のこぎり波

音声信号⇒図形

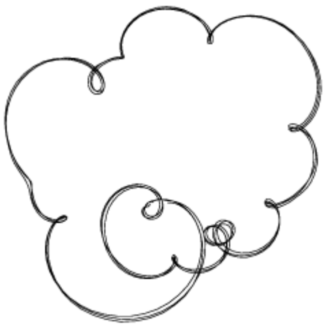
生楽器

ピアノ

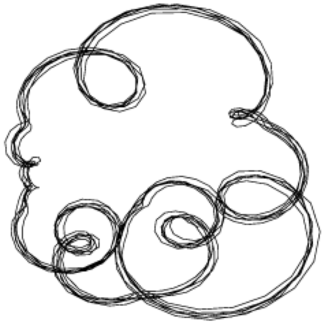
ピアノ（高）



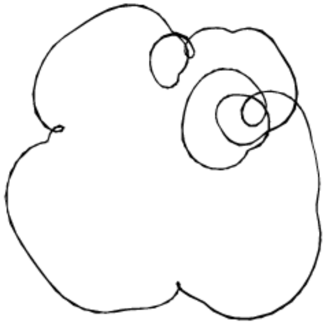
トランペット



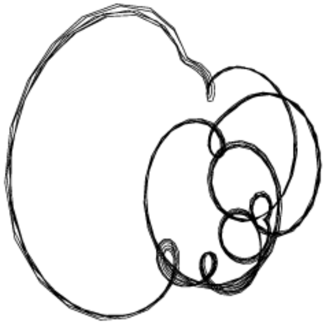
ギター



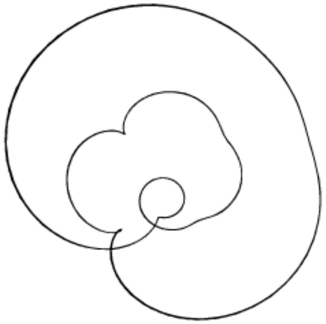
ギター（高）



ヴァイオリン



ハーモニカ

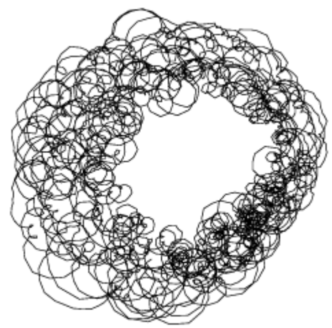


三味線

音声信号⇒図形

声

あ



い

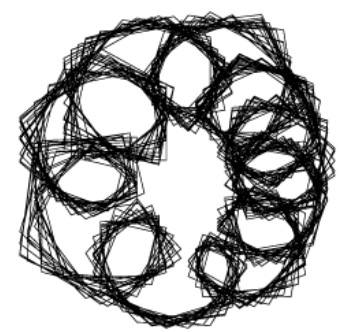
う

え

お

口笛

ホーミー1



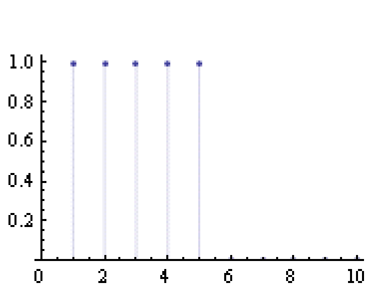
ホーミー2

図形がもつ特徴と音声信号の関わり

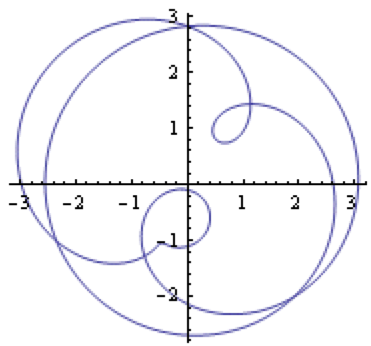
- スペクトルの違いによって図形的な特徴(回転数、曲線の長さ等)がどう変わるか
- 低周波数成分を多く含む信号と高周波数成分を多く含む信号を100ずつ比較

意図的な解析信号

- 低周波

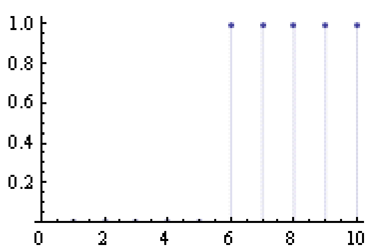


スペクトル

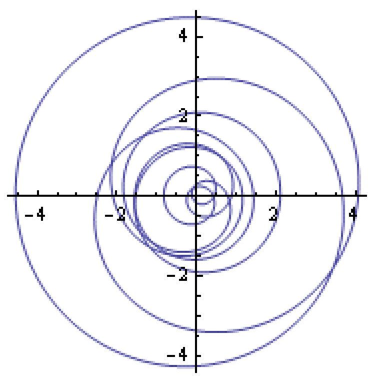


解析信号

- 高周波



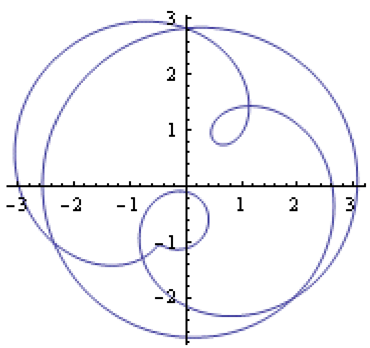
スペクトル



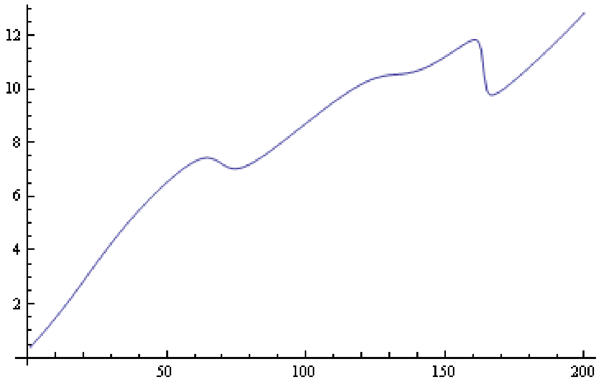
解析信号

- スペクトルの絶対値が一定
- 位相がランダム

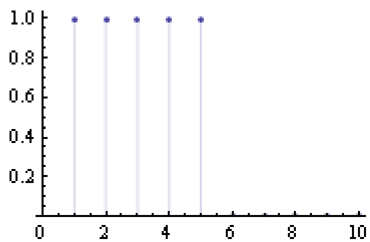
低周波の例



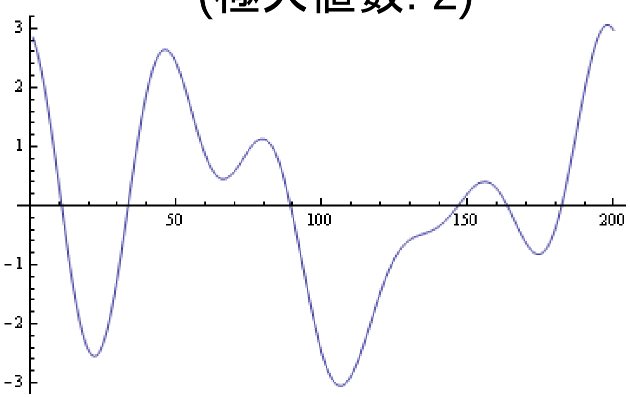
解析信号
(回転数: 5)
(交点数: 5)



偏角
(極大値数: 2)

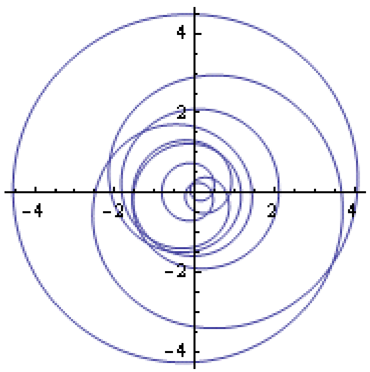


スペクトル

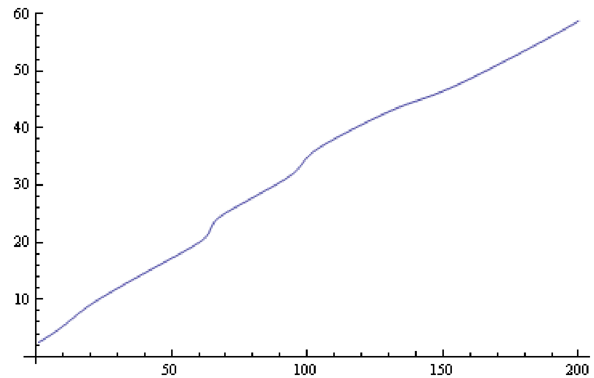


実信号
(極大値数: 4)

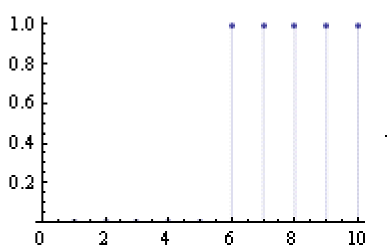
高周波の例



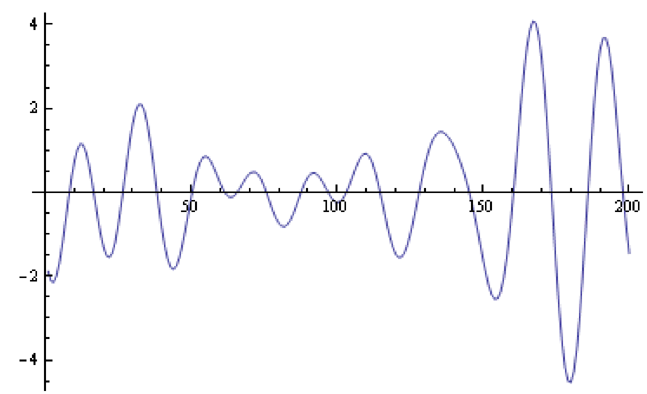
解析信号
(回転数: 9)
(交点数: 10)



偏角
(極大値数: 0)



