音響信号の MIDI 符号化ツール「オート符」の Windows10 対応に伴う改修

Improvement of an Audio-MIDI Encoder Tool “Auto-F” for Supporting Windows 10 Platforms

茂出木　敏雄
MODEGI Toshio

[抄録]

既開発の音響信号から MIDI 符号に自動変換するツール「オート符」は、一般化調和解析に基づく周波数解析を採用することにより倍音を含む和音を高精度に解析できるという特徴があり、楽器音に限らずボーカルなど、標準的な MIDI 音源で近似的に原音響信号を再現可能な MIDI データを生成できる。このソフトウェアは、1996 年に Win32 アプリケーションとして Windows NT/95 上で開発に着手し、2001 年より一般財団法人デジタルコンテンツ協会のホームページより約 14 年間無償配布を行いながら、改良を重ねてきた。その間、Microsoft の Windows OS は数々のアップグレードがなされてきたが、本ツールはそれらの影響を受けず、最新 OS の Windows 10/64bits 版まで問題無く稼働し続けてきた。しかし、2016 年夏に行われた ”Windows10 Anniversary Update” により、MIDI 再生系に不具合が生じるようになった。これに対し、アプリケーション側に改良を加え、従来通り安定に稼働するように復旧させることができた。本稿では、本問題の調査結果と回避策について報告する。

キーワード：
MIDI 符号化、MIDI 再生、Windows10 アップデート、MIDI 音源 , ソフトウェア障害、オート符

[Abstract]

Our previously developed audio to MIDI-event converter tool “Auto-F” has a feature of high-precision harmonic tone analysis functions based on the Generalized Harmonic Analysis algorithm. Applying this tool, from given vocal acoustic signals we can create MIDI data, which enable to playback voice-like signals with a standard MIDI synthesizer. We began developing this software as a Win32 application on the Windows NT/95 platform in 1996. Since 2001, we have distributed this software tool for free at the Website of the Digital Content Association Japan for about 14 years, with several improvements added. In these years, Microsoft has released several upgraded Windows Operating Systems, and our developed tool could catch up with these updates and continue running on newer platforms until the latest Windows 10/64bits. However, the “Windows 10 Anniversary Update” released in the last summer, has influenced on the MIDI sequencer function of this tool.

Then, we have improved this tool and we could successfully make its execution stable on this newest platform. In this report, we present results of our tracking of this software defect and our adopted technique for avoiding it.

Keywords:
MIDI encoder, MIDI sequencer, Windows10 Update, MIDI device, software bug, Auto-F

1．はじめに

先報告[1]において、「あらゆる音を音符に変換できる“オート符”」というツールについて言及し、その後の報告[2]において、本ツールの具体的なMIDI符号化アルゴリズムの詳細について紹介してきた。本稿では、最新OSのWindows 10のアップデートにより本ツールのMIDI再生系に生じた問題点とその回避策について報告する。

このソフトウェアは、1995年に筆者が旧郵政省・通信総合研究所（現、国立研究開発法人・情報通信研究機構）に特別研究員として出向していた際に、Windows NT/95上で開発に着手したWin32アプリケーションが基盤となっている[3]。配属された部門の主な研究テーマは遠隔医療で、筆者は聴診器で拾った心音・肺音の病的特徴を損なわずに低ビットレートで伝送する技術開発を担当し、新規な音響符号化技術としてコンピュータミュージックで使用されているMIDI規格[4]に注目した。当時活用が始まった音響符号化規格MP3（MPEG-1/Layer3）より2桁少ないビットレートで病的特徴を損なわずに符号化でき、ナローバンド回線でも十分対応できた。また、この規格に準拠して符号化ソフトウェアを開発すると、復号系の開発は不要で、コンピュータミュージック分野で既に開発・商品化されているMIDIシーケンサ・MIDI音源などをそのまま流用できるという利点がある[5]。

これと類似した機能をもつソフトウェアに自動採譜ツールがあり、本ツールの開発に着手した頃から、鼻歌を入力すると楽譜に変換するツールが既に販売されていた。しかし、自動採譜ツールでは五線譜に変換することが目的で、基本的には単旋律のメロディーしか拾えず、当時流行し始めた携帯電話の着信メロディー制作の支援が主要な用途であった。これに対し、本ツールは、心音などの任意の音響信号をMIDI音源で再現可能なMIDIデータに変換するもので、五線譜では表現できない倍音分布や音量変化（ベロシティ）などの演奏表情を含めてMIDIデータに符号化されるという特徴がある。

その後、筆者の本務先である大日本印刷（株）に戻り、本ソフトウェアをマルチメディア・音楽制作分野に広く活用できるように、符号化対象の拡充に努め、特にMIDIでは再現が困難なボーカルを標準的なMIDI音源で再現できることを目標に開発を進めた[6]。当時から商品化されているProToolsなどのオーディオの統合編集ツールDAWでは、波形オーディオトラックとMIDIトラックの混在編集が可能であり、MIDIトラックのコンテンツを波形トラックに変換転送することは可能である。しかし、逆に波形トラックのコンテンツをMIDIトラックに逆変換して転送することは不可能であった。そこで、筆者は、波形トラックとMIDIトラックの相互変換が可能な統合オーディオ編集ツールへの活用も提案した[7]。そして、一般化調和解析に基づく周波数解析を採用することにより、楽器音の和

