

自動演奏ロボットのための協調演奏システム： 外部演奏環境との協調用基本技術

若林, 直哉 / WAKABAYASHI, Naoya

(出版者 / Publisher)

法政大学大学院理工学・工学研究科

(雑誌名 / Journal or Publication Title)

法政大学大学院紀要. 理工学・工学研究科編 / 法政大学大学院紀要. 理工学・工学研究科編

(巻 / Volume)

57

(開始ページ / Start Page)

1

(終了ページ / End Page)

8

(発行年 / Year)

2016-03-24

(URL)

<https://doi.org/10.15002/00013006>

自動演奏ロボットのための協調演奏システム -外部演奏環境との協調用基本技術-

COOPERATIVE PERFORMANCE SYSTEM FOR AUTOMATIC PLAYING ROBOTS -BASIC TECHNOLOGY FOR SYNCHRONIZING TO THE SURROUNDING PERFORMANCE-

若林直哉

Naoya WAKABAYASHI

指導教員 高島俊

法政大学大学院理工学工学研究科機械工学専攻修士課程

Automatic playing robots are the robots that can play musical instruments just like human players by using artificial mechanical systems and the intelligent computer systems. When we want to try concerts with those robots, it is inevitably necessary to synchronize “our” performance to the robots. As an “automatic” robot, the robot should synchronize to our performance in tempo, expressions and etc. In this paper, a new type of automatic playing robot is proposed that can synchronize the musical environment with human players and perform a perfect concert, that is, a cooperative playing system. As a basic cooperative system, the system developed can change the playing tempo along with an external click signal. It can play music written in the MIDI format by the existing saxophone playing robot with synchronizing the tapping sequences input by a human.

Key Words: Automatic Playing Robot, Tempo Fitting, Saxophone Robot, Cooperative play

1. 緒論

自動演奏ロボットとは、人間が楽器を演奏するメカニズムを機構によって再現することによって楽器の演奏ができるロボットである¹⁾²⁾。このような自動演奏ロボットと人間とで合奏を行おうと試みた場合、従来のシステムにおいてはロボットの演奏に人間が合わせる必要がある。そこで、本研究においては、ロボットが人間側の演奏、即ち外部の演奏環境に合わせて自動演奏を行う「協調演奏」のシステムを開発することを目的としている。

協調演奏を行うためには、ロボットが周囲の奏者の演奏音からテンポを取得し、得られたテンポに合わせて自らの演奏テンポを変更しながら演奏を行うことが最も重要である。演奏中にそのような処理をリアルタイムで行うため、ロボットの演奏制御にはデータサイズが小さくテンポなどの編集も容易な MIDI ファイルを用いている。

本研究では、対象となる自動演奏ロボットとしてサクソフォン自動演奏ロボットを選択した。このロボットは MIDI データとして与えられた演奏情報に従って楽曲の自動演奏を行うシステム³⁾となっており、その演奏テンポは MIDI ファイルに基づいた一定の値である。そこで、演奏制御プログラムに変更を加え、自動演奏ロボットが協

調演奏を行うための基本技術として、演奏中に外部からの操作によって演奏テンポを変更するシステムを開発した。

2. 協調演奏

ロボットの演奏に人間が合わせて合奏を行う場合、人間の演奏特有の表現を損なうことが考えられる。人間の演奏にロボットが協調することでそのような表現を保ったまま合奏を成立させることが可能と考え、周囲の演奏音から随時テンポを取得しロボットの演奏に反映する協調演奏システムを考案した。

(1) 人間による演奏

人間の演奏には、「少し早い」「少し遅い」といった人間が持つリズム感によるゆらぎが反映される。一方でロボットの演奏にはそれがなく、一定のテンポに基づいた自律演奏が行われるため、この点において人間による演奏との乖離が生じていると考えられる。ゆらぎを人間特有の演奏表現のひとつと考えた場合、人間がロボットに合わせる同時演奏ではこれが失われてしまう。また、特に後述する多人数での合奏を想定した場合、ロボットの演奏を基準としてしまうと、人間の全奏者がロボット

の演奏を正確に聞き分けてそれと判断し同期しなくてはならず、本来人間同士で行われる合奏における指揮やインザツ出し等の視覚的補助が失われたきわめて不自由な演奏を強いられる。

(2) 多人数による合奏

大人数による合奏を行うことを想定した場合、ロボットがどの音に対して協調を行うかを想定することは必須である。バンド全体の演奏音を1機のマイクロフォンで集音することを考えた場合、音源分離の過程が必要になり、情報処理量が増加してしまい、指揮者が存在する場合でも視覚的情報から演奏に反映させられるレベルの正確なテンポを分析するのには処理時間が多くかかる。したがって、特に大人数での合奏の場合には処理情報が最も簡潔である「定期的なビートを発音する打楽器」を協調基準とすることを想定した。