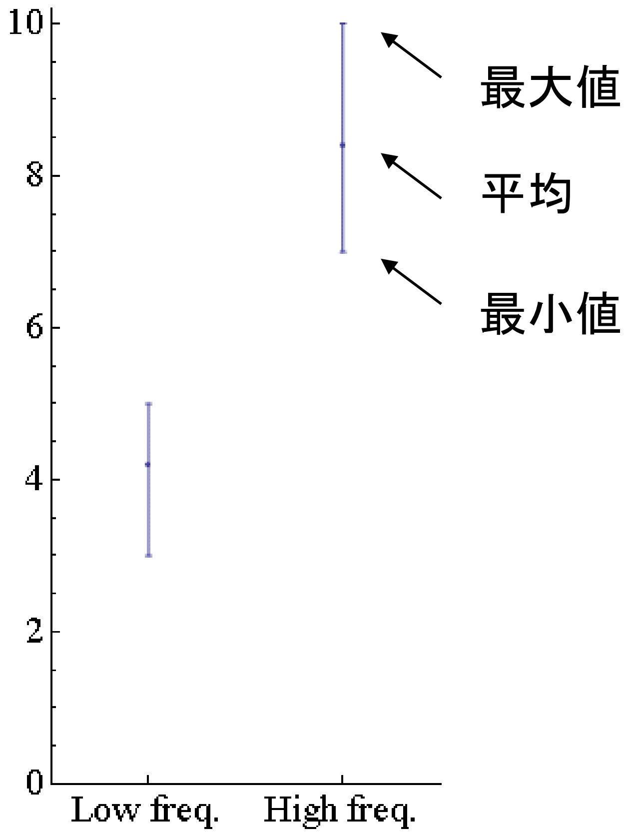
スペクトル



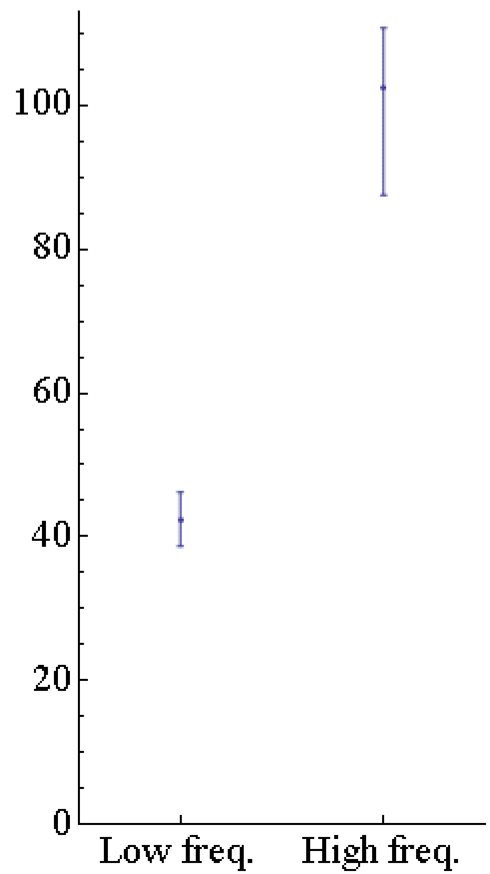
実信号
(極大値数: 9)

結果

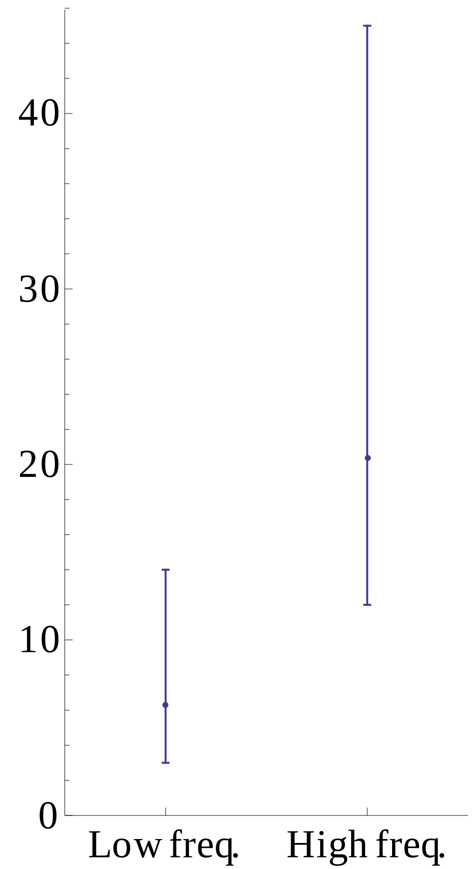
- 回転数



- 曲線の長さ

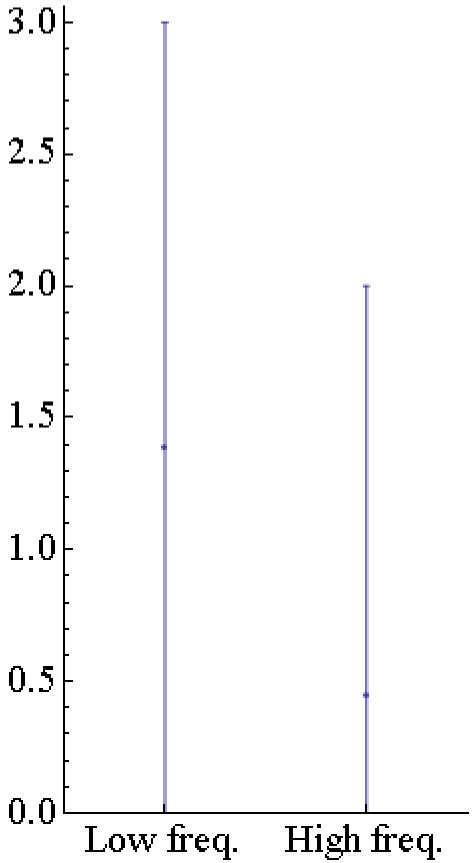


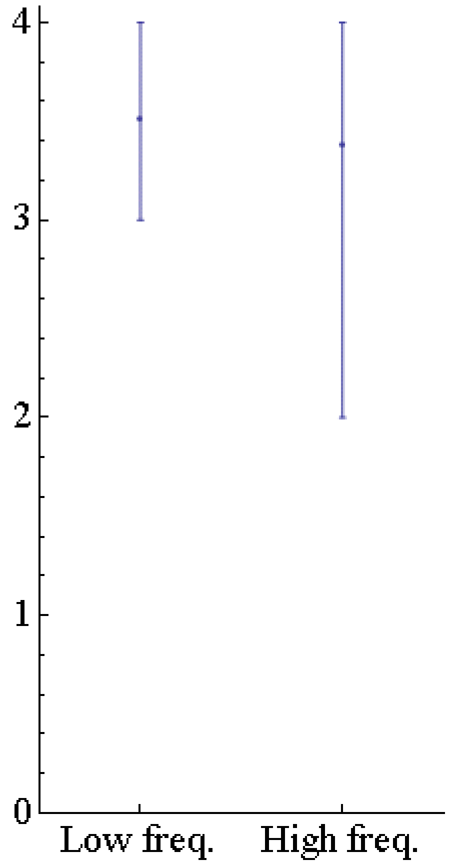
- 交点数

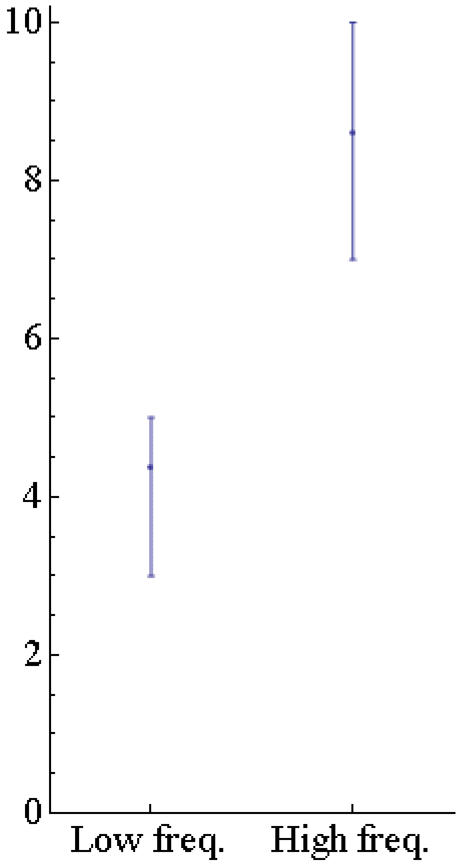


結果

- 偏角の極大値数
- 絶対値の極大値数
- 実信号の極大値数







第4章

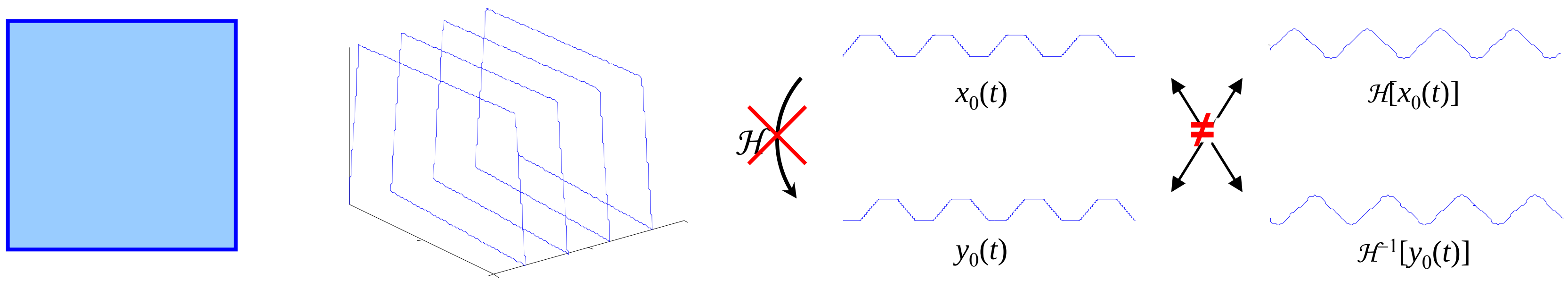
図形から音声信号への変換

「形⇒音」の変換

- 「音⇒形」の逆を考える
- パラメータ変換により、与えられた曲線を解析信号に近づける

境界追跡による擬似解析信号

シルエット図形　　　境界追跡信号



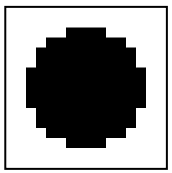
$$\tilde{s}_0(t) = x_0(t) + i\, y_0(t)$$

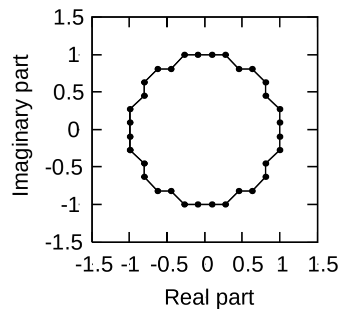
$$\xi(t) = x(t) + i\, \mathcal{H}[x(t)]$$
$$\eta(t) = \mathcal{H}^{-1}[y(t)] + i\, y(t)$$

