

Jean-Claude Risset 温故知新

長嶋洋一¹

概要：ピッチに関する有名な錯聴のShepard Toneを、Jean-Claude Rissetがリズムに拡張した錯聴がRisset Rhythmであり、いずれも音楽情報科学する者であれば教養として知っていたい古典的な話題である。メディアデザイン教育において錯覚をテーマで取り上げると、必ずと言っていいほどShepard Toneは登場するものの、Risset Rhythmについては本人監修?のデモサウンドYouTubeを鑑賞する程度で過ぎ去ってしまうことも多い。本稿では、錯覚を鑑賞の対象として再生するだけでなく、インタラクティブな錯覚体験を指向するメディアデザイン教育に関連して、COVID-19のために遠隔教育体制となった2020年前期にあらためてRisset Rhythmを温故知新として取り上げた事例を紹介するとともに、音楽におけるRisset Rhythmの新たな可能性について検討してみたい。

キーワード：シェパードトーン、リセ・リズム、錯聴

Learning from the past : Jean-Claude Risset

YOICHI NAGASGIMAMA^{†1}

Abstract: The "Shepard Tone" is the famous illusion of hearing about pitch. The "Risset Rhythm" is an illusion that Jean-Claude Risset has extended the idea to rhythm, and both are classic topics that any information scientist should be familiar with. Whenever the theme of illusion is taken up in media design education, it is almost always the case that "Shepard Tone" appears, not appears about "Risset Rhythm". I will not only aim the illusion as an object of appreciation, but also provide an interactive illusionary experience. In this paper, I will report about media design education in SUAC with a distance education system against COVID-19, with taking up "Risset Rhythm". I also will discuss and examine the new possibilities of "Risset Rhythm" in music.

Keywords: Shepard Tone, Risset Rhythm, Illusion

1. はじめに

ピッチに関する有名な錯聴の「Shepard Tone」を、Jean-Claude Rissetがリズムに拡張した錯聴が「Risset Rhythm」であり、いずれも音楽情報科学する者であれば教養として知っていたい古典的な話題である。メディアデザイン教育において「錯覚」をテーマで取り上げると、必ずと言っていいほど「Shepard Tone」は登場するものの、「Risset Rhythm」については本人監修?とも言われるデモサウンドのYouTube動画[1]を鑑賞する程度で過ぎ去ってしまうことも少なくない。本稿では、「錯覚」を単なる鑑賞の対象として「再生」するだけでなく、インタラクティブな錯覚体験を指向するという筆者のメディアデザイン教育に関連して、COVID-19のために遠隔教育体制となった2020年前期にあらためて「Risset Rhythm」を温故知新として取り上げた事例について紹介するとともに、将来的には、音楽における「Risset Rhythm」の新たな可能性について検討してみたい。

ことの発端は、Shepard ToneをJean-Claude Rissetがリズムに拡張した錯聴として有名な「Risset Rhythm」について、筆者のゼミに北京から遠隔で参加しているSUAC研究生(王福瀛)が研究テーマとして探してきたことを発端に、筆者のSUACでの担当講義「音楽情報科学」の受講学生とともに検討するための資料を作ってみよう、と思い立ったことである。YouTubeにある動画[1]をmp3サウンド化したものが[2]であり、当初の目標としてはRisset Rhythmの原理を理解した上で、このようなサウンドを生成するMaxパッチ、あるいは手法を確立することである。

2. 「Shepard Tone」の復習

「Shepard Tone」も「Risset Rhythm」も、その基本にあるのは「複合サウンドの漸増/漸減」であるために、まず最初に「Shepard Tone」について確認した。図1は「Shepard_01.maxpat」と「Shepard_02.maxpat」という2つのMaxパッチのスクリーンショットである。いずれも14個の同一のパーシャル生成サブパッチ「sub_block」を順にトリガするという構成になっており、「sub_block」内の元波形ジェネレータが、前者はサウンドが「cycle~」による純音であり、後者はサウンドが「phasor~」による鋸