(3) テンポ取得

周囲の演奏環境から演奏テンポを取得する方法に関しては、発音時刻を取得しテンポ算出を行うという竹内による実験がなされている¹⁾。しかし、テンポ取得の精度は高くなく、想定する演奏環境も限定的なものであった。より高度なテンポ取得のためには集音対象となる楽器の数を増やし、取得結果を平均化、平均値から著しく離れた値は除外といったプロセスが必要になると考えられるが、情報処理量が増加するためリアルタイムでの演奏制御に適さなくなる恐れがある。

したがって本研究においては、打楽器の発音時間取得およびテンポ算出の代替として、制御PCのキーボードで一定間隔の打鍵を行うことによって擬似的にビートを表現する暫定的手法を用いて演奏実験を行った。

3. サクソフォン自動演奏ロボット⁴⁾

本研究で使ったサクソフォン自動演奏ロボットの外観をFig.1に示す。



Fig.1 External view of the automatic saxophone robot

このロボットは、人間がサクソフォンを吹鳴する際に使用している口、呼吸器系、指の3つの器官の役割を人工口顎部装置、送気システム、運指機構に置き換えることで人間と同様にサクソフォンを吹鳴することが可能となっている。インターフェースとして、送気圧制御装置

と人工口顎部装置内のリード圧制御装置およびタンギングパッドの制御にD/A変換ボード、運指機構の制御にDOボードを使用している。ロボット全体の制御の流れをFig.2に示す。

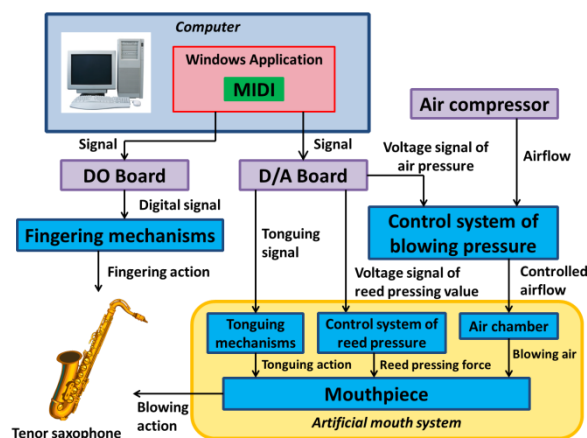


Fig.2 Global schematic diagram of the system

このシステムに周囲の演奏音からテンポを取得し反映させる協調演奏システムを加えたシステムの概略をFig.3に示す。なお、この概略においてはロボット自身の演奏を検出対象として想定していないため、Fig.3における制御もFig.2のものと同様に開制御であることに留意されたい。このような制御を想定した理由としては、協調演奏システムの目的が人間の演奏に協調することだからであり、あくまで合奏の成立は両者の演奏状況のデータの合致ではなく聴衆の感覚的な判断にゆだねられるものであるとしている。

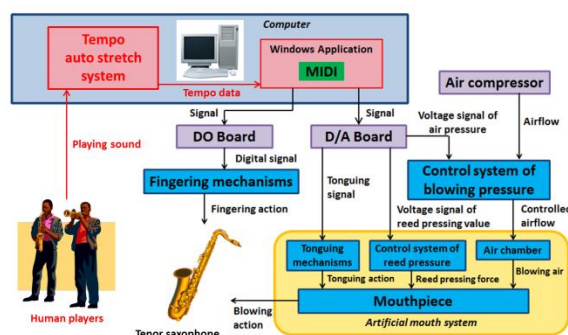


Fig.3 Diagram of the cooperative performance system

4. MIDI規格⁵⁾⁶⁾

(1) MIDI規格の概要

MIDI(Musical Instrumental Digital Interface)とは、電子楽器やコンピュータなどを用いて音色などの情報伝達を行う通信方法の世界共通規格である。

MIDIに対応している電子機器同士をMIDIケーブルで接続し演奏情報を伝える。MIDIデータは波形で表される音声データではなく信号で表されるバイナリデータであり、これだけでは音を鳴らすことはできず、音源定義フ

ファイルと呼ばれるデータがあつて初めて再生できる。その分、WAVE ファイルなどに代表される音声ファイルに比べてデータサイズが非常に小さく、信号の種類を判別するだけで波形解析などの操作を必要とせずに音の判別を行えるため、自動演奏ロボットの制御に適していると考えられる。

(2) MIDI メッセージ

MIDI が伝達する情報を MIDI メッセージといい、演奏動作を信号の形式で伝達する。この信号の種類によって音符や休符、その他の演奏情報の識別を行うようになっており、情報を任意に編集して伝達することで簡単に音色の選択や編集が可能にもなっている。