$\tilde{s}(t)$が解析信号なら$\xi(t) = \eta(t)$

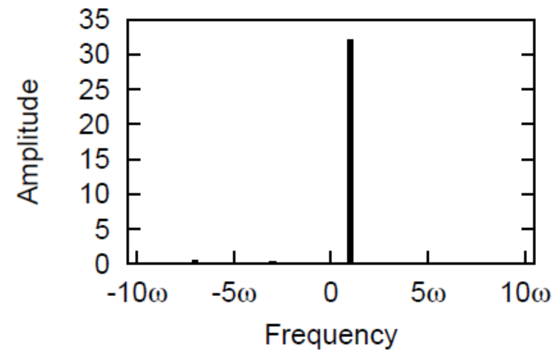
入力画像と輪郭線

画像1

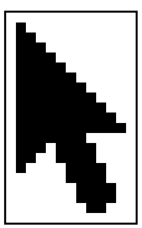


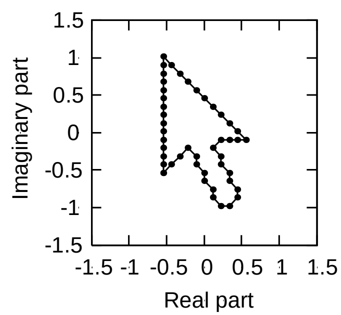


$\xrightarrow{\mathcal{F}}$

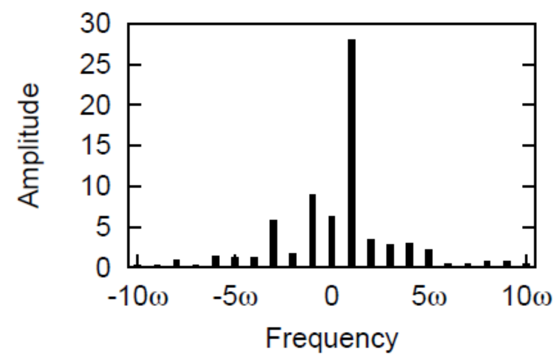


画像2

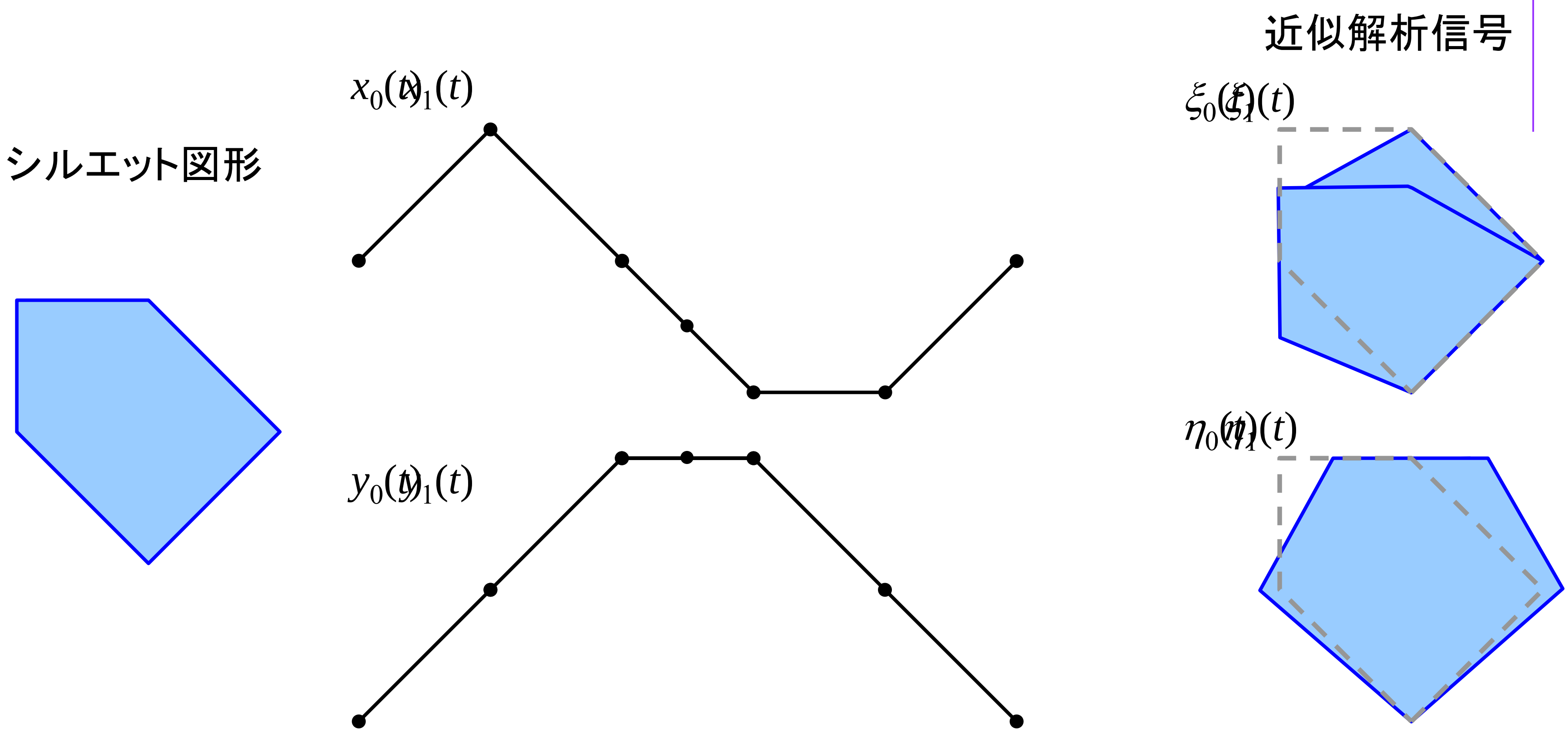




$\xrightarrow{\mathcal{F}}$

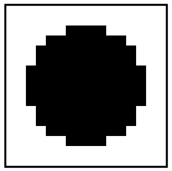


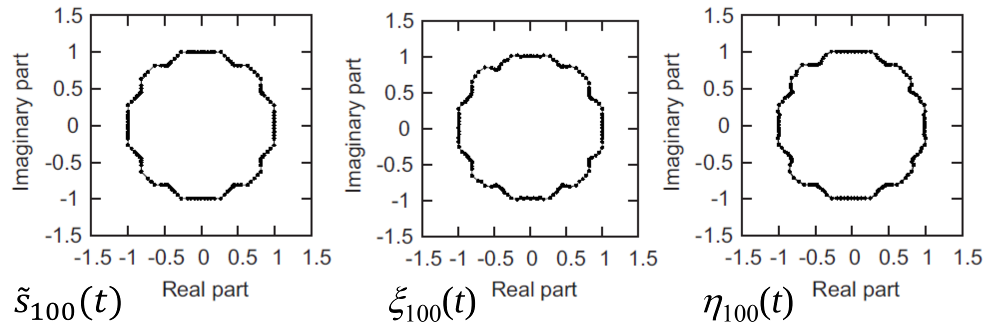
解析信号への近似（方法A）

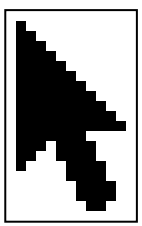


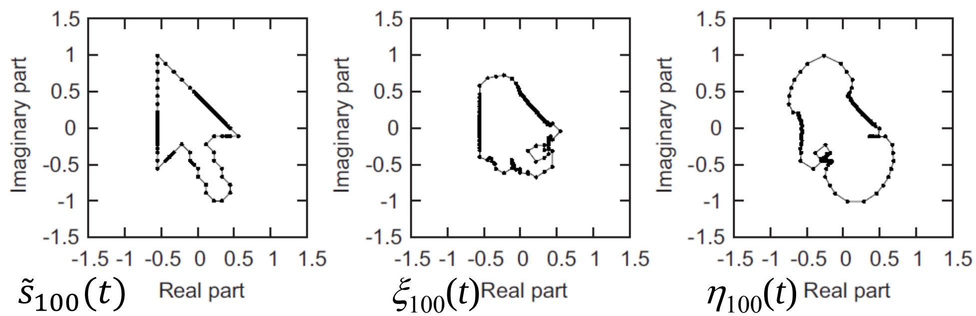