¹ 静岡文化芸術大学

Shizuoka University of Art and Culture

歯状波である、という違いがあるだけで、他は完全に同一である。共通の「sub_block」は外部からbangでトリガされると、「line」オブジェクト(浮動小数点演算)により、20.0から127.0に向けて、浮動小数点数(実数)を20秒間(20000msec)かけて線形(リニア)に増加させていく。この数値は「mtof」オブジェクトによって浮動小数点演算で補間されて、入力MIDIノートナンバに対応/補間された周波数(ピッチ)となって、音源のオシレータを駆動する。この「mtof」でのMIDIノートナンバとピッチとの関係は「オクターブ上がる(MIDIノートが12増える)ごとにピッチが2倍になる」という指数関数的な変換である。また一方で、テーブルを参照した値によって、ゼロから最大値127まで直線的に増加し、その後は127からゼロまで直線的に減少するエンベロープとなって、オシレータの出力と乗算される。結果として、この「sub_block」のパーシャルでは、ピッチがMIDI=20のノートからMIDI=127のノートに対応して指数関数的な一定速度で上昇するサウンドが、最初はゼロから次第に上昇して途中で最大値となり、その後は次第にゼロに向かうような音量変化を持つことになる。このパーシャルが14個並んで1.6秒ないし1.45秒ごとに次々にトリガされることで、全体としての音響は「上昇し続ける」という「Shepard Tone」を構成する。

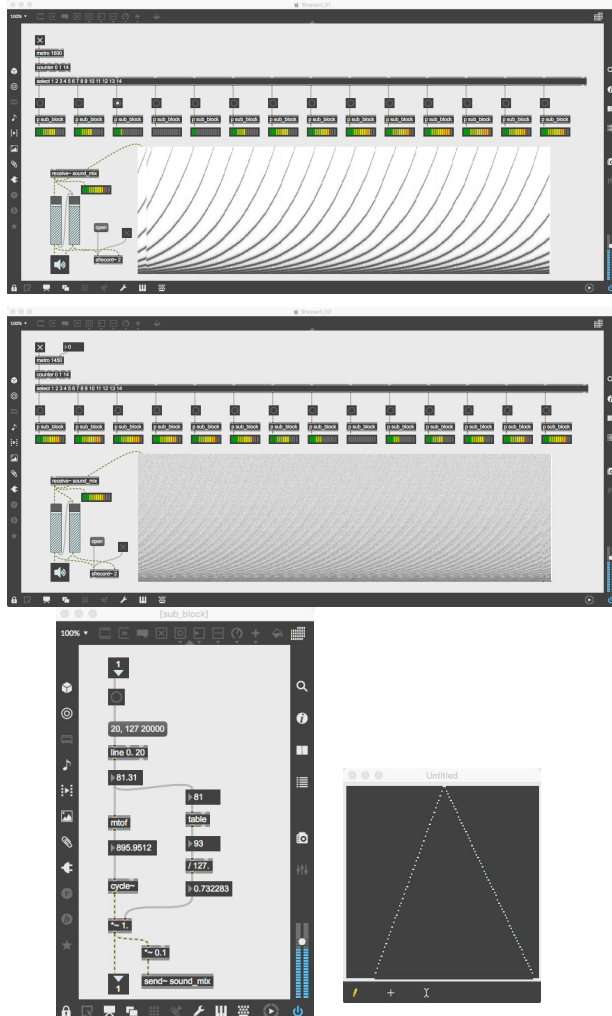


図1 シェパード・トーンのMaxパッチ
Figure 1 Max8 Patch of Shepard Tone.