音や音声のフォルマント成分を高精度に解析できるようになり、ボーカルを含む楽器音を汎用的なGM規格のMIDI音源を用いてある程度再現可能になった[8]。

そんな折、本ソフトウェア開発が、旧財団法人マルチメディアコンテンツ振興協会（現、一般財団法人デジタルコンテンツ協会）[9]の2000年度の公募事業に採択され、経済産業省の支援を受けながら、フリーウェアとして音楽コンテンツ制作業界の採譜業務等に広く活用いただけるようにブラッシュアップを行った。2001年春から、一般財団法人デジタルコンテンツ協会のホームページにて無償配布を開始し[10]、その後もソフトウェアの改良を重ねながら、約14年間同サイトにて無償配布を進めてきた。丁度この頃より、本学・情報表現学科とも縁ができ、基礎演習Ⅰ（現、マルチフィールド体験演習）という授業の教材として本学情報表現学科の受講生に配布しながら、述べ16年間使用してきた。その間、MicrosoftのWindows OSは、Windows 98, Me, 2000, XP, Vista, 7, 8, 8.1と、数々のアップグレードがなされてきたが、本ツールはそれらの影響を受けず、最新OSのWindows 10/64bits版まで問題無く稼働し続けてきた。しかし、2016年夏に行われた”Windows10 Anniversary Update”により、MIDI再生系に不具合が生じるようになった。これに対し、原因を究明しアプリケーション側に改良を加え、従来通り安定に稼働するように復旧させることができた。本稿では、本問題の調査結果と回避策について報告する。

2．音響信号のMIDI符号化ツール「オート符」の概要

2．1．　音響信号のMIDI符号化ツール「オート符」の全体構成

図1は筆者らが開発してきた音響信号のMIDI符号化ツール「オート符」の全体構成を示す[10]。主要機能は、左上に示す「1）WAV形式PCMデータ」を左下に示す「3）SMF形式MIDIデータ」に自動変換するものである。はじめに、ソース音響信号として左上に示す「1）WAV形式PCMデータ」ファイルが「6）PCMデータ構造体」に読み込まれる。WAV形式はMicrosoft社のフォーマットであるが、業界標準として各社の音響ツールで入出力がサポートされている。この段階で、「13）WAVデータグラフィック表示」および「17）ディスプレイ」により、読み込まれたPCMデータを波形で可視化でき、「12）サウンドプレーヤ」および「16）サウンドカード」を介して音響再生できる。「16）サウンドカード」は一般的なパソコンに内蔵されているが、再生音にHDDノイズの混入を防ぐためUSB接続の外付けのハードウェアが使用されることもある。

次に、本ツールのメインであるMIDI符号化処理の流れを説明する。この処理は2段階あり、第1段階として「9）周波数解析処理」を実行する。これは、「6）PCMデータ構造体」に記録されているPCMデータを、「21）周波数解析パラメータ指定」により指定された解析フレームに分断しながら時系列に周波数スペクトルを算出し、「7）周波数解析データ構造体」に記録する。周波数解析は、MIDI規格の128種のノートナンバーに対応する周波数に対する強度分布を指定された時間間隔で時系列に生成する。記録されたデータは、「2）FFT形式スペクトル」という本ツール独自の中間ファイルにも保存でき、次回に同一のソース音響信号を基に、MIDI符号化処理を行う際、比較的処理時間を要する「9）周波数解析処理」を省略できる。

続いて、第2段階として「10）MIDI変換処理」を実行する。これは、「7）周波数解析

データ構造体」に記録されている周波数スペクトルの中で、比較的強度が大きい周波数成分を時系列に連結させ音符を構成し、MIDI 規格のノートオン／ノートオフイベントに変換し、「8）MIDI データ構造体」に記録する。音符を構成する条件は、「20）MIDI 変換パラメータ指定」で設定するが、「11）MIDI 編集処理」および「19）MIDI 編集条件指定」により、変換された音符に対して対話形式に修正を加えることもできる。