→サンプル点挿入により$\xi(t)$と$\eta(t)$との二乗誤差を最小化

シミュレーション結果（方法A）



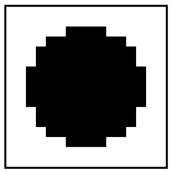


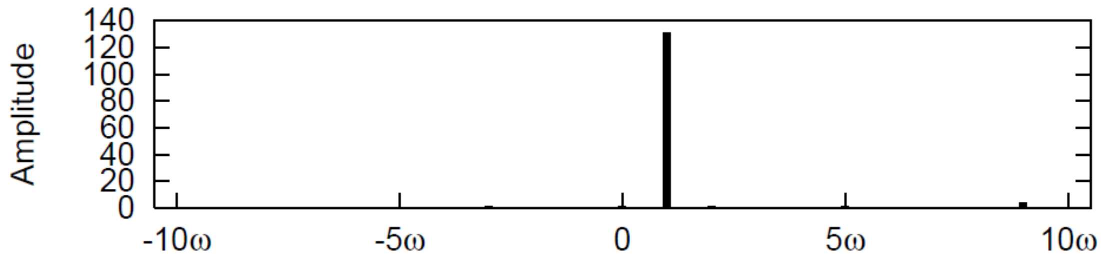


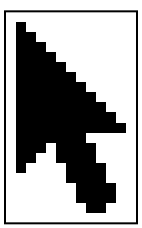


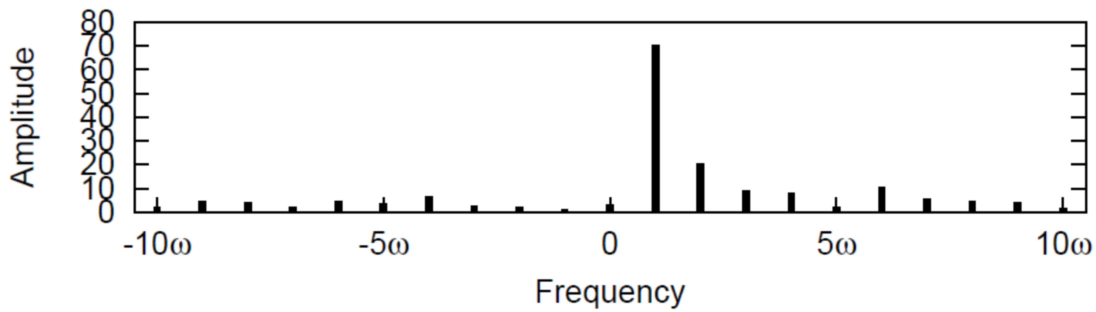
- 画像1はほぼ合致
- 画像2は原信号から乖離

シミュレーション結果（方法A）





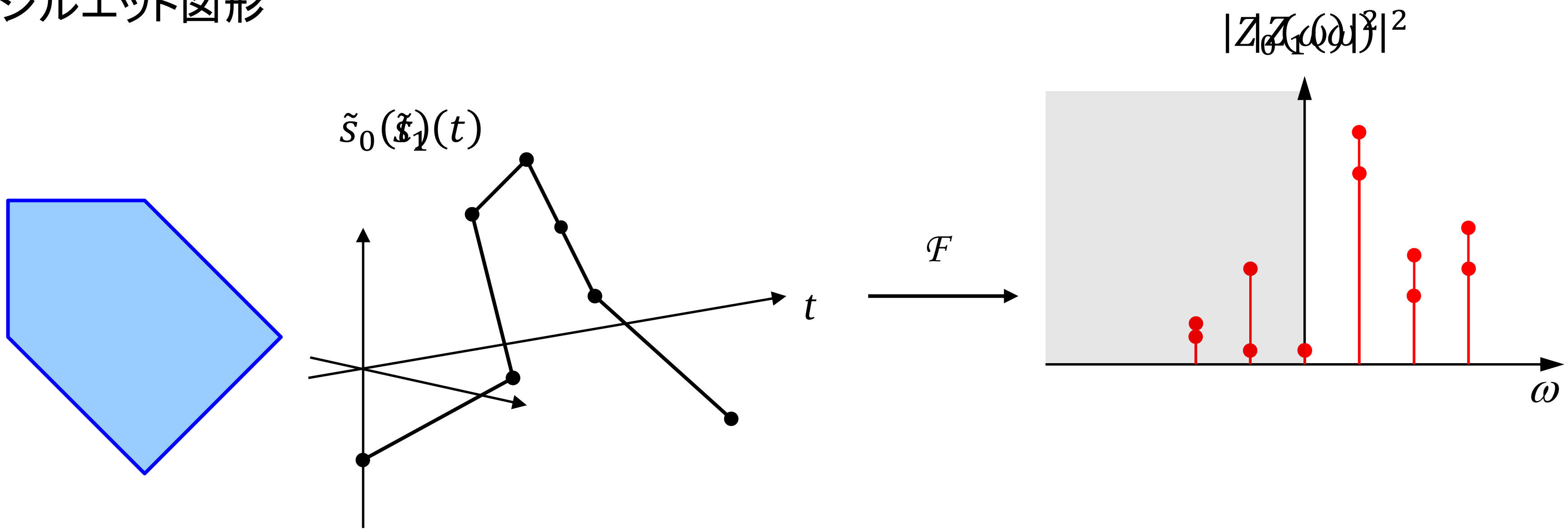




- 負周波数成分は減ったが全ては消えない
- 局所最適解の可能性もある

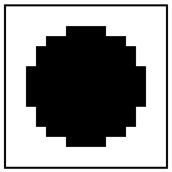
解析信号への近似（方法B）

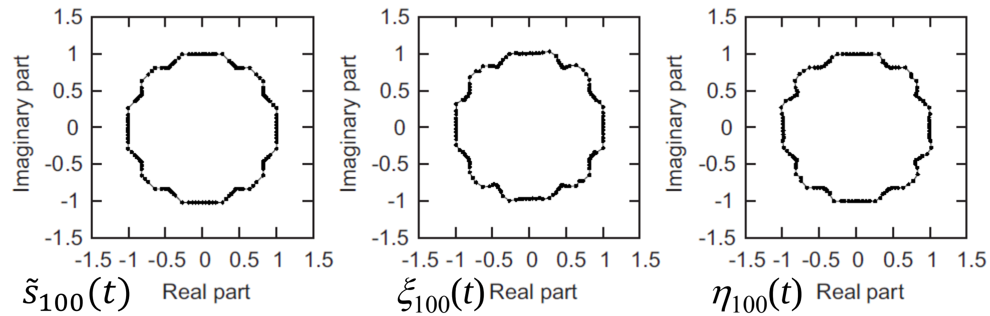
シルエット図形

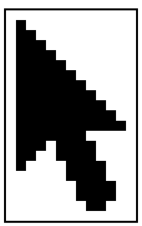