3. 「Ghost Notes」仮説

サウンド[2]を最初に聞いた時の筆者の第一印象としては、Risset Rhythmの「肝」は、スネアに典型的な「ゴースト・ノート(Ghost Notes)」の活用ではないか、と判断した。加速している場合、ゴースト・ノートでビートの倍の荒さで叩いている音が加速とともに現れてくることで「元に戻って」くるというのがこの錯覚の本質であると思われた。また、一定のスピードで加速している感じになるためには、metroの数値をリニアに変化させては駄目なのは明らかだった。そこで、ゴーストノートを加えたドラムパターンを生成する図2のようなパッチを作ったが、テンポを決定するmetroの数値で「一定の加速感(指数関数的変化?)」を実装するのは難しそうだったので、このシーケンスを延々と走らせたものを「seq」オブジェクトで記録して「test.mid」というmidiファイルに書き出し、「test.mid」をQuickTimePlayer7に読み込ませてリアルサウンドのAIFFファイルとしてexportした(およそ11回のパターンで49秒ほどの素材ファイル)サウンドは、[3]のようになった。

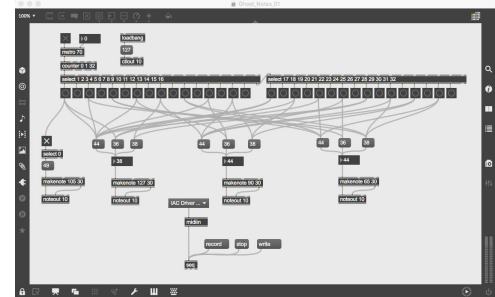
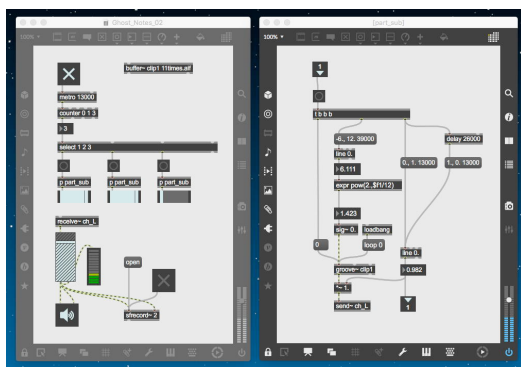


図2 サンプルリズムのMaxパッチ
Figure 2 Max8 Patch of Sample Rhythm.

単調に長く続くリズムパターンの素材サウンドファイルを、「刻々と一定の指数関数的な加速感」でテンポアップしつつ再生するために最適なものは「groove~」である。そこで図3の「Ghost_Notes_02.maxpat」では、前述のShepard Toneとほぼ同じ戦略で、同一のサブパッチ「part_sub」を3つ並べて、それぞれを順に駆動する構成とした。「groove~」に「sig~」経由で与える再生ピッチとは、「1.0」で元サウンドと同一のピッチ、「0.5」であればオクターブ下(半分の速度)、「2.0」であればオクターブ上(2倍の速度)なので、「mtof」と同様に指数関数的な関係でサウンドファイルの再生速度を制御できる。そこで、音楽関係では定番の「 $\text{expr pow}(2, \$f1/12)$ 」という関数を定義した。これは、12等分平均律に対応して「2の12乗根のx乗」(xは浮動小数点数)という「指数関数的な相対比」を演算する関数であり、 $x=0$ なら1.0(元のまま)、 $x=12$ なら2.0(オクターブ上)、 $x=-12$ なら0.5(オクターブ下)、というように、再生速度をMIDIノートナンバに相当する感覚で指定できることになる。この「Ghost_Notes_02.maxpat」では、3つの同等のサブパッチ「part_sub」を13秒ごとにトリガしているが、その「part_sub」内では、「-6., 12. 39000」という指定で「line」オブジェクト(浮動小数点演算)を駆動しているの、で「6半音だけ下からスタートして12半音(オクターブ上)のゴールまでを、等速で39秒間かけて増加」させていることになる。「39秒」という長さの理

を用いるのではなくて、シンプルなパターンをMaxパッチ内でループさせる(無限長の素材)方針とした。一見すると似ているようだが、図4のパッチ「Risset_001.maxpat」では、まず音素材としてページ[6]にあった素材サウンドの一つを採用し、厳密にはループしていなかったのを耳を頼りに手作業でトリミングしたサウンド[9]を「groove~」でループ再生している。