MIDI メッセージは通常、バイト単位での処理となるため 16 進数で表記され、メッセージの種類を表すステータス・バイトが 1 バイト、メッセージの内容を表すデータ・バイトが 1 バイトから数バイトの 2 種類によって構成される。ステータス・バイトが「音を鳴らす」「ビブラートをかける」等といった演奏動作そのものの種類を決定し、必ず MIDI メッセージの先頭に置かれる。ステータス・バイトの後に続いて表記されるデータ・バイトが「どの音階の音を鳴らすか」「どれほどの音量で鳴らすか」といった演奏動作の内容を示し、メッセージの種類によって指定しなければならない値の数が決まっている。

(3) MIDI チャンネル

MIDI にはチャンネルという概念が導入されている。各々の楽器をそれぞれチャンネルに振り分けることで各演奏情報を混同することなく情報のやりとりを行う。MIDI チャンネルは 16 個まで個別に設定でき、演奏情報、音色情報をテレビ電波のように特定のチャンネルで送受信できる。

(4) SMF (Standard MIDI-file Format)

SMF(Standard MIDI-File Format)は、MIDI の演奏情報を保存する標準データフォーマットである。SMF に対応しているシーケンサー同士であれば、CD-ROM などのメディアや通信を通し、シーケンス・データのやり取りが可能であり、現在ではほとんどのシーケンサーが SMF に対応している。

SMF ファイルは「チャンク」と呼ばれる一連のデータの塊から構成されており、楽曲全体の情報を含むヘッダ・チャンクと実際の演奏情報を格納したトラック・チャンクの 2 種類に分類される。原則、SMF には形式の違いにより 3 種類のフォーマット(フォーマット 0, フォーマット 1, フォーマット 2)が定義されているが、本研究においてはアプリケーションの設定を考慮し、フォーマット 1 のファイルを利用することを想定している。フォーマット 1 とはトラック・チャンクがトラック(≒楽器)の数に分けて表記してある形式のことであり、ヘッダ・チャンク 1 つとトラック・チャンク複数という構造を取る。

a) ヘッダ・チャンク

ヘッダ・チャンクに含まれる情報は、ヘッダ・チャンク

クであるということを表す「Mthd」の識別子、ヘッダ・チャンクのデータ長、すなわち以下何バイトのデータがヘッダ・チャンクに含まれるかという値、SMF のフォーマット、楽曲のトラック数、時間単位(タイム・ベース)の 5 種類である。

b) トラック・チャンク

トラック・チャンクに含まれる情報は、トラック・チャンクであるということを表す「Mtrk」の識別子、トラック・チャンクのデータ長、実際のデータの 3 種類である。データ部分には MIDI メッセージで表される演奏情報である MIDI イベント、MIDI メッセージによって共通化できない機種固有の情報を送受信するためのシステム・エクスクルーシブ・メッセージ、調や拍子の分け方や著作権情報といった演奏内容に直接影響しない情報に相当するメタ・イベントと、それらのタイミングを指定するデルタ・タイムが格納されている。

5. アプリケーション

(1) サックス演奏制御アプリケーション

サクソフォン自動演奏ロボットの制御処理は全て、サクソ演奏制御アプリケーション「sax.exe」上で行われる。Fig.4 にアプリケーションの初期画面を示す。



Fig.4 Initial window display of the application

制御は SMF ファイルを用いて行う。入力された SMF ファイルのイベント内容に基づいて人工口顎部装置、送気圧制御装置、運指機構にそれぞれ送出する電圧信号の羅列を行い、1 曲分の演奏制御情報とする。演奏制御情報の作成を終えてから再生操作を行うことによって、作成された演奏情報通りに自動演奏が行われる。

Fig.4 のメイン画面において、[ファイル]タブによって演奏に用いる MIDI ファイルに関連した操作を、[チューニング]タブによって D/A ボードに送出する各音の電圧情

報の設定を,[マニュアル]タブによって操作方法の確認をそれぞれ行う。

(2) 操作の手順

アプリケーションの操作手順を以下に記す。

a) チューニング

自動演奏を行う前に、チューニングダイアログによってロボットのチューニングを行う必要がある。[チューニング]タブからダイアログを開く。チューニングダイアログ画面を Fig.5 に示す。