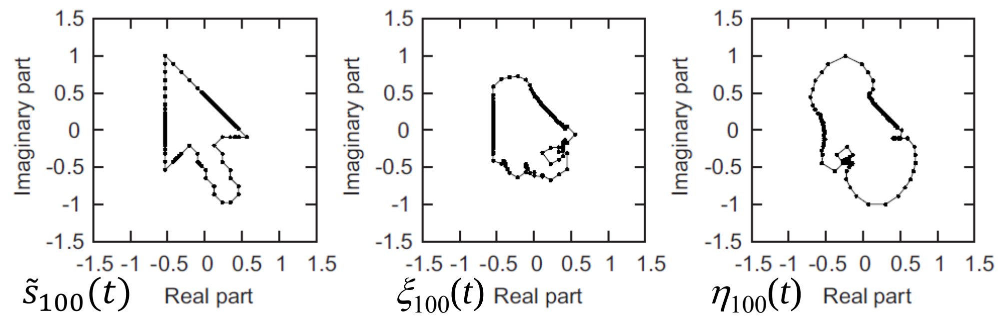
→サンプル点挿入により負周波エネルギーを最小化

シミュレーション結果（方法B）



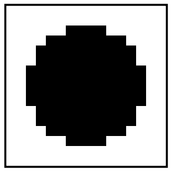


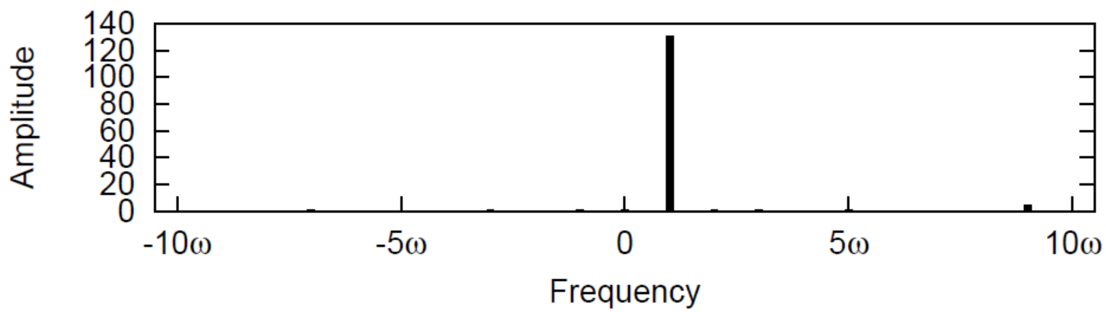


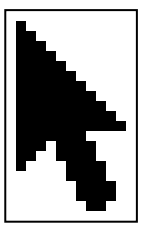


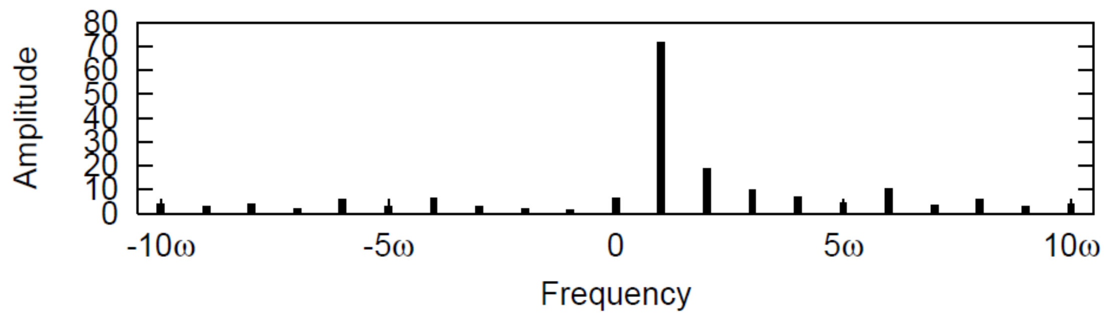
- 方法Aとほぼ同様の結果

シミュレーション結果（方法B）





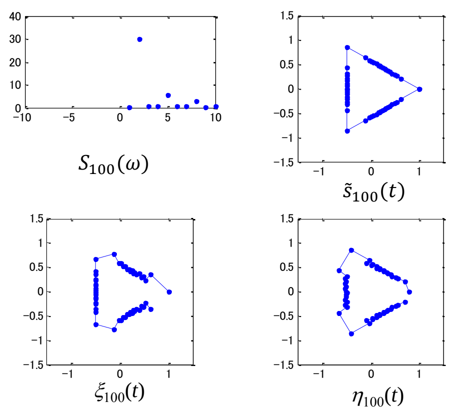




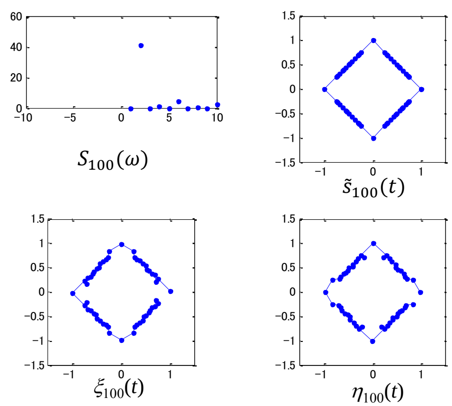
- 方法Aとほぼ同様の結果

多角形の変換

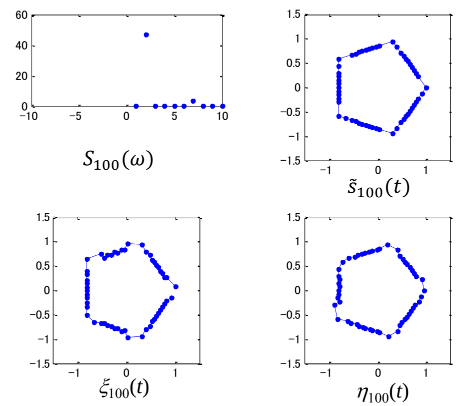
- 方法Aで同様にシミュレーション



正3角形



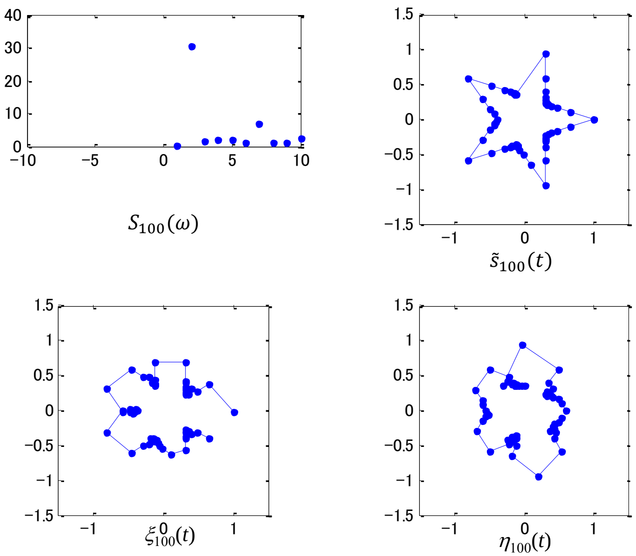