The screenshot displays a hierarchical tree structure within a software application. The tree is organized as follows:

- Root Node:** A large black square icon labeled 'A'.
- Level 1 Nodes:** Four main branches labeled 'B', 'C', 'D', and 'E'.
- Level 2 Nodes (under B):** 'B1', 'B2', 'B3', 'B4', 'B5', 'B6', 'B7', 'B8', 'B9', 'B10', 'B11', 'B12', 'B13', 'B14', 'B15', 'B16', 'B17', 'B18', 'B19', 'B20', 'B21', 'B22', 'B23', 'B24', 'B25', 'B26', 'B27', 'B28', 'B29', 'B30', 'B31', 'B32', 'B33', 'B34', 'B35', 'B36', 'B37', 'B38', 'B39', 'B40', 'B41', 'B42', 'B43', 'B44', 'B45', 'B46', 'B47', 'B48', 'B49', 'B50', 'B51', 'B52', 'B53', 'B54', 'B55', 'B56', 'B57', 'B58', 'B59', 'B60', 'B61', 'B62', 'B63', 'B64', 'B65', 'B66', 'B67', 'B68', 'B69', 'B70', 'B71', 'B72', 'B73', 'B74', 'B75', 'B76', 'B77', 'B78', 'B79', 'B80', 'B81', 'B82', 'B83', 'B84', 'B85', 'B86', 'B87', 'B88', 'B89', 'B90', 'B91', 'B92', 'B93', 'B94', 'B95', 'B96', 'B97', 'B98', 'B99', 'B100'.
- Level 2 Nodes (under C):** 'C1', 'C2', 'C3', 'C4', 'C5', 'C6', 'C7', 'C8', 'C9', 'C10', 'C11', 'C12', 'C13', 'C14', 'C15', 'C16', 'C17', 'C18', 'C19', 'C20', 'C21', 'C22', 'C23', 'C24', 'C25', 'C26', 'C27', 'C28', 'C29', 'C30', 'C31', 'C32', 'C33', 'C34', 'C35', 'C36', 'C37', 'C38', 'C39', 'C40', 'C41', 'C42', 'C43', 'C44', 'C45', 'C46', 'C47', 'C48', 'C49', 'C50', 'C51', 'C52', 'C53', 'C54', 'C55', 'C56', 'C57', 'C58', 'C59', 'C60', 'C61', 'C62', 'C63', 'C64', 'C65', 'C66', 'C67', 'C68', 'C69', 'C70', 'C71', 'C72', 'C73', 'C74', 'C75', 'C76', 'C77', 'C78', 'C79', 'C80', 'C81', 'C82', 'C83', 'C84', 'C85', 'C86', 'C87', 'C88', 'C89', 'C90', 'C91', 'C92', 'C93', 'C94', 'C95', 'C96', 'C97', 'C98', 'C99', 'C100'.
- Level 2 Nodes (under D):** 'D1', 'D2', 'D3', 'D4', 'D5', 'D6', 'D7', 'D8', 'D9', 'D10', 'D11', 'D12', 'D13', 'D14', 'D15', 'D16', 'D17', 'D18', 'D19', 'D20', 'D21', 'D22', 'D23', 'D24', 'D25', 'D26', 'D27', 'D28', 'D29', 'D30', 'D31', 'D32', 'D33', 'D34', 'D35', 'D36', 'D37', 'D38', 'D39', 'D40', 'D41', 'D42', 'D43', 'D44', 'D45', 'D46', 'D47', 'D48', 'D49', 'D50', 'D51', 'D52', 'D53', 'D54', 'D55', 'D56', 'D57', 'D58', 'D59', 'D60', 'D61', 'D62', 'D63', 'D64', 'D65', 'D66', 'D67', 'D68', 'D69', 'D70', 'D71', 'D72', 'D73', 'D74', 'D75', 'D76', 'D77', 'D78', 'D79', 'D80', 'D81', 'D82', 'D83', 'D84', 'D85', 'D86', 'D87', 'D88', 'D89', 'D90', 'D91', 'D92', 'D93', 'D94', 'D95', 'D96', 'D97', 'D98', 'D99', 'D100'.
- Level 2 Nodes (under E):** 'E1', 'E2', 'E3', 'E4', 'E5', 'E6', 'E7', 'E8', 'E9', 'E10', 'E11', 'E12', 'E13', 'E14', 'E15', 'E16', 'E17', 'E18', 'E19', 'E20', 'E21', 'E22', 'E23', 'E24', 'E25', 'E26', 'E27', 'E28', 'E29', 'E30', 'E31', 'E32', 'E33', 'E34', 'E35', 'E36', 'E37', 'E38', 'E39', 'E40', 'E41', 'E42', 'E43', 'E44', 'E45', 'E46', 'E47', 'E48', 'E49', 'E50', 'E51', 'E52', 'E53', 'E54', 'E55', 'E56', 'E57', 'E58', 'E59', 'E60', 'E61', 'E62', 'E63', 'E64', 'E65', 'E66', 'E67', 'E68', 'E69', 'E70', 'E71', 'E72', 'E73', 'E74', 'E75', 'E76', 'E77', 'E78', 'E79', 'E80', 'E81', 'E82', 'E83', 'E84', 'E85', 'E86', 'E87', 'E88', 'E89', 'E90', 'E91', 'E92', 'E93', 'E94', 'E95', 'E96', 'E97', 'E98', 'E99', 'E100'.