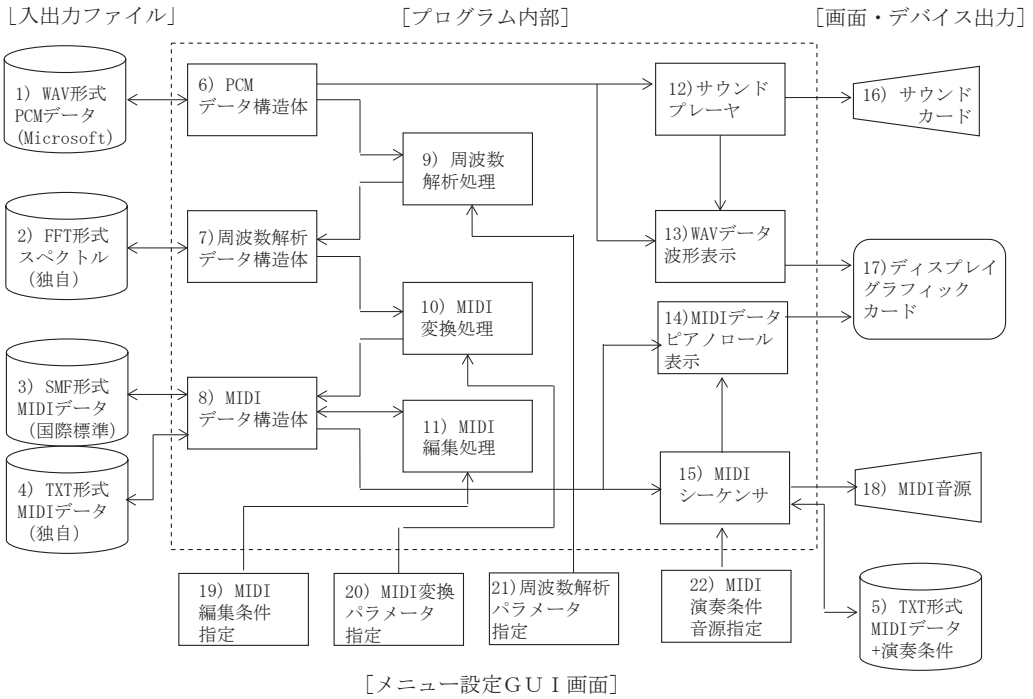

図１　音響信号の MIDI 符号化ツール「オート符」の全体構成

「8）MIDI データ構造体」に記録されたデータは、「3）SMF 形式 MIDI データ」または「4）TXT 形式 MIDI データ」ファイルに保存できる。前者は、国際標準の Standard MIDI File で各社の音楽制作ツールで入出力がサポートされているが、バイナリ形式のため中身を直接読めず加工しにくい。そこで、本ツール独自に ASCII 形式の「4）TXT 形式 MIDI データ」でも保存できるようにした。「8）MIDI データ構造体」に記録されたデータは、「14）MIDI データピアノロール表示」および「17）ディスプレイ」によりピアノロールで可視化できるとともに、「15）MIDI シーケンサ」および「18）MIDI 音源」で音響再生できる。この音響再生時には、「22）MIDI 演奏条件音源指定」により音色やテンポ等を変更できるのが、前述の PCM データの音響再生と異なる点である。「18）MIDI 音源」は、本ツールの開発当初は USB 経由でパソコンに外付けする専用ハードウェア形態であったが、現在ではウェーブテーブルを用いたソフトウェア MIDI 音源が主流である。これは、MIDI 音源の機能がソフトウェアとしてパソコン内にインストールされ、前述の PCM データと同様にパソコン内蔵の「16）サウンドカード」を用いて音響再生を行う。「15）MIDI シーケンサ」で制御する対象のデータは、「5）TXT 形式 MIDI データ + 演奏条件」にファイル保存することができ、これは本稿の主題である改良項目で使用するもので、詳細は後述する。

２.２.　本ツールの MIDI 符号化処理系の概要

図２は、本ツールのメインである「9）周波数解析処理」と「10）MIDI 変換処理」における処理内容を説明したもので、具体的な詳細は文献 2）に記載されているが、その後に改良を加えた箇所もあり[11]、本稿ではそれを含めて概要を説明する。

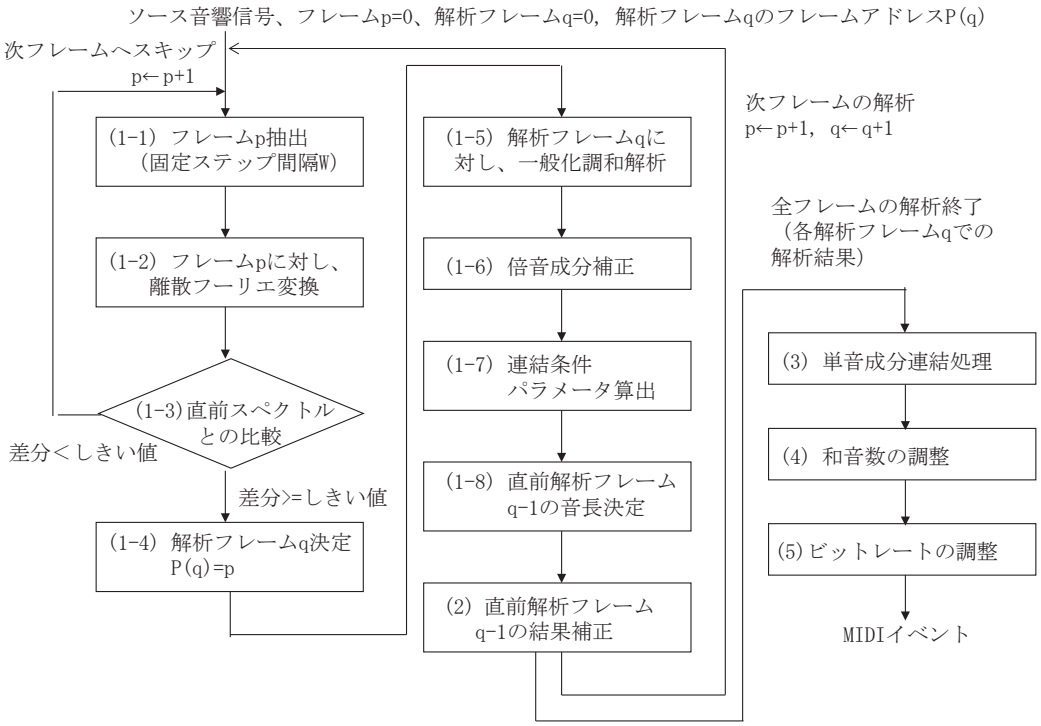

図２　本ツールの MIDI 符号化処理系のソフトウェア構成

まず、与えられたソース音響信号に対して、音響信号の変化が比較的大きい箇所を時系列に探索して、解析フレームを設定する。具体的には、「(1-1) フレーム p 抽出」により、固定ステップ間隔 W で固定の解析フレーム長のフレーム p を抽出する。「(1-2) フレーム p に対し、離散フーリエ変換」の処理を行ってスペクトルを算出し、「(1-3) 直前スペクトルとの比較」により直前フレーム p-1 に対して離散フーリエ変換を行った直前スペクトルと比較する。この差があまり大きくなければ、「(1-1) フレーム p 抽出」に戻り次のフレーム p+1 に対して同様な処理を行い、この差がある程度大きければ、「(1-4) 解析フレーム q 決定」により、抽出したフレーム p を解析対象の解析フレーム q に設定する。

次に、「(1-5) 解析フレーム q に対し、一般化調和解析」を施す。この処理の詳細は文献 2）に譲るが、離散フーリエ変換を繰り返し行って複数の周波数成分を 1 つずつ抽出する方法で、和音が高精度に算出できる。そして、スペクトルに対して必要に応じ「(1-6) 倍音成分補正」を行い、直前の解析フレーム q-1 のスペクトルの対応する周波数成分どうしで連続性があるか否かを判断するため、「(1-7) 連結条件パラメータ算出」を行う。この段階で、直前解析フレーム q-1 から現解析フレーム q までの時間を「(1-8) 直前解析フレーム q-1 の音長」として決定できる。「(2) 直前解析フレーム q-1 の結果補正」は、二重