正4角形



正5角形

星形の変換

- 方法Aで同様にシミュレーション

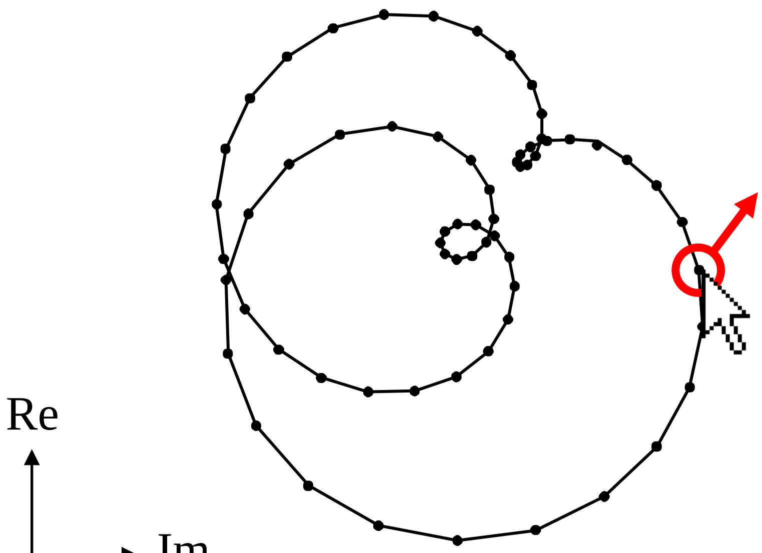


第5章

解析信号シンセサイザ「CLOSYNTH」の開発

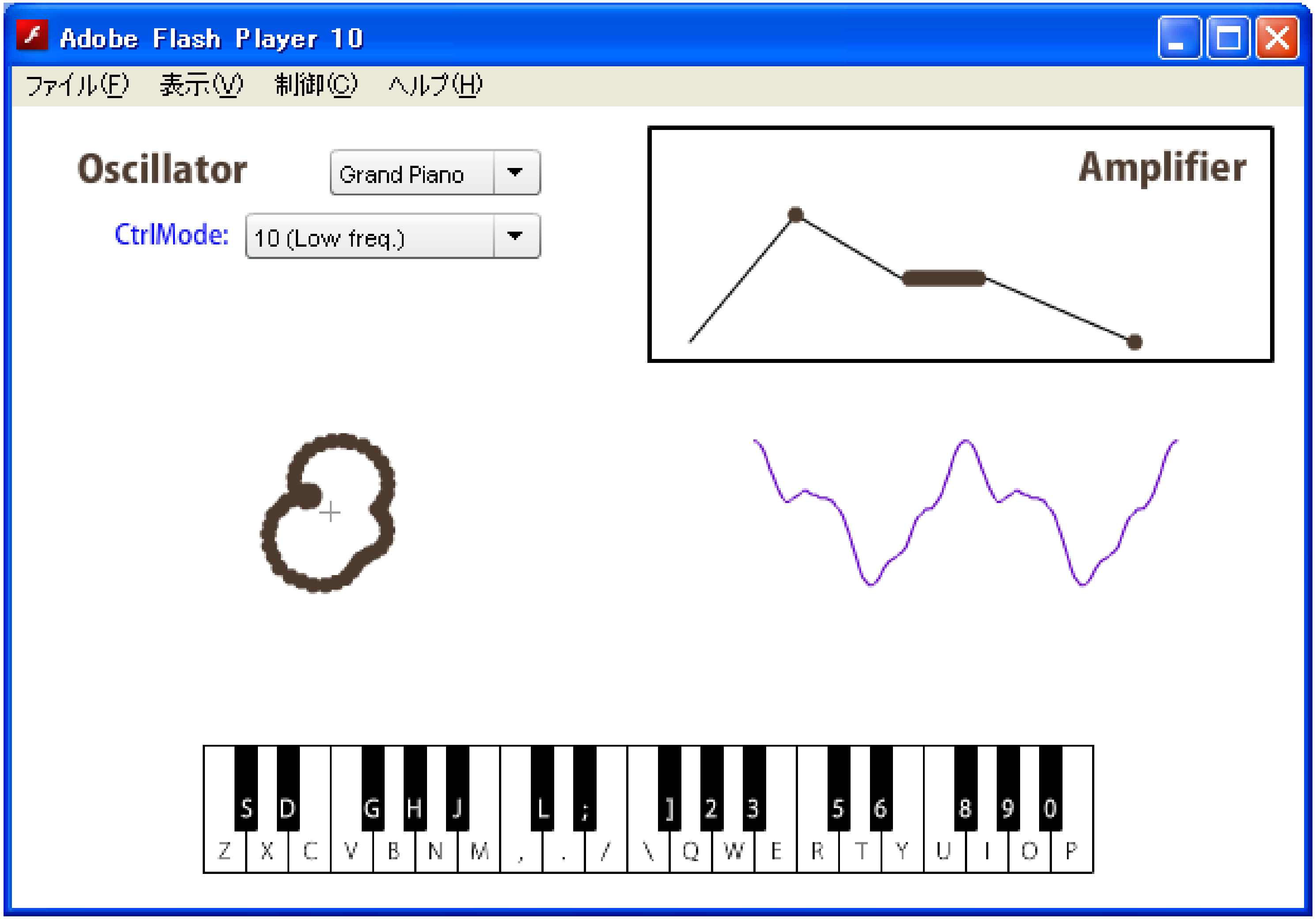
CloSynth: 解析信号シンセサイザ

- 解析信号＋解析信号＝解析信号
- 制御点をドラッグする度に解析信号を付加
- ツマミのないシンセサイザ

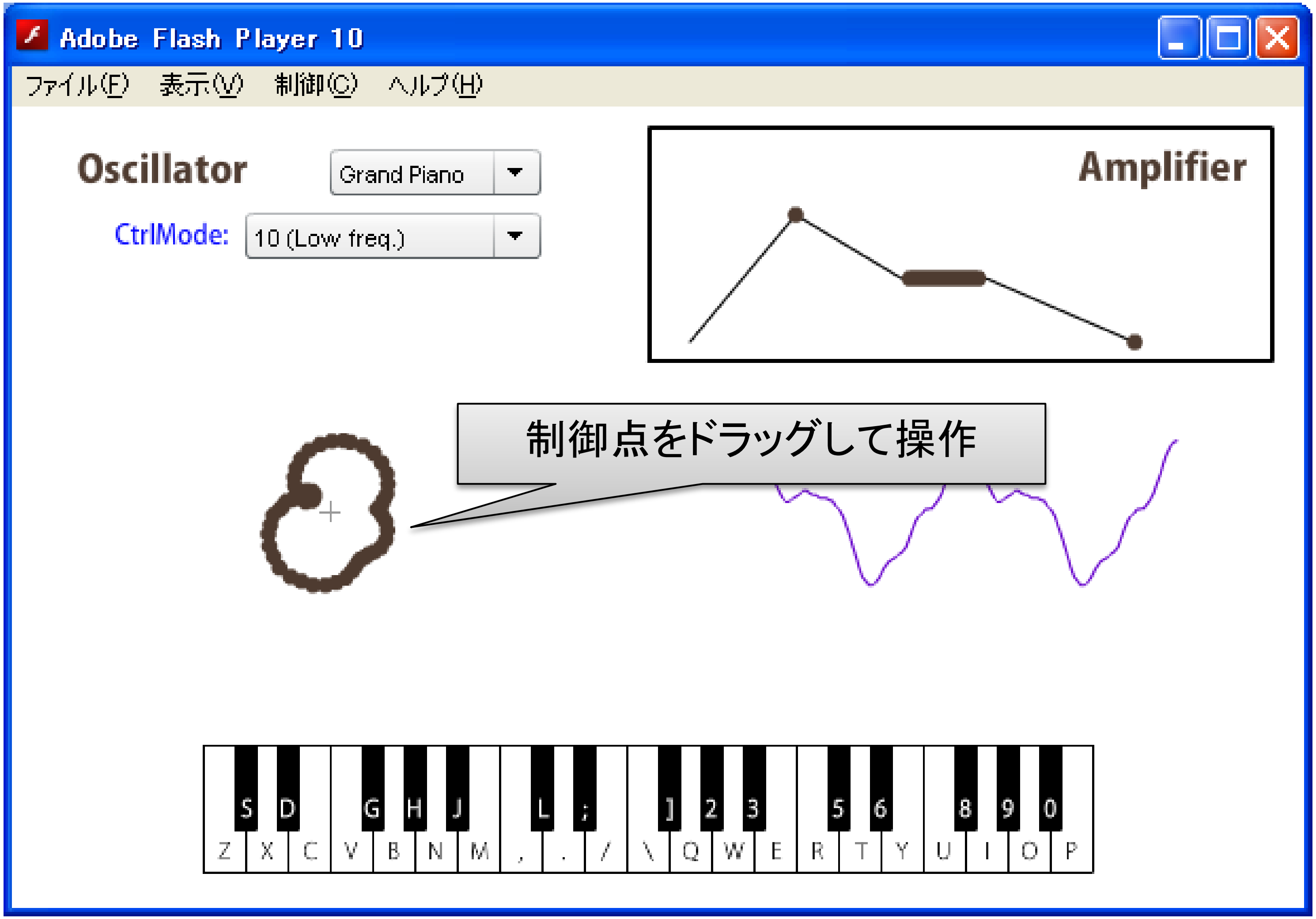


64個の制御点

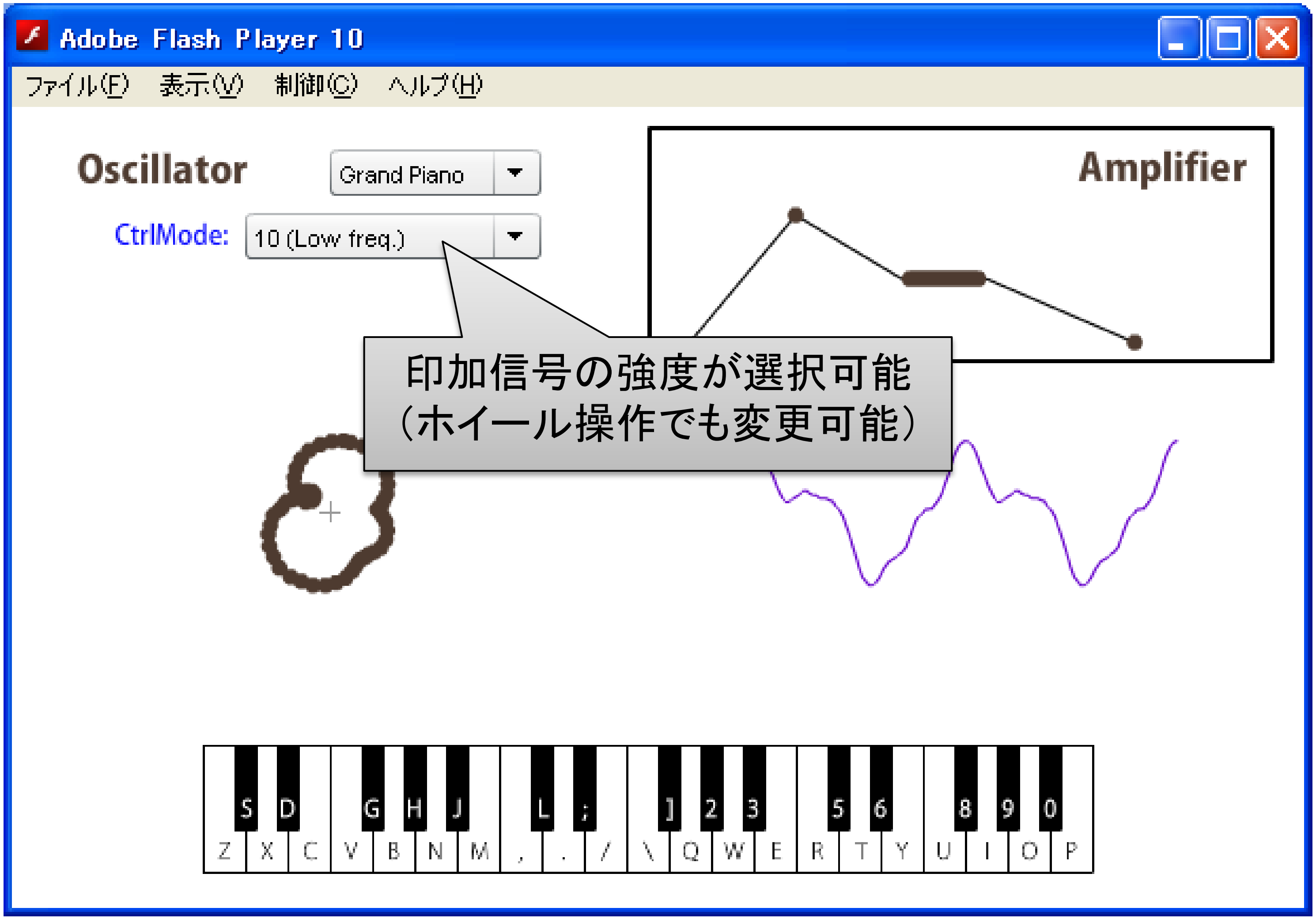
CloSynthの実行画面



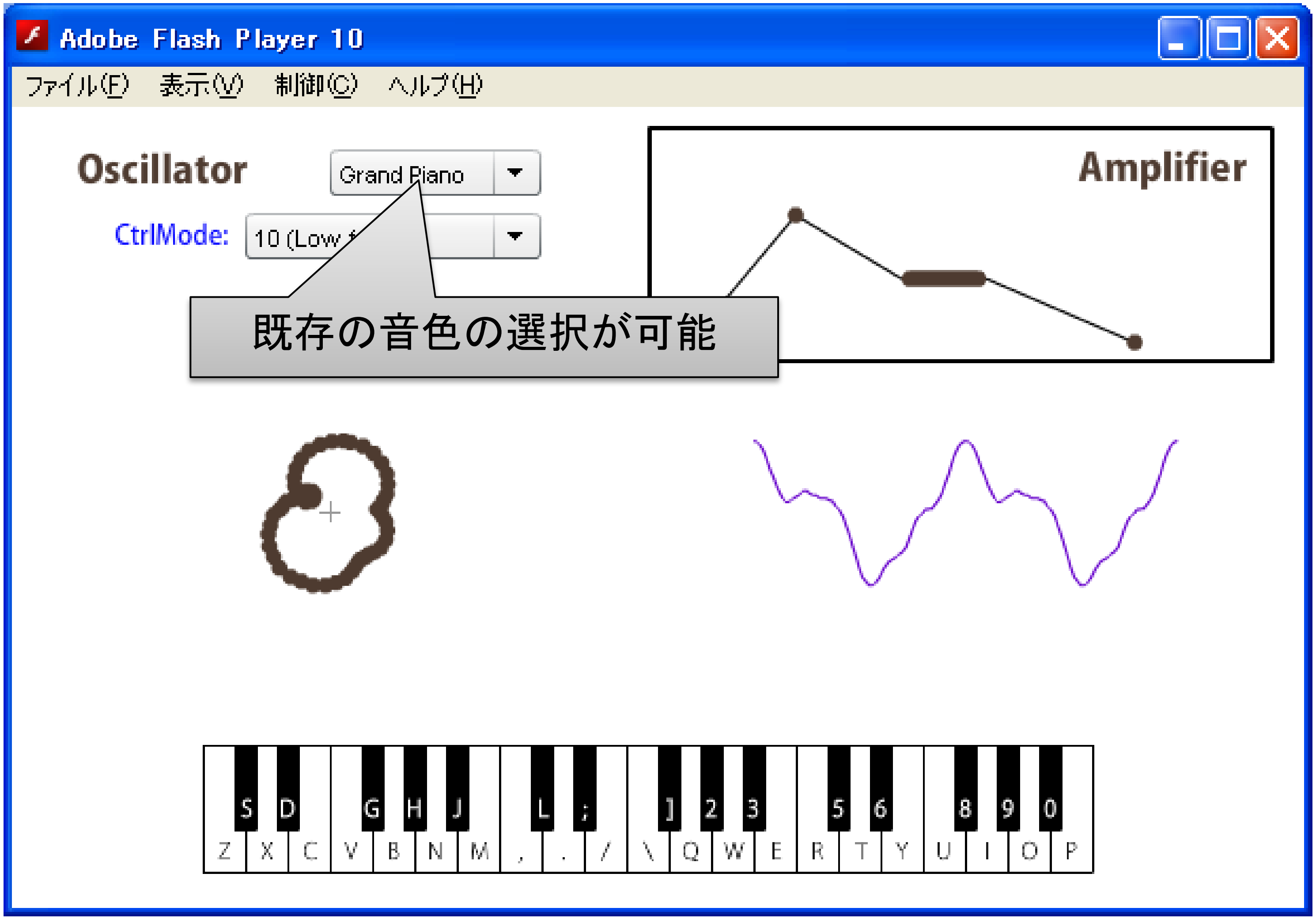
CloSynthの実行画面



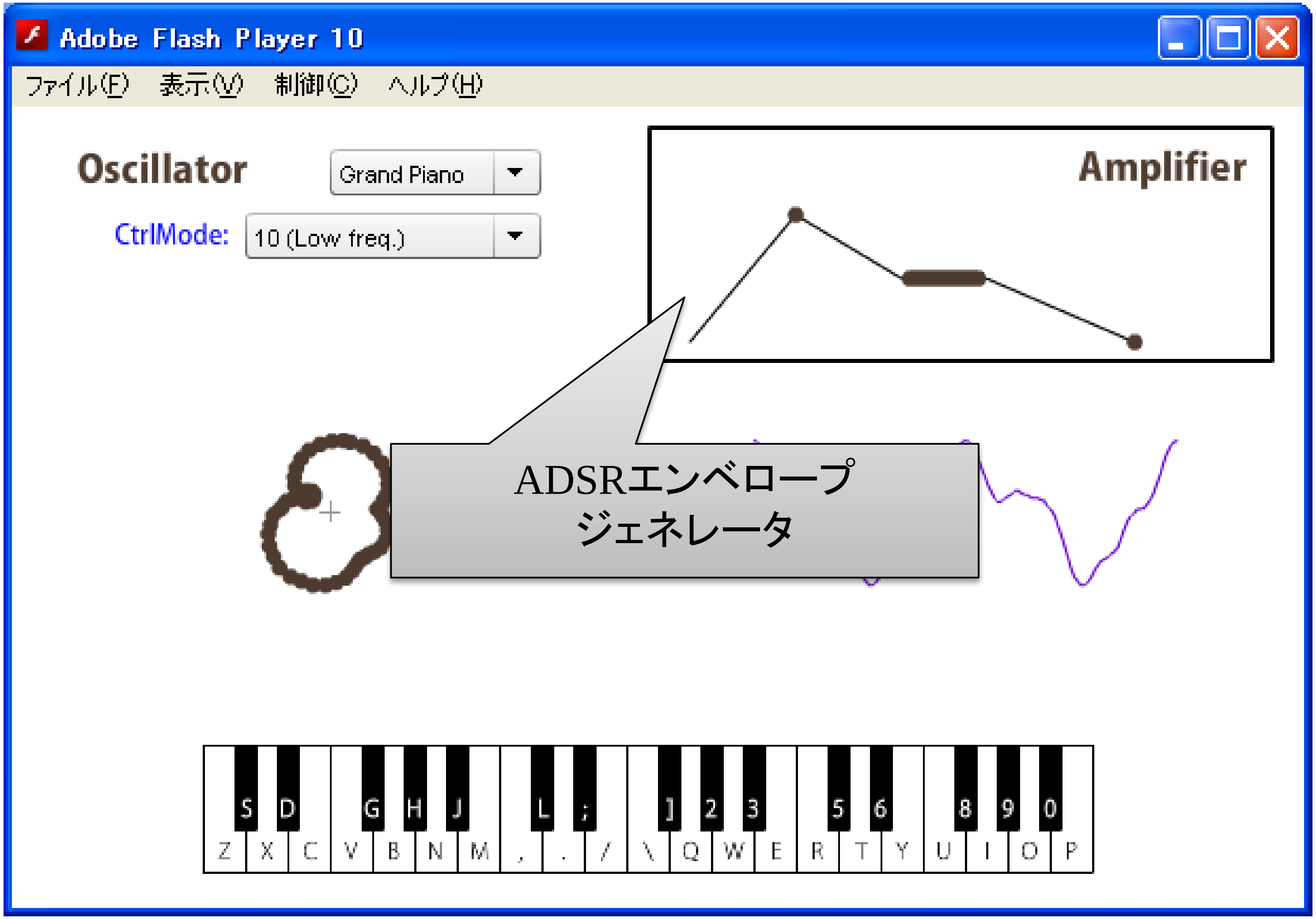
CloSynthの実行画面



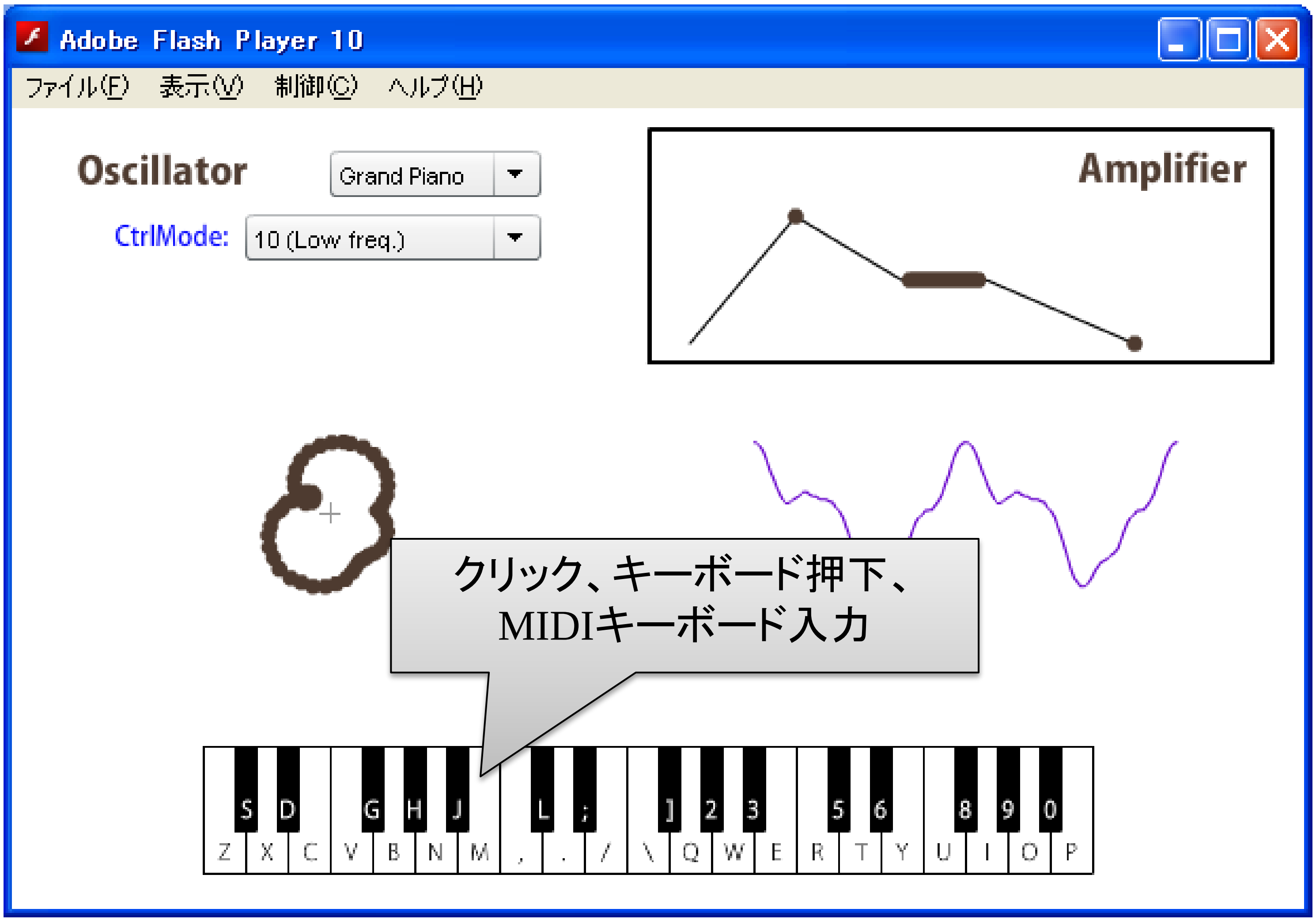