The interface includes a sidebar on the left with icons for file operations and a top bar with a search field and a 'Filter' button. The bottom of the window shows a taskbar with various system icons.

そして同一の4つのブロックでは、「-6., 18. 61520」ということで「原音ピッチの半音6つ下」から「原音ピッチのオクターブと半音6つ上」までの、計2オクターブにわたって、61.52秒かけてテンポアップしており、この時間を4分割した区間のうち最初の区間1でゼロから1へ音量が増加して、「原音→オクターブ上」のテンポアップに対応した区間2と区間3は最大音量1のままで、最後の区間4で1からゼロまで音量が減少するようにした。4つのブロックはこの区間の幅に相当するインターバルで次々にトリガされてクロスフェードしていくので、全体として「原音→オクターブ上」のテンポアップに対応した区間は常に最大音量で鳴り続けていることになる。

この61520msecという時間幅については適当に与えた値ではなくて、以下のように予備実験として「位相を合わせる」ことを追求した結果の値である。最初におよそ30秒程度ということで「0., 12. 30000」という値を走らせて、その際に「groove~」の右端出力(波形読み出しのランプ関数がゼロクロスする「Loop Sync Output」)からLoop Endのタイミングを抽出し、全体のスタートからの経過時間を計測する「clocker」をn回目のLoop Endでストップさせるというアルゴリズムを走らせると、n=24の時にclockerは30510から30525あたりになった。そこで値を「0., 12. 31000」とするとclockerは30790あたりになり、この間で両者がほぼ等しくなる数値を試行錯誤的に調べて、このパッチのように30760という値(この時にclockerは30730付近)を決定した。これにより、各ブロックの位相はほぼ合っていることになり、ループの先頭のビートはほぼ一致することになる。この結果を録音してみるとサウンド[10]になった。前節よりも合っている感じもするが、しかしまだまだである。その理由としては、整数でビートが揃うわけではない「6半音」というポイントで相互に接続しているところに問題があるように思われた。

5. Loopからの生成とCrossFade

第3節の実験ではそれなりの加速感があったものの、全体として騒然としていた原因が、まさにこの「位相」の欠如にあるということから、ゴーストノートのアイデアをとりあえず棚上げして、有限な長さの素材サウンドファイル