フレーム解析法と称する文献2）以降に改良を加えた項目で、直前解析フレーム q-1 と現解析フレーム q 間のオーバーラップ成分を補正するものである[11]。周波数解析を高精度に行うためには、解析フレーム長を大き目に設定する必要があり、隣接する解析フレーム間隔に比べて顕著に長い。そのため、隣接する解析フレームから算出されるスペクトルにはかなりオーバーラップ成分が含まれている。これは時間分解能の低下につながり、ボーカルなど周波数変化の激しい音響信号を明瞭に再現することが難しくなる。そこで、各々算出されるスペクトルどうしで AND 演算を行い、直前解析フレーム q-1 の周波数成分より、この AND 演算された周波数成分を差し引く補正を行う。補正された周波数成分は、直前解析フレーム q-1 と現解析フレーム q との時間差に対応する正味の周波数成分に対応し、時間分解能が向上する。

ここまでの処理が、「9）周波数解析処理」に対応し、これ以降の「（3）単音成分連結処理」、「（4）和音数の調整」、「（5）ビットレートの調整」は「10）MIDI 変換処理」に対応する。「（3）単音成分連結処理」は、「（1-7）連結条件パラメータ算出」で得られた条件に基づき、周波数ごとに隣接する解析フレームを順次連結させ音符を構成する。その際、ピッチベンド奏法等を再現する場合には、微分音成分を考慮し周波数の変動を加味して連結させる必要があるが、詳細は文献2）に譲る。構成された各音符は、MIDI 規格のノートオン／ノートオフイベントに符号化されるが、使用する「18）MIDI 音源」の再生能力に合わせて「（4）和音数の調整」および「（5）ビットレートの調整」を行う必要がある。

３．音響信号の MIDI 符号化ツール「オート符」におけるサウンド再生系の構成

後述するように、Windows10 Anniversary Update により本ツールにおいて不具合が生じたのは MIDI サウンド再生系であるが、はじめに本ツールに実装されているサウンド再生系全般について説明する。本ツールには、図１の「12）サウンドプレーヤ」および「16）サウンドカード」に対応する WAV サウンド再生系と、「15）MIDI シーケンサ」および「18）MIDI 音源」に対応する MIDI サウンド再生系の２種のサウンド再生系があり、各々異なるレベルの Win32API（Windows 32bits Application Programming Interface）を使用している。

３．１．　本ツールの WAV サウンド再生系

図３は本ツールの WAV サウンド再生系の構成で図１の「12）サウンドプレーヤ」に対応する。本ツールは、定常状態では「a）操作指示待ち状態」にあり、ユーザからのメニュー指示を待っている。ここで、WAV 再生指示を受信すると、「b）WAV 再生スレッド起動」が実行され「c）MCIWnd 再生スレッド」を起動し、再び「a）操作指示待ち状態」に戻る。Win32API を用いた WAV サウンド再生方法には、最も低水準の waveOutWrite（）関数を用いる方法から最も高水準の sndPlaySound（）関数まで各種あり、更に音声・動画などマルチメディア機能を統一させたインタフェースでアクセス可能な高水準の MCI（Multimedia Control Interface）と呼ばれる関数群が用意されている[12) 13)]。これには、MCIWnd（MCI Window）と呼ばれ MCI 関数を用いて録音・録画・再生・停止ボタンや進捗バー表示などの対話操作 GUI を付加した高水準なマルチメディアプレーヤ部品も提供されている。本ツールの WAV サウンド再生では、MCIWnd 関数を用いており、この