Fig.5 Window display of the tuning-dialog

各音高に対してそれぞれの装置(リード圧制御装置・送気圧制御装置)に送出する電圧値を変えながら適切な吹鳴音を出せる値を探し、その値を保存する操作を、実際の楽器の調律作業に準えてチューニングと称する。ここで設定した値がアプリケーション内で各変数として保存され、実際の演奏に反映される。また、配列として[sax]ファイルに保存しておき、一括で呼び出すことも可能である。

b) SMF ファイルの読み込み

[ファイル]タブメニューの[開く]から演奏に使用する SMF ファイルを選択し開くと、基本的な楽曲情報の読み込みが行われ、画面中央上方のコンボボックスにトラックの一覧が表示される。現状のアプリケーションにおいては、問題なく読み込みおよび自動演奏が可能なファイルは拡張子が[mid]となっているフォーマット 1 の SMF ファイルに限定している。

c) ロボットによる演奏

サクソフォン自動演奏ロボットに演奏させるトラックを選択してから画面中央の[実行]ボタンを押すと、選択したトラックの演奏情報がロボットの自動演奏のための演奏制御情報に変換される。画面下部に新たに表示される[再生]ボタンを押すことでロボットによる演奏が開始され同時に選択したトラックを除いた SMF が再生される。

(3) MIDI のタイマー関連

SMF ファイルにおける時間の計算はデルタ・タイム(単位は Tick)で行われる。Tick は MIDI における時間の最小単位となっており、1Tick が何秒にあたるかを決定するのが前述のヘッダ・チャンク格納のタイム・ベース、およびトラック・チャンク格納のメタ・イベントに分類されるテンポ情報である。

タイム・ベースは分解能とも呼び、4 分音符 1 個が何 Tick にあたるかを定義する。例えばタイム・ベースの値が 480 だった場合、4 分音符が 480Tick, 8 分音符が 240Tick, 2 分音符が 960Tick といった長さになる。

テンポ情報は BPM(Beat Per Minute)で表され、BPM は 1 分間に 4 分音符が何個収まるかを示す値である。BPM 値が 120 であれば、4 分音符 1 個の長さは 500 ミリ秒となる。

すなわち、タイム・ベース 480, BPM120 の楽曲において 1Tick の長さは $500/480=1.042$ ミリ秒である。このように、多くの楽曲において小数点以下の値を取り扱う必要があるため、実際のプログラムにおいては整数型(int)ではなく浮動小数点型(double)変数を用いている。

(4) タイマープログラム

サクソフォン自動演奏ロボットの演奏におけるタイミングの指定は、自作プログラム[wait]関数によって行っている。

従来のプログラムにおいては、この wait 関数によって取り扱う待ち時間はデルタ・タイム、タイム・ベース、テンポの 3 つから算出されたミリ秒の値であり、演奏中にテンポを変更したい場合に計算が複雑化してしまうため、簡単化のために引数をあらかじめ「ミリ秒×テンポ」としておくことで、テンポの値のみを直接変更するだけで演奏テンポが変えられるようなプログラムに修正した。

wait 関数の引数に待ち時間を指定し、関数の呼び出しと同時に現在のカウンタの値を double 型変数「Timer1」に保存してからループに入り、ループの終端で取得したカウンタの値を double 型変数「Timer2」に随時上書き保存し続け、Timer2 と Timer1 の差が引数を超過するかアプリケーション上で特定の操作がなされるまでループを続けさせることで待機処理としている。

(5) マニュアル

演奏中、アプリケーションがアクティブな状態で特定のキー操作を行うことで様々な演奏挙動の変更を行うプログラムを、同修士 2 年の森と共同で開発した。それに伴って、対応するキーの一覧とアプリケーションの操作方法をまとめたガイド画面を[マニュアル]タブによって表示させる機能を追加した。Fig.6 にマニュアルダイアログの表示画面を示す。

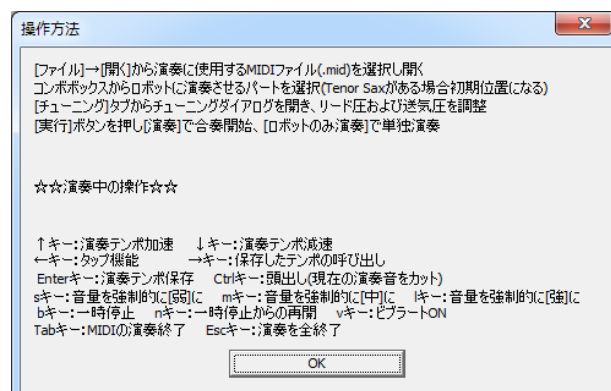


Fig.6 Window display of the manual-dialog

(6) 相対時間と絶対時間

SMF ファイルなどの電子音源に対する同期演奏の確度を高める技術の一環として、絶対時間によるカウントで自動演奏を行うシステムを作成した。

過去サクソフォン自動演奏ロボットに用いられていた Pascal のプログラムにおいては、曲の総演奏時間に対する各音の発音タイミングを絶対的に指定して演奏を行うプログラムであったため、各音の発音および曲の開始時間と終了時間にはタイマー自体の分解能である 1 ミリ秒未満の誤差しか生じなかった。しかし、現在使用している C 言語によるサクソ演奏制御アプリケーションにおいては、SMF の特徴である「イベントごとの間隔をデルタ・タイムで表現する」という仕様に則って相対時間を扱っているため、ミリ秒未満の誤差であっても曲を通して蓄積し、人間の耳でも判別できる程度のズレが生じてしまう恐れがあった。