CloSynthの実行画面



CloSynthの実行画面

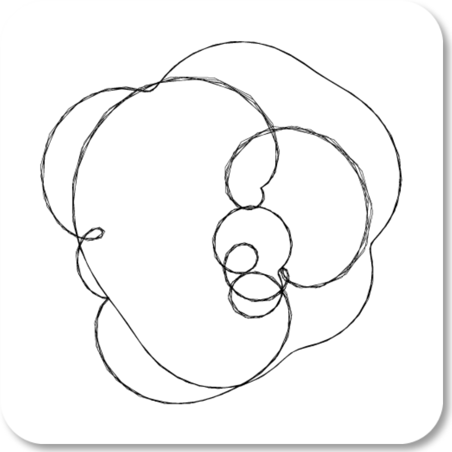


CloSynthの実行画面

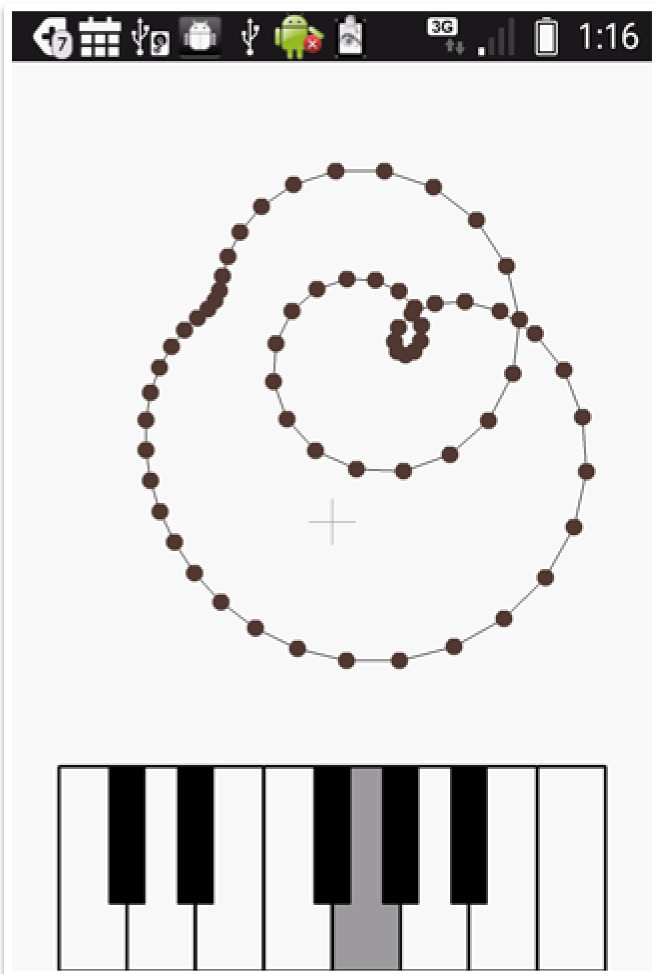


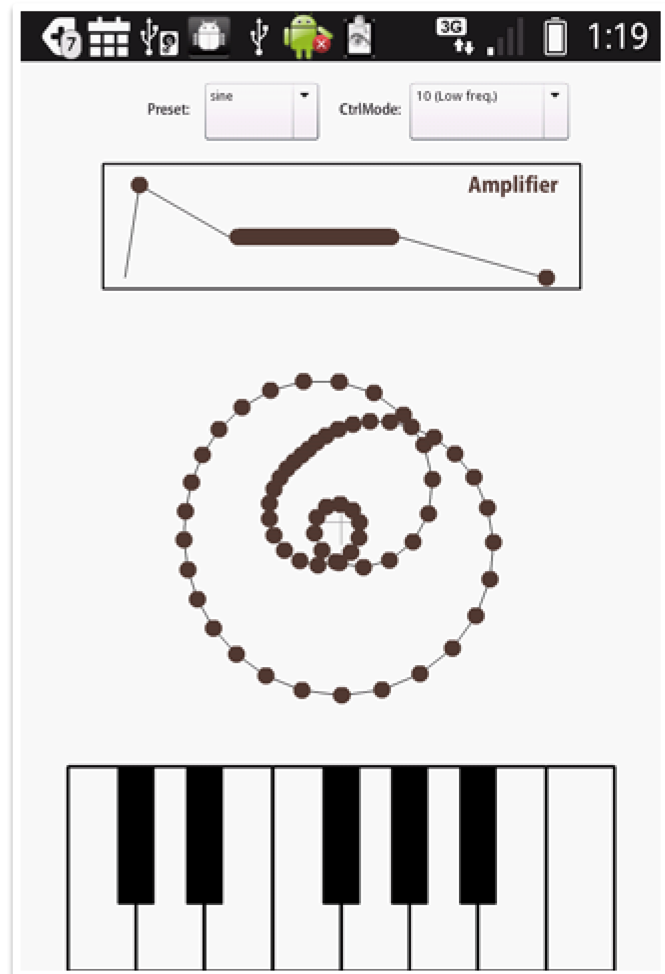
Android版CloSynth

- AIR for Androidで実装
- Android 2.2以上で動作
- Androidマーケットで配布中



Android版CloSynthの実行画面





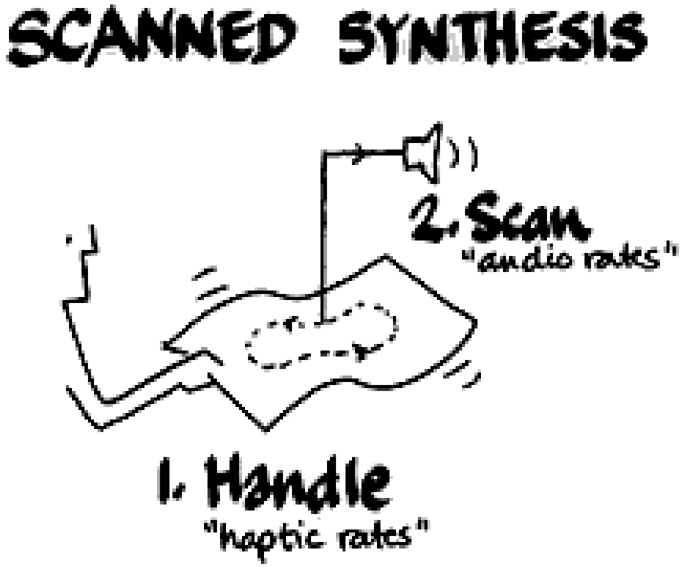
Android版CloSynthの問題点

- 制御点のタッチ反応領域が狭い
- 画面の広さの制約上、多くのモジュールを置けない
- 音声出力のタイムラグが大きい

Kanazawa University

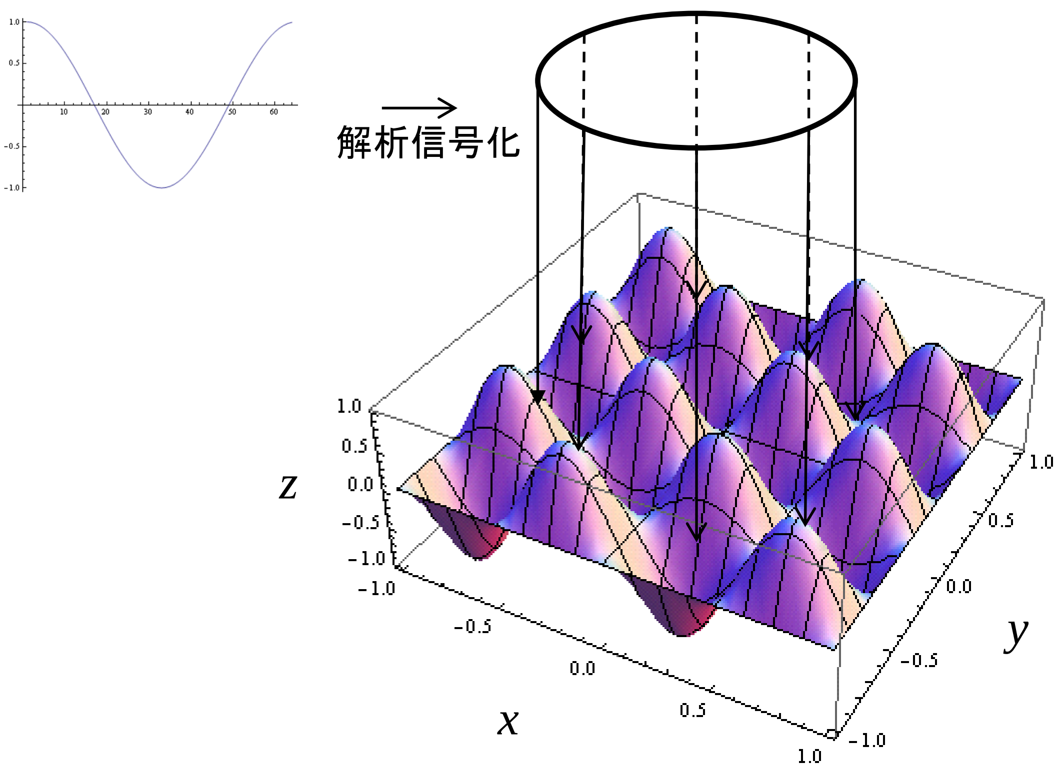
MeRL Stands for Microelectronics Research Lab.