MCIWnd の進捗バー表示機能を用いて「13）WAV データ波形表示」に重畳させている。

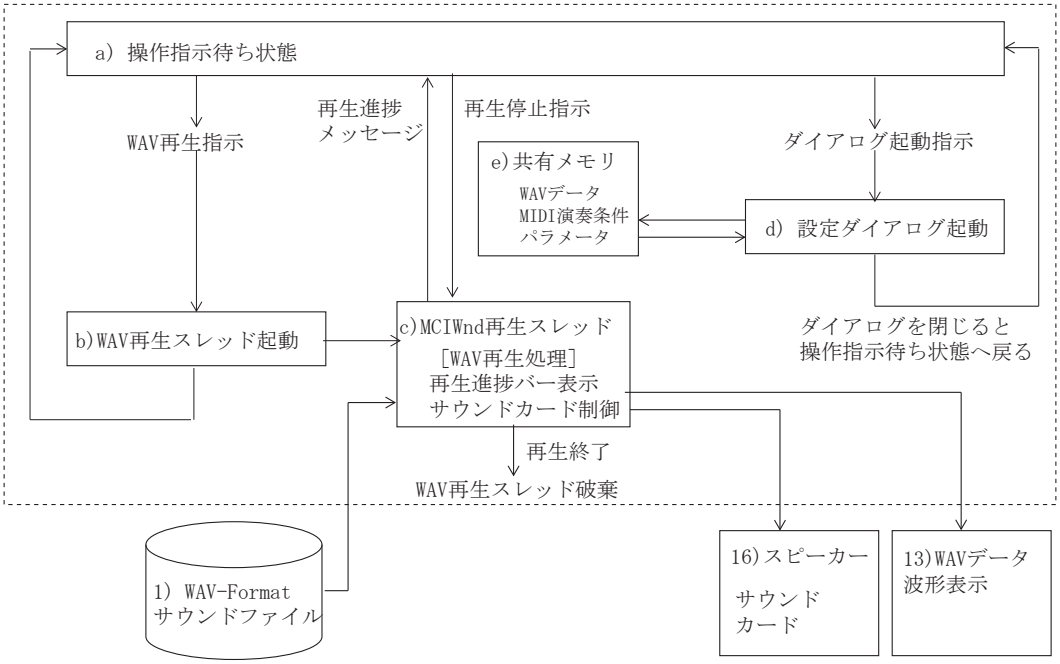

図３　本ツールの WAV サウンド再生系のソフトウェア処理構成

MCIWndCreate（）関数を用いてマルチメディアプレーヤのウィンドウを生成するが、図４に示されるように、ウィンドウの横幅を本ツール画面に合わせ、本ツール画面の波形表示部の上端にウィンドウを配置してマルチメディアプレーヤの進捗バーだけが表示されるように、横長のウィンドウを設定した。マルチメディアプレーヤには再生・停止ボタンが付いており、これ単体で直接操作が可能であるが、本ツール GUI の再生・停止メニューから間接的に操作できるようにした。WAV 再生指示を受けると、前述の MCIWndCreate（）関数によりマルチメディアプレーヤのウィンドウを生成し、MCIWndOpen（）関数を用いて再生対象の「1）WAV-Format サウンドファイル」を指示し、MCIWndPlay（）関数で再生させる。再生開始後は「a）操作指示待ち状態」に戻り再生終了まで待機状態になるが、その間、指定した間隔（本ツールでは 100msec に設定）でマルチメディアプレーヤから再生進捗メッセージが届き、再生終了を受信した場合は、MCIWndClose（）および MCIWndDestroy（）関数で MCIWnd マルチメディアプレーヤを閉じる。また、「a）操作指示待ち状態」で再生停止指示を受けた場合は、随時 MCIWndDestroy（）関数で MCIWnd マルチメディアプレーヤを閉じることができる。WAV 再生終了後は、「a）操作指示待ち状態」より引き続き種々のメニュー指示を出すことができる。「e）共有メモリ」には、あらかじめ「1）WAV-Format サウンドファイル」から読み込まれた WAV データが記録されているが、「c）MCIWnd 再生スレッド」はこのデータを参照せず、「1）WAV-Format サウンドファイル」から読み込みながら再生する。

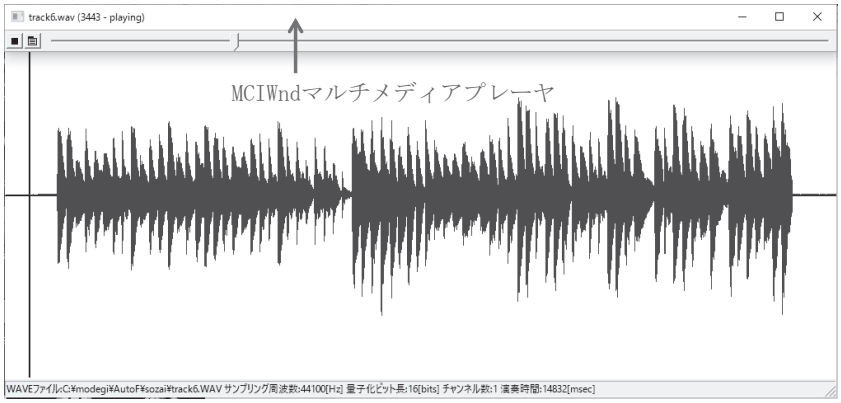

図４　本ツールの WAV サウンド再生時の GUI 画面

３.２.　本ツールの MIDI サウンド再生系

図５は本ツールの MIDI サウンド再生系の構成で図１の「15）MIDI シーケンサ」に対応する。本ツールは、定常状態では同様に「a）操作指示待ち状態」にあり、ユーザからのメニュー指示を待っている。ここで、MIDI 再生指示を受信すると、「b）MIDI 再生スレッド起動」を実行し「c）MIDI 再生スレッド」が起動され、再び「a）操作指示待ち状態」に戻る。一方、起動された「c）MIDI 再生スレッド」は「e）共有メモリ」に記録されている MIDI データや MIDI 演奏条件パラメータを参照しながら再生を開始する。Win32API を用いた MIDI サウンド再生方法には、低水準の midiOutShortMsg（）関数を用いる方法の他に、WAV サウンドと同様に MCI および MCIWnd 関数群を用いる方法がある [12)]。しかし、MIDI サウンド再生時に、図１の「22）MIDI 演奏条件音源指定」により音色やテンポ等を変更できるようにしたい、再生進捗を単純なバー表示ではなく「14）MIDI データピアノロール表示」を再生に同期させるように動的に制御したいという本ツールの設計思想から、MCIWnd 関数群では実現困難と判断し、低水準の midiOutShortMsg（）関数を用いる方法を採用した。