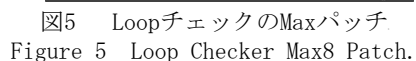
6. 2の累乗の完全Loop

前節の結果がいまいちだったものの、問題意識がかなり明確になってきたのか、論文[8]をあらためて読んでみて、

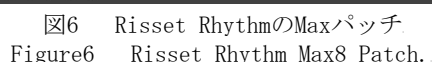
7. 別サウンドで手法の確認

8. 逆方向：無限に遅くなるサウンド

論文[7]の最後のあたりでは、Risset Rhythmの音楽的な考察をしている。これまでの作曲理論や音楽美学では、音楽の速度がテンポアップすることは、気持ちの高揚や興奮といった影響があり、作曲家も指揮者も演奏家も聴衆も、その効果を暗黙の共通理解として共有していた。ところがRisset Rhythmで「無限に加速する」となれば、そんなに永遠に気持ちの高揚や興奮が続くはずがないので、これは美学的におおいなる問題提起であり、新たな音楽の可能性を提示することにもなる。ところで、ここまで「無限に加速するサウンド」ばかりを扱ってきたが、サウンドのテンポ変更の方向を単純に逆向きにするだけで、「無限に減速するサウンド」も実現できる筈である。音楽心理学的には、テンポダウンによって、気持ちが落ち着いたり気分が落ち込んだりするはずなので、「無限に減速する」となれば、これはどこまでも落ち込む気分を齎す可能性がある。そこで、第6節で「無限に加速」していた素材をそのまま使用して、加速指定のメッセージの起点と終点を入れ替えただけの図7のパッチ「Risset_004.maxpat」を作ってみたところ、見事にサウンド[14]になり、まさに「無限に遅くなり続けるリズム」が出来た。じっと聞いていると、たしかに、どんどん落ち込む感じがする。さらに、前節で「無限に加速」していたピッチ感のある素材[11]をそのまま使用して、加速指定のメッセージの起点と終点を入れ替えただけパッチ「Risset_005.maxpat」を作ってみたところ、見事にサウンド[15]になり、「無限に遅くなり続けるリズム」が出来た。こちらもしっと聞いていると、たしかに、どんどん落ち込む感じがするし、落ち着いてくるのでこちらの方が精神衛生上は良い感じもある。



これを受けて、図6のようなパッチ「Risset_002.maxpat」を作ってみたところ、見事にサウンド[11]が得られて、完全に立派なRisset Rhythmとなった。パッチ「Risset_002.maxpat」全体は5つの加速ブロックからなっていて、下から順に、「2オクターブと6半音だけ下→1オクターブと6半音だけ下」に加速する第1ブロック、「1オクターブと6半音だけ下→6半音だけ下」に加速する第2ブロック、「6半音だけ下→6半音だけ上」に加速する第3ブロック、「6半音だけ上→1オクターブと6半音だけ上」に加速する第4ブロック、「1オクターブ6半音だけ上→2オクターブと6半音だけ上」に加速する第5ブロック、からなる。音量変化を与えているのは、全体をかけてゼロから1に増加する第1ブロックと、全体をかけて1からゼロに減少する第5ブロックの二つだけであり、残りの第2ブロックから第4ブロックまでは音量はずっと最大音量の1のままである。この「metabar」を単純ループすると、例えば第1ブロックの最後は第2ブロックの先頭と自然に連結し、第2ブロックの最後は第2ブロックの先頭と自然に連結し・・・ということで、全体は「加速ブロックの螺旋」のような構造のために、無限に加速し続けるサウンドとなる。これがRisset Rhythmの本質であった。



9. その後の展開

2020年の音楽情報科学研究会・夏のシンポジウムはオンライン開催となり、さらに通常に比べて情報処理学会への研究報告の予稿提出期限が異常に早いルールが告知されたため、本稿ではここまでとするが、2020年7月に開始した本研究はさらに進んでいる。ここからの部分については、2020年10月24/25日にオンライン開催予定の、**日本音楽知覚認知学会2020年度秋季研究発表会**（日本音響学会音楽音響研究会2020年10月研究会と共催）において報告する予定である。既に成果が得られている本研究の「その後」について、ここでの発表の予告編として簡単に紹介しておくと、以下のようなものがある。