すべてのイベントの間隔を演奏開始タイミングからの総経過時間としての数値に直し、演奏開始時のカウンタの値を保存、以降全てのイベントにおいてこの値を一律で利用することで絶対時間による自動演奏システムを確立した。演奏実験により相対時間を用いた従来の自動演奏システムとの比較による精度検証を行った結果を、以下の Fig.7 および Fig8 に示す。

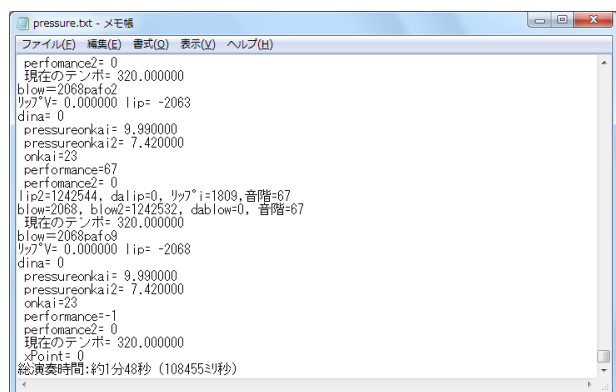


Fig.7 Result of the performance by relative-time count

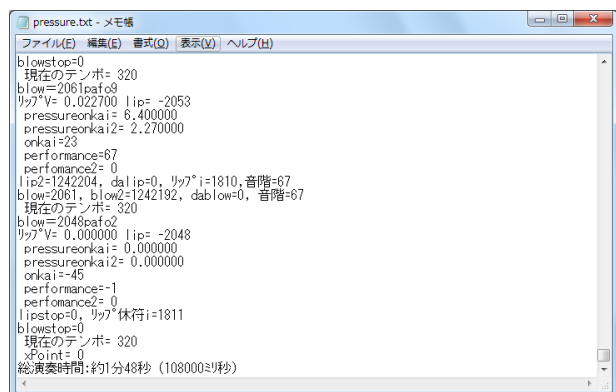


Fig.8 Result of the performance by absolute-time count

図は共に演奏実験ログファイル「Pressure.txt」としてプログラム上で出力した文章であり、相対時間においては

演奏開始から演奏終了までのカウンタの経過時間の総和を、絶対時間においては演奏開始時と演奏終了時のカウンタ値の差を、それぞれ総演奏時間として表示している。使用した楽曲の演奏時間は 1 分 48 秒であったが、相対時間を用いたものではミリ秒未満のズレが蓄積していたのに対し、絶対時間では演奏終了時点でのズレがミリ秒未満となっているためちょうど 1 分 48 秒(108000 ミリ秒)となっている。

こうして求められる確度とはあくまでゆらぎのない音源との同時演奏に対するものであり、人間の演奏に対して同期を行う場合はその限りではない。また、MIDI の音源や PC 上で再生する音声ファイルに関しても、機器の性能などによっては定められた精度が必ずしも保証されるわけではない。そのため、演奏テンポを変更するプログラムはおおよそどのような演奏形態においても必要になると考える。

6. テンポ変更プログラムと演奏実験

(1) テンポ変更プログラムの概要

演奏中にテンポを変更できるよう新たに追加した機能をテンポ変更プログラムと総称する。これらは本研究の目的である協調演奏のための基本技術として作成した。

5 章 4 節において述べたキー操作による機能のうち、加速機能、減速機能、タップ機能、カット機能、ポーズ・再開機能、テンポ保存機能、テンポリセット機能がテンポ変更プログラムにあたる追加機能である。手動の操作を必要とするものの、外部からの入力情報に応じて随時演奏テンポを変えながら自動演奏を行うことが可能なシステムとなっている。

以下に、開発した機能の詳細を記述する。

a) 加速機能

加速機能は、演奏テンポを一定量早める機能である。

ロボットの自動演奏中に、アプリケーションがアクティブな状態で制御用 PC のキーボードの[↑]キーを押下すると、ロボットの自動演奏テンポの BPM を 1 増やす機能である。感覚的に「少し遅い」と感じた場合に用いる機能だが、テンポの速い曲と遅い曲とでは BPM1 の差によるテンポの変わり方に差があり、速い曲では人間の耳で聞いて判別できるほどの違いではない場合がある。そのため、キー押下 1 回分の BPM 変更レンジを大きくすることも容易なプログラムにしてある。