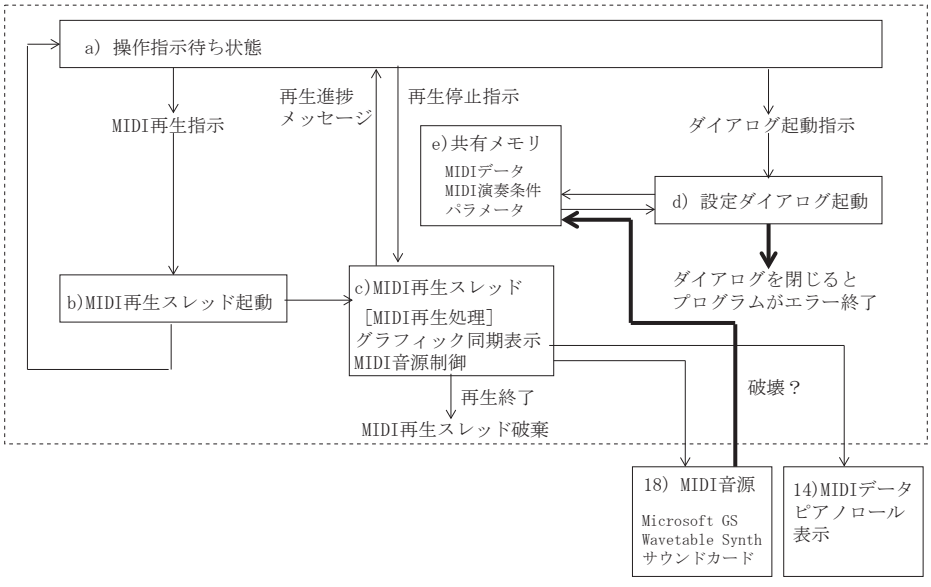

図５　本ツールの MIDI サウンド再生系のソフトウェア処理構成

図６に「c）MIDI 再生スレッド」の詳細構成を示す。主な役割は、midiOutOpen（）関数を用いて「f）MIDI デバイスオープン」を行いデバイス ID で指定した「18）MIDI 音源」とのコネクションを確立後、timeSetEvent（）関数を用いて「g）タイマー設定・開始」により一定間隔（本ツールでは 10msec に設定）で「h）MIDI 音源制御処理」を実行するためのタイマー Callback 関数を起動することにある。デバイス ID を指示するにあたり、WindowsOS にインストールされている「18）MIDI 音源」のデバイスの個数は midiOutGetNumDevs（）関数で、デバイス 属性については、midiOutGetDevCaps（）関数で取得できる。また、「f）MIDI デバイスオープン」を行った後、MIDI イベントデータの送出を開始する前に、「18）MIDI 音源」に対してプログラム（楽器音色）などの MIDI 演奏条件パラメータの設定を midiOutShortMsg（）関数または midiOutLongMsg（）関数を用いて行う。「h）MIDI 音源制御処理」は、「e）共有メモリ」に記録されている MIDI イベントデータを参照するため、この段階で MIDI イベントデータにアクセスするためのポインタを初期化する。

その後は、「c）MIDI 再生スレッド」を一旦抜けてメイン側の「a）操作指示待ち状態」に制御が戻り、再生終了まで待機状態になるが、その間、指定した間隔（本ツールでは 10msec に設定）で再生進捗メッセージが届き、再生終了を受信した場合は、timeKillEvent（）関数を用いて「i）タイマー終了」を実行し、midiOutClose（）関数を用いて「j）MIDI デバイスクローズ」を実行して「c）MIDI 再生スレッド」を破棄する。また、「a）操作指示待ち状態」で再生停止指示を受けた場合は、随時「i）タイマー終了」と「j）MIDI デバイスクローズ」に進み、「c）MIDI 再生スレッド」を破棄する処理を行う。

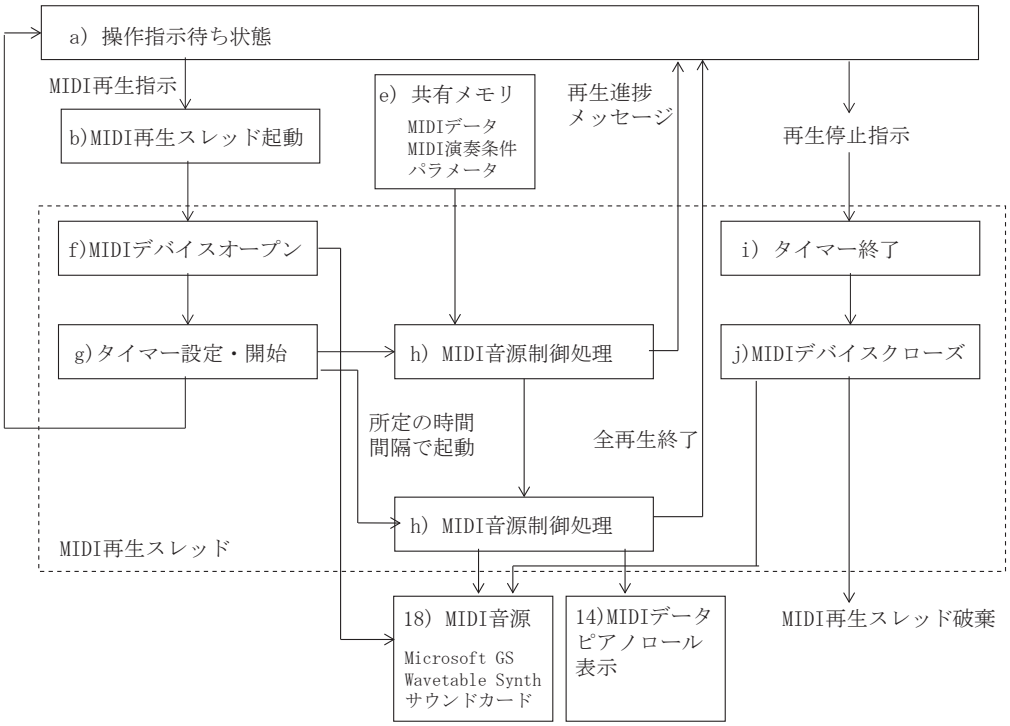

図６　MIDI 再生スレッドのソフトウェア詳細構成

図７にタイマーにより一定間隔に起動される「h）MIDI 音源制御処理」の詳細構成を示す。はじめに、タイマーを起動した時からの経過時間である「k）再生時刻の取得」を行う。「e）共有メモリ」に記録されている「l）MIDI データより次のイベントを取得」し、「m）イベントのタイムコードと再生時刻の比較」を行い、タイムコードが再生時刻の範囲内であれば、「18）MIDI 音源」に対して取得した「n）MIDI イベントの送出」を midiOutShortMsg（）関数を用いて行うとともに、図８に示すように、「o）MIDI イベントの描画」を行い、「14）MIDI データピアノロール表示」において１イベント分だけ描画更新を行い、「l）MIDI データより次のイベントを取得」に戻る。「m）イベントのタイムコードと再生時刻の比較」において、タイムコードが再生時刻を超えていれば、現在起動中の「h）MIDI 音源制御処理」は処理終了とする。また、「l）MIDI データより次のイベントを取得」において、ポインタが最終のイベントを指し、後続のイベントが存在しない場合は全再生終了とする。この時は、「a）操作指示待ち状態」に再生終了のメッセージを送り、図６の「c）MIDI 再生スレッド」を「i）タイマー終了」と「j）MIDI デバイスクローズ」に進ませ、「c）MIDI 再生スレッド」を破棄する。MIDI 再生終了後は、「a）操作指示待ち状態」より引き続き種々のメニュー指示を出すことができる。