We Will Rock You

メロディーと長さがそこそこある、印象的なリフレインを素材としたらどうなるか、という実験を行った。QueenのRockミュージカル「We Will Rock You」の、あのリフレインを使ったRisset Rhythmを作成して、見事に完成した。さらに逆方向の変化で「無限に減速するあのリフレイン」を作ったところ、繰り返して盛り上がる筈の曲が、サウンドを聞いているとどこまでも落ち込んでいく不思議に気分になった。

サウンド素材特性の実験

Queenでの成功に味をしめて、さらに大きくキャラの異なる3種類のサウンド素材を使つての実験を進めた。素材の歌手としてはせっかくなので筆者と同級生の、マイケルジャクソン（鬼籍入り）と石川さゆり（同年生まれ1学年上）と山口百恵の3人を選択し、いずれの音源もずっと昔に購入していたCDから取り出した。具体的には、マイケルジャクソン「スリラー」より、石川さゆり「津軽海峡・冬景色」より、山口百恵「プレイバックpart2」より、ということで、いずれも興味深いRisset Rhythmが出来た。

「逆Shepard Tone」の復習

Risset Rhythmに適用するための予備実験として、過去に作成していた「逆Shepard Tone」について確認した。これは「無限に上昇している」と同時に「下降し続けている」Shepard Tone、というものである（逆パターンもあり）。これについてMaxパッチで容易に実装できることを確認した

新たな「逆Shepard Tone」探求

前項の復習を受けて、さらに新しいアイデアでの「逆Shepard Tone」を探求した。復習した発想の視点は「周波数特性」にあったが、ここでは「空間定位」の視点から追求した。しかし結果としては、ちょっと予想外に問題点が出現してきた。

metabar生成規則性の探求

それまで手作業のトライアンドエラーで進めてきた「Risset Rhythm生成アルゴリズム」の「metabar計測」について、積み重ねた事例からなんらかの法則性を見出せるかどうかを整理した。そして一般則として使用できそうな定数を決定し、さらに「定位移動」の視点のRisset Rhythm

生成システムも試作してみた。

Real Time Risset Rhythm Generator

ただ既存のサウンドファイルを「再生」するRisset Rhythmから、リアルタイムComputer Musicパフォーマンスのために発展する最初の一步を実験してみた。この部分はまだ本稿執筆時点では未確定である。

10. おわりに

温故知新とはいいい言葉で、長い歴史のある音楽情報科学においては、遠い過去のもののようで、新しい道具や発想からの光を当てればまだまだ発掘できる宝石が埋蔵されている。今後もさらに新しい温故知新を進めていきたい。

参考文献

1. https://youtu.be/oQf_tS5WAP4
2. <https://nagasm.org/1106/mac/RissetRhythm.mp3>
3. <https://nagasm.org/ASL/RissetRhythm/sound/003.mp3>
4. <https://nagasm.org/ASL/RissetRhythm/sound/004.mp3>
5. <https://swiki.hfbk-hamburg.de/MusicTechnology/826>
6. https://en.wikipedia.org/wiki/File:Risset_accelerando_beat1_MCLD.ogg
7. <https://www.fransabsil.nl/htm/rissetcalc.htm>
8. <http://c4dm.eecs.qmul.ac.uk/papers/2011/Stowell2011icmc.pdf>
9. <https://nagasm.org/ASL/RissetRhythm/sound/006.mp3>
10. <https://nagasm.org/ASL/RissetRhythm/sound/007.mp3>
11. <https://nagasm.org/ASL/RissetRhythm/sound/008.mp3>
12. <https://nagasm.org/ASL/RissetRhythm/sound/009.mp3>
13. <https://nagasm.org/ASL/RissetRhythm/sound/010.mp3>
14. <https://nagasm.org/ASL/RissetRhythm/sound/011.mp3>
15. <https://nagasm.org/ASL/RissetRhythm/sound/012.mp3>