元のテンポの値の 2 倍の値を超える場合、原曲を大きく逸脱するものとして加算をキャンセルする上限設定を設けている。ただし、元のテンポの値とは SMF ファイルに設定された BPM の値を指す。

b) 減速機能

減速機能は、演奏テンポを一定量遅くする機能である。

ロボットの自動演奏中に、アプリケーションがアクティブな状態で制御用 PC のキーボードの[↓]キーを押下すると、ロボットの演奏テンポの BPM を 1 減らす機能で

ある。基本的には加速機能と同様の用途を想定しているが、BPM の性質上、BPM を 1 増減した時に変化する 1 拍あたりのミリ秒数の変化量は、加算時より減算時の方が大きいので、1 回分の BPM 変更レンジを加速機能と同じ値にすることは必要ではない。

なお、加速と減速でそのような違いが生じるのは、BPM と 1 拍のミリ秒との関係式が反比例型の非線形をなしているためである。ミリ秒の方で線形性を保つのは、変更後のミリ秒から(60000/変更後のミリ秒)によって求めた BPM に直接変更することで可能である。例えば BPM120 の曲で 1 拍あたりの長さが 100 ミリ秒だけ減るような加速を行いたい場合、元のテンポにおける 1 拍分の長さが 500 ミリ秒であるから、100 ミリ秒減らした 400 ミリ秒を 1 拍とする(60000/400)で得られる BPM150 が変更後のテンポである。

現状の手動操作においては、どちらがより加減速機能として適切かどうかは操作者の感覚に全て委ねられる。通常は両者の BPM 変更形態を加減算に、変更レンジを 1 に規定しておき、操作者が実験に応じて使いやすと思われる値を模索し設定を変更する作業が必要となる。

c) タップ機能

タップ機能は、任意のタイミングで複数回カウントを取るとその間隔からテンポを算出する機能である。市販のメトロノーム等に附属しているものが多い。

ロボットの自動演奏中に、アプリケーションがアクティブな状態で[←]キーを押下すると、タイマーからカウント値を取得する。連続して[←]キーを押下することでキー押下の間隔をミリ秒で算出し、前項にて記述した計算方法によって(60000/取得したミリ秒)で得られたテンポに直接変更する。

d) カット機能

カット機能は、同期対象と音の頭(発音タイミング)を合わせるための頭出し機能である。

演奏中にアプリケーションがアクティブな状態で制御 PC のキーボードの[Ctrl]キーを押すと、現在演奏中のイベントを強制終了し次のイベントに移行する。タップ機能により BPM を合わせた後、発音タイミングがずれていた場合、同期対象が次の音を発音するのに合わせてキーを押下することで発音タイミングを合わせることが可能である。ただし、この機能で発音タイミングを合わせることができるのは、自動演奏ロボットの演奏が同期対象の演奏よりも遅れている場合のみである。

e) ポーズ・再開機能

ポーズ・再開機能は、カット機能と同様に発音タイミング合わせを行うための機能である。これらはカット機能に対し、自動演奏ロボットの演奏が同期対象の演奏よりも先走っている場合に用いる。

ポーズ機能は、演奏中にアプリケーションがアクティブな状態で制御 PC のキーボードの[b]キーを押すと、現在演奏中のイベントを保持したまま待機する機能である。

ポーズ中に[n]キーを押すことで演奏を再開する。[n]キーを押した時点でカウンタの差が wait 関数の引数を超過していなかった場合は再開時点からカウントを再開、超過していた場合はキー押下と同時に次のイベントに移行する。すなわち、再開キーをカット機能と同様に用いることが可能である。

f) テンポ保存・リセット機能

テンポ保存・リセット機能は、上記のテンポ変更プログラムを利用して合わせた演奏テンポを保存、呼び出しできるようにした機能である。

テンポ保存機能は、演奏中にアプリケーションがアクティブな状態で制御 PC のキーボードの[Enter]キーを押すと、現在の BPM を保存することができる機能である。また、キーボードの[→]キーを押すと、ロボットの演奏テンポをただちに最新の保存テンポに変更できる。テンポ保存を一度も行っていない場合、[→]キーによって呼び出される値は SMF ファイルから読み込まれた原曲の BPM であるため、この機能をリセット機能と称している。

(2) 同期演奏実験

前節で述べた機能の効果を評価するため、人間のオーケストラによる演奏の音源にサクソフォン自動演奏ロボットの演奏を同期させる実験を行った。

実験の方法としては、人間のオーケストラによって演奏されたことがわかっている楽曲を Web 上で用意し、その主旋律を SMF ファイルとして作成したものをを用いて自動演奏を行う。原則として、演奏開始の発音タイミングを合わせるときだけカット機能を用い、以降はタップ機能のみを用いて毎拍カウントを取り、テンポ変更の推移を逐次記録したデータを Excel 上で「タップ回数・BPM」のグラフにして表示した。これはカット機能による発音時刻の変動がテンポ推移に反映されないのと、別のキーを押すことでタップのタイミングがずれるおそれがあると考えたためである。