Scanned synthesis



- **Handle**: 動的なモデル (例: ばね)
- **Scan**: 軌跡に沿って標本化する

解析信号で走査する
Scanned Synthesis



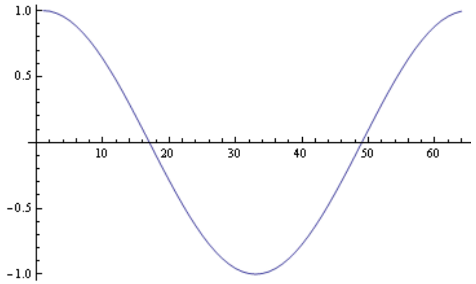
←Scan

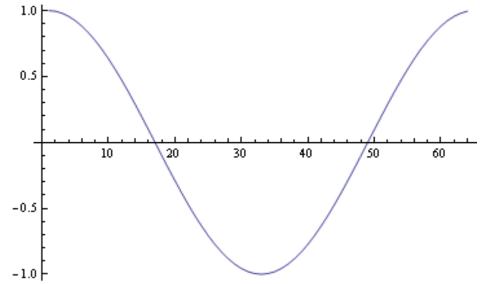
←Handle

$$z = f(\Re[\tilde{s}(t), \Im[\tilde{s}(t)])$$

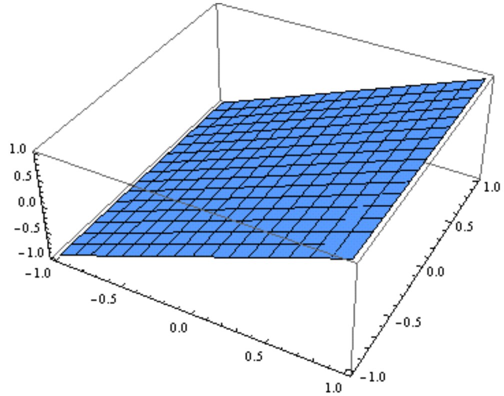
解析信号で走査する
Scanned Synthesis

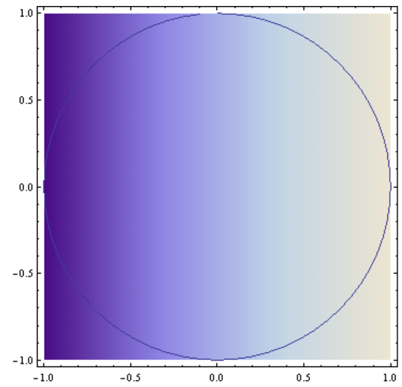
- $z = x \quad \rightarrow \quad z = \Re[\tilde{s}(t)] = s_x(t)$





↓ ↑

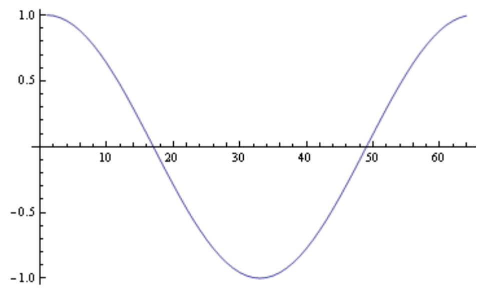


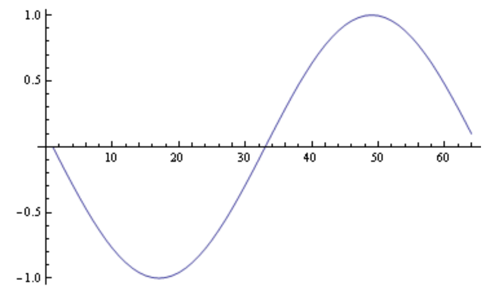


解析信号で走査する
Scanned Synthesis

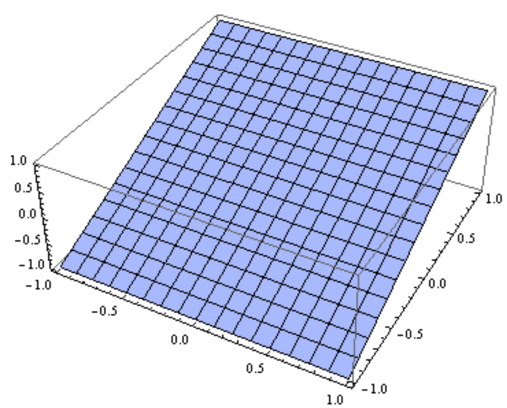
ヒルベルト変換

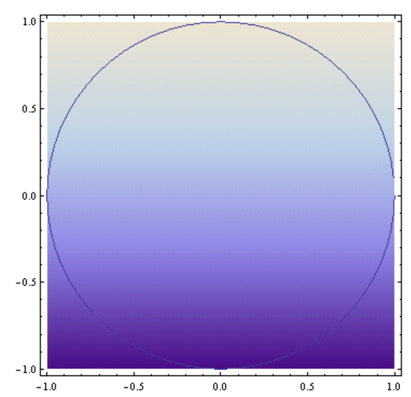
- $z = y \quad \rightarrow \quad z = \Im[\tilde{s}(t)] = \mathcal{H}[s_x(t)]$





↓ ↑

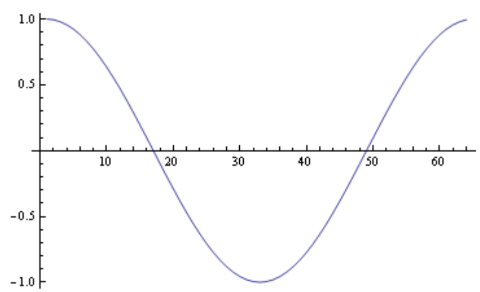


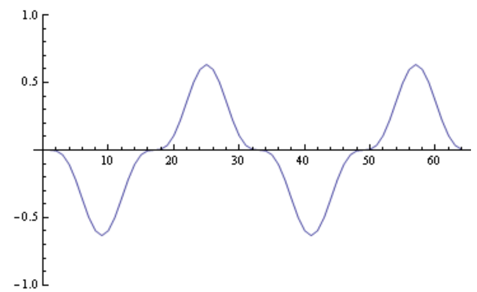


解析信号で走査する
Scanned Synthesis

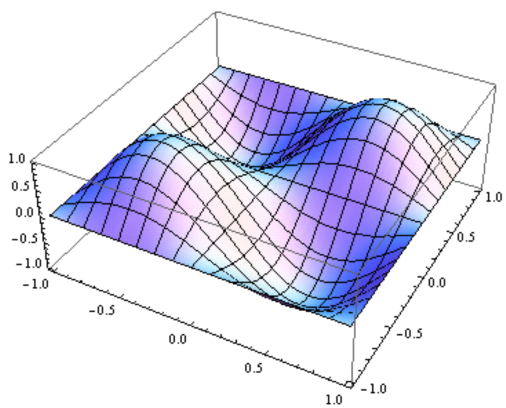
- $z = \sin(\pi x)\sin(\pi y)$

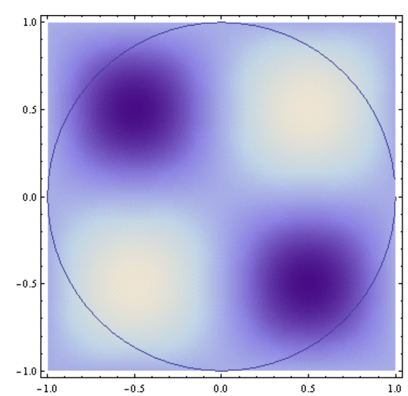
$$\rightarrow z = \sin(\pi s_x(t))\sin(\pi s_y(t))$$



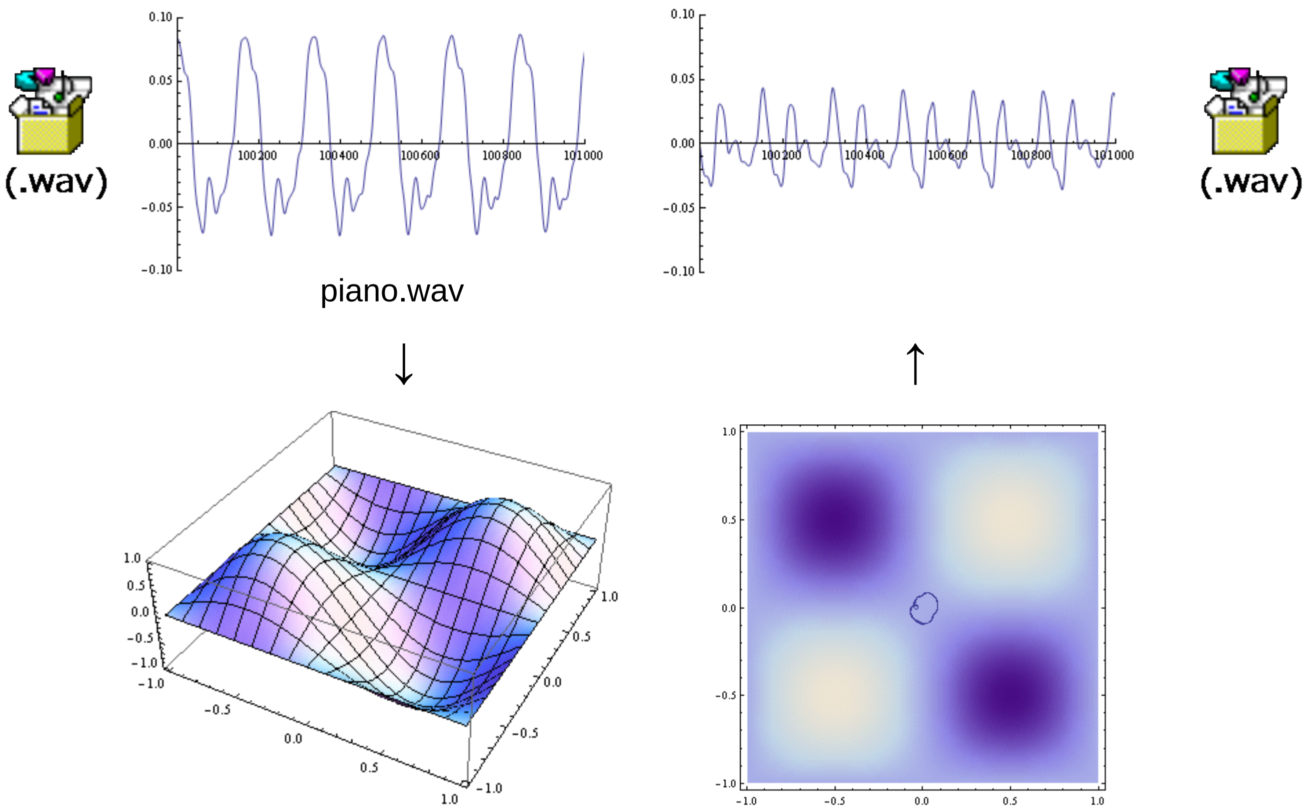


↓ ↑





解析信号で走査する
Scanned Synthesis



まとめ

- 解析信号を用いて、図形と音声の相互変換の手法を提案した
  - 音色を閉曲線図形として見ることが可能になった
- 解析信号を用いたシンセサイザを開発した
  - 直感的であるとはいえないが、GUIを有効活用したインタフェースを実現できた