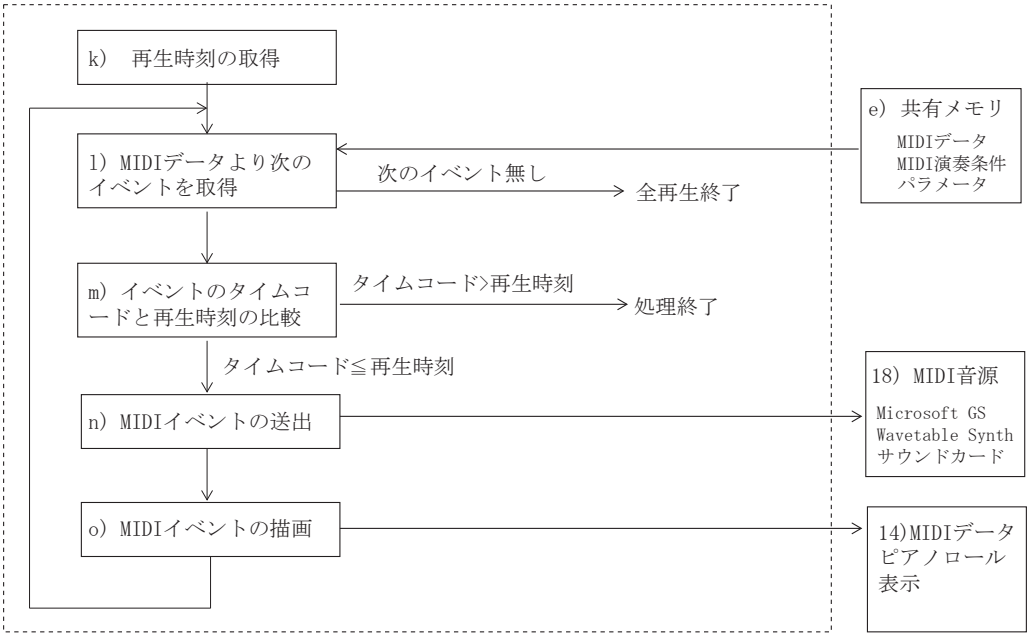

図７　MIDI 音源制御処理のソフトウェア詳細構成

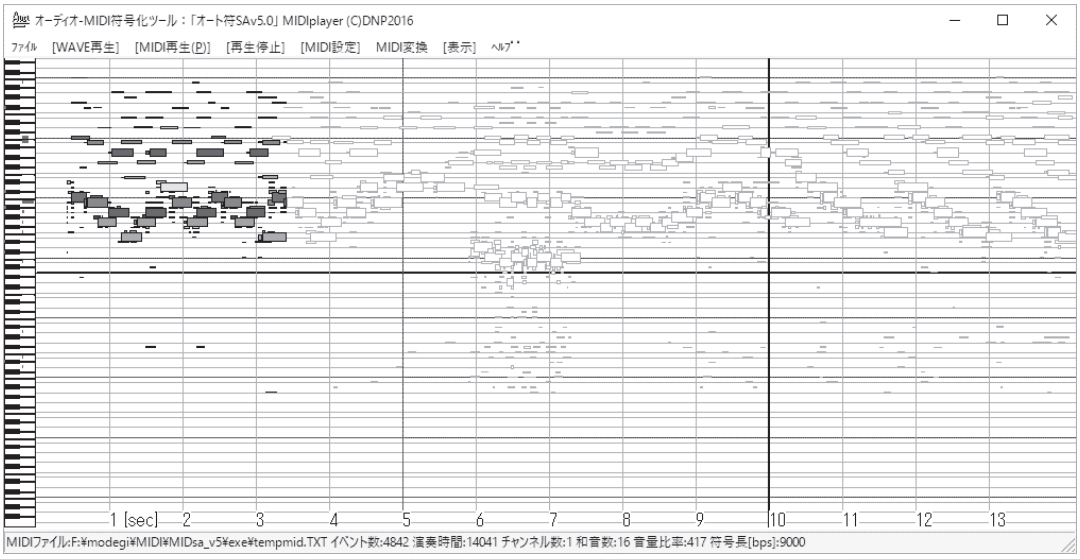

図８本ツールの MIDI サウンド再生時の GUI 画面

４．”Windows10 Anniversary Update” に伴う本ツールの MIDI サウンド再生系の不具合分析

本稿の主題である 2016 年夏に行われた ”Windows10 Anniversary Update” により生じた本ツールの不具合について説明する。この現象はアップデート前の Windows10 では発生しておらず、2016 年末に確認した段階では最新状態に OS アップデートをかけても、Windows8.1 以下の OS であれば 32/64bits を問わず本問題は発生していない。本ツールにおいて再現性のある現象としては、図５で MIDI 再生指示を出して「c）MIDI 再生スレッド」を起動し、再生終了後に「a）操作指示待ち状態」に戻った段階で、「d）設定ダイアログ起動」を実行してダイアログを閉じるとツールがダウンする。この時、いかなるダイアログを起動しても同様な問題が発生するが、メニュー操作だけで完結する処理では発生しない。例えば、MIDI 再生指示や再生停止指示を繰り返すだけであれば本問題は発生しない。また、図３で WAV 再生指示を出して「c）MCIWnd 再生スレッド」を起動した場合には、同様にダイアログの起動・終了の操作を行っても本問題は発生しない。以上のことから、MIDI サウンド再生系に問題があり、「c）MIDI 再生スレッド」を実行することにより、本ツールの GUI ダイアログを管理する「e）共有メモリ」上のリソースが上書き・破壊されていることが推察される。

図５の MIDI サウンド再生系において「e）共有メモリ」に対して破壊が行われる候補原因としては、「c）MIDI 再生スレッド」と「18）MIDI 音源」の２点が考えられる。