テンポの算出方法は以下の 2 通りを考案し、それぞれの方法で実験を行いテンポ推移の状況を比較した。

①常に最新の 2 回分を参照

②最初の 2 回は間隔をそのまま、3 回目より 1,2 回目の間隔と 2,3 回目の間隔を平均、4 回目以降は最新の 4 回分の間隔を平均

考案段階でのねらいは、①の方法では即応性を重視し、1 度のタップで大きくテンポが変わることを想定している。②の方法では確度を重視し、多少のタップの崩れでは大きくテンポが変わることがないようにしてある。

実験に使用した楽曲は、SMF ファイルではないため正確な BPM は不明であった。そのため、サクソフォン自動演奏ロボットによる同期演奏とともに録音したファイルをフリーの波形解析ソフト「WaveTone」で解析し、音量のピーク周期からテンポを算出したところ、112.97bpm であった。解析画面を Fig.9 に、実験結果のグラフを Fig.10 および Fig.11 に示す。

なお、複数回の実験を行って再現性を取ることをしていないのは、同期対象が既に録音された音源だからである。繰り返しのによって得られるのは操作者のリズム感の特性となってしまうため、1回ずつの実験とした。

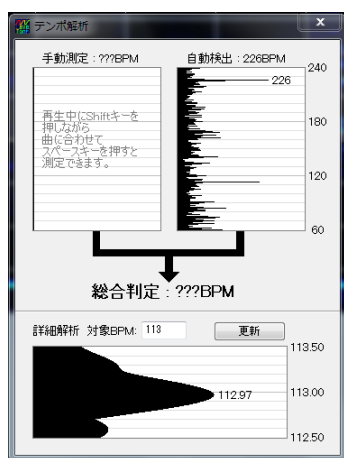


Fig.9 Analyzed Result by the wave analyzer “WaveTone”

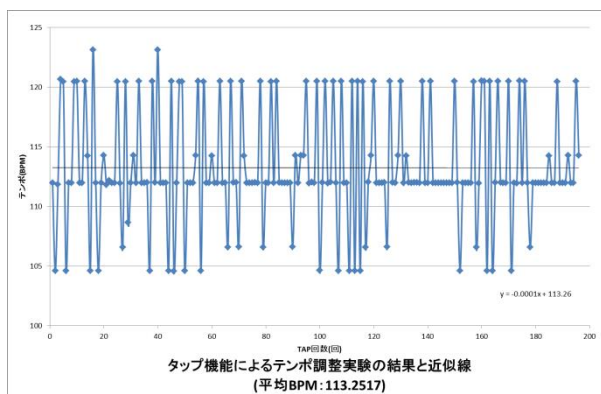


Fig.10 Result of the tempo adjustment experiment by tap-function (using average of recently twice tap interval)

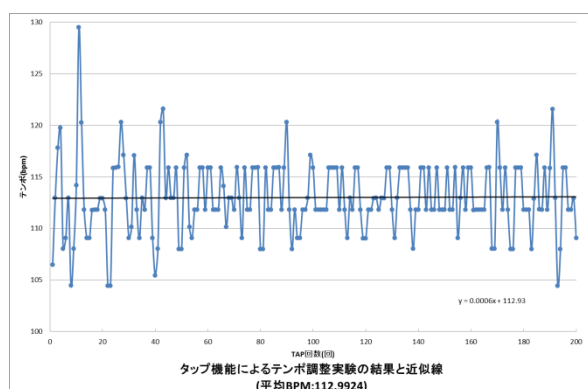


Fig.11 Result of the tempo adjustment experiment by tap-function (using average of recently four times tap interval)

いずれの結果においても、113bpm 付近を中心に上下に振れるようにテンポの推移が行われていることがわかる。

①の方法による結果の特徴としては、タップ1回でのテンポ移動幅が大きいいため、振れ幅は上下に15～20bpmほどとなっているが、その分ずれたテンポを修正するの

も容易である。一方、②の方法による結果の特徴としては、1回1回の振れ幅は①に比べて小さく、平均値も解析したテンポとほぼ一致した値を取っているが、タップ1回での大きなテンポ移動が見込めないため、テンポがずれた場合の修正に時間がかかり、結果として曲後半にロボットの側の演奏に遅れが見られた。

操作者としての体感的な感想としては、タップ間隔がそのまま実際の演奏テンポに現れる①の方法がよりテンポ修正に優れていたという印象を受けたが、使用する楽曲やタップ方法によってはこの限りではないと考えられる。例えば、ゆったりとしたリズムの曲を用いる場合や4分音符ではなく2分音符でタップを行う場合などタップ間隔が大きくなる時には1回1回のずれが同期にもたらず影響とテンポ修正にかかる時間もより大きくなるため、同期演奏に使用する楽曲によってこれらの手段を使い分けることが重要であると考えられる。

7. 同期演奏の自動化に向けての考察

実験結果のグラフから、人間がテンポを合わせようと試みた場合の打鍵のタイミングは、平均値を中心としてほぼ交互に現れていることから、「少し遅い」と感じればタップ間隔は早まり、「少し早い」と感じればタップ間隔は遅まるという意識が働いているということがわかる。このような人間的な感覚をアルゴリズムに落とし込むことができれば、同期演奏の自動化への足掛かりになるのではないかと考えられる。

また、今回行ったタップ機能による同期演奏実験では、参照ビートを打鍵による直接の入力としたが、竹内の想定したテンポ算出プログラムの精度を高めれば「マイクの前で手拍子」したビートをタップと同様に参照することが可能であり、それは即ち打楽器に対する同期、外部演奏環境との協調演奏となると考えられる。今後は、より少ない情報から最小限の処理で正確なテンポを算出するプログラムをロボットの自動演奏と同時にリアルタイムで行うシステムの実現が望まれる。

8. 結論

本研究によって、打鍵からのテンポ算出と演奏への反映は、タップ機能をはじめとしたテンポ変更プログラムの開発によって手動操作であれば行えることがわかった。

また、本研究における実験はすべてサクソフォン自動演奏ロボットを用いて行ったが、当研究室の自動演奏ロボットは他にもトランペット自動演奏ロボット、トロンボーン自動演奏ロボット、尺八自動演奏ロボットなどがあり、いずれも制御PCの移行にあたってサクソ演奏制御アプリケーションを基にした演奏制御プログラムの開発が進められていたため、今回追加したプログラムは将来的には全てサクソフォン以外の自動演奏ロボットにも適用することが可能となる。また、周囲の演奏からテンポを随時取得し、テンポ変更プログラムによって演奏テ

ンポに反映させることができれば、協調演奏システムの完成の足掛かりとなると考えられる。

本研究にて言及した「外部演奏環境」について、主に人間の楽器奏者による演奏を取り扱うものとしてきたが、人間と複数の自動演奏ロボットで合奏を行う場合においては、協調演奏システムがすべての自動演奏ロボットに導入されていれば問題なく合奏が可能であると考ええる。ロボット同士の合奏においては、互いの演奏はズレなく正確に行われると考えられるため、絶対時間によるタイマーを用いて演奏開始時間のみを合わせればよい。しかし、これらの合奏が成立しているかどうかは、数値やデータではなく人間の聴衆の感覚的な判断に委ねられるところが大きいため、協調演奏システムを用いて合奏を行おうと試みた場合には、実際にオーケストラの中で繰り返し演奏実験を行い、聴衆が感覚的に受け入れられる音楽かどうかを調査する必要があると考えられる。

謝辞：本研究を行うにあたり終始、数多くのアドバイスをはじめとした格別のご指導ご鞭撻を賜りました法政大学理工学部機械工学科 高島 俊教授には、この場を借り深く感謝の意を表すとともに、ここに厚く御礼申し上げます。また、本研究に類する研究を行ってきた高島研究室の同期・後輩の皆様のご助力にも感謝します。

参考文献

- 1) 竹内慧 “自動演奏ロボットの高度な演奏制御-MIDI 入力装置による制御と自律合奏のためのシステム-” (2012),法政大学院工学研究科機械工学専攻修士論文.
- 2) 宮脇毅“サクソフォン自動演奏ロボットの制御-スタンダード MIDI ファイルを用いた演奏制御-” (2003), 修士論文, 法政大学大学院工学研究科機械工学専攻.
- 3) S. Takashima,“Control of an Automatic Performance Robot of Saxophone: Performance Control Using Standard MIDI Files”, IROS 2006 Workshop(Beijing) : Musical Performance Robots and Its Applications, PP30-35,2006.
- 4) S. Takashima and H. Shimoda, “Automatic Performance Robot of Wind Instruments –The Case of Tenor Saxophone-,” Proceeding of JSME Int. Conf. on Advanced Mechatronics, pp64-69, Aug. 1993
- 5) 中島安貴彦 “MIDI パイプ I MIDI 1.0 規格 基礎編” (1996), 株式会社リットーミュージック
- 6) 高橋信之 “コンプリート MIDI ブック” (2005), 株式会社リットーミュージック