４．１．　不具合原因候補１：MIDI 再生スレッド

「c）MIDI 再生スレッド」では、前述の通り、Win32API における低水準の midiOutShortMsg（）関数を用いており、Windows ライブラリの ”Winmm.lib” を参照している。”Winmm.lib” の実体は、Windows システムの ”C:¥Windows¥SysWOW64¥winmm.dll” にあり、”Windows10 Anniversary Update” により更新されていることを確認している。そこで、低水準の midiOutShortMsg（）関数を用いず、図３と同様に再生中に「e）共有

メモリ」上の MIDI データを参照しないようにすると現象が改善するかを確認した。

図９は、図５における「c）MIDI 再生スレッド」に対して、図３の WAV サウンド再生系と同様に MCIWnd 関数群を用いて実装し直したものである。この方法で MIDI サウンドを再生するには、３．１節で述べた MCIWndOpen（）関数において、「1）WAV-Format サウンドファイル」の代わりに「3）SMF-format MIDI ファイル」を指示すれば良い。そのため、「b）MIDI 再生スレッド起動」の前に、「e）共有メモリ」に記録されている MIDI データを、記録されている MIDI 演奏条件パラメータに基づいて加工を加えた上で、「p）SMF ファイルへ保存」する処理を加える必要がある。このように変更した図９の構成を試した結果、図５と同様な問題が発生し、「c）MIDI 再生スレッド」において低水準の midiOutShortMsg（）関数を用いていることや再生中に「e）共有メモリ」を参照することが原因である可能性は低いと考えられる。ただし、スレッドを管理するリソースも「e）共有メモリ」に記録されているため、たとえ再生中に「e）共有メモリ」への参照を行わないようにしても、「c）MIDI 再生スレッド」が「e）共有メモリ」にアクセスすることがなくなる訳ではなく、「c）MIDI 再生スレッド」が原因になっていることは否定できない。

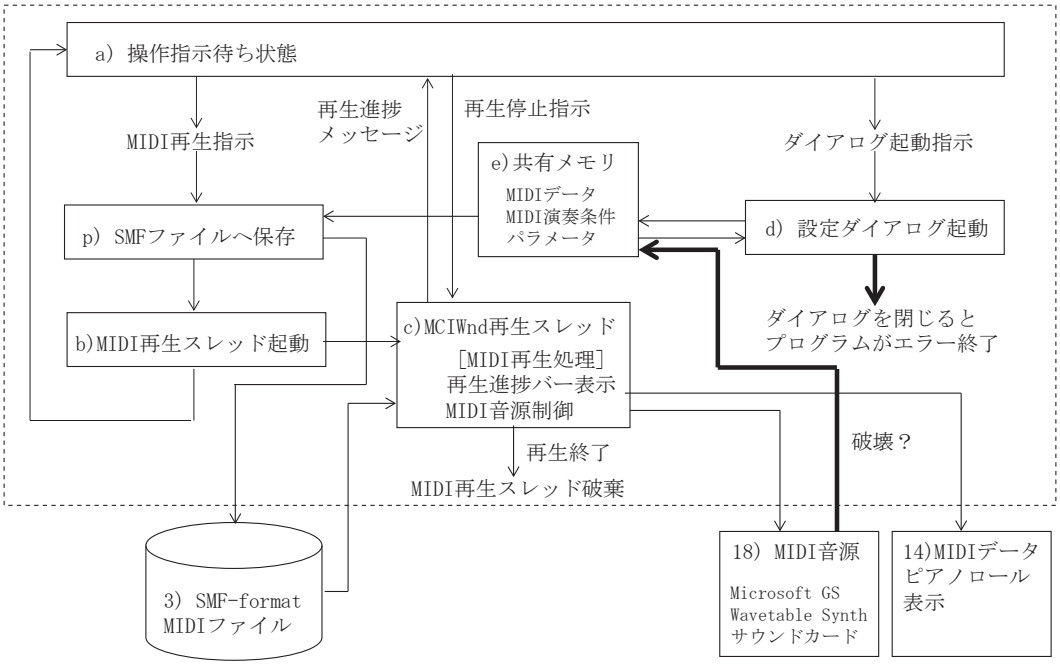

図９　WAV サウンド再生系と同様な MCIWnd を用いて実装した例

続いて、再生中に「e）共有メモリ」へのアクセスを一切遮断すると、現象が改善するかを確認した。そのためには、MIDI サウンドの再生を、本ツールとは切り離された外部プログラムで実行させるようにすれば良い。図１０は、図５における「c）MIDI 再生スレッド」の代わりに、「r）Windows Media Player」という外部プログラムを起動するように実装し直したものである。外部プログラムを起動するには、あらかじめ SMF 形式 MIDI ファイルの拡張子 ".mid" と「r）Windows Media Player」とファイル関連付けを WindowsOS 側で設定しておき、Win32API の ShellExecute（）関数で同ファイルを

Open させれば良い。そのため、図９と同様に、「q）再生プログラム起動」の前に、「e）共有メモリ」に記録さている MIDI データを、記録されている MIDI 演奏条件パラメータに基づいて加工を加えた上で、「p）SMF ファイルへ保存」する処理を加える必要がある。このように変更した図１０の構成を試した結果、図５の不具合は解消され、「c）MIDI 再生スレッド」の実行に伴う「e）共有メモリ」へのアクセスが主原因である可能性が濃厚になった。

４.２.　不具合原因候補２：MIDI 音源

本ツールで通常使用している「18）MIDI 音源」は、Windows OS に同梱されている "Microsoft GS Wavetable Synth" と称するソフトウェア MIDI 音源であり、その実体は、Windows システムの "C:¥Windows¥System32¥drivers¥gm.dls" にあり、"Windows10 Anniversary Update" により更新されていることを確認している。「c）MIDI 再生スレッド」実行中には「18）MIDI 音源」にアクセスするため、「18）MIDI 音源」が主原因である可能性も考えられ、本節ではこれについて分析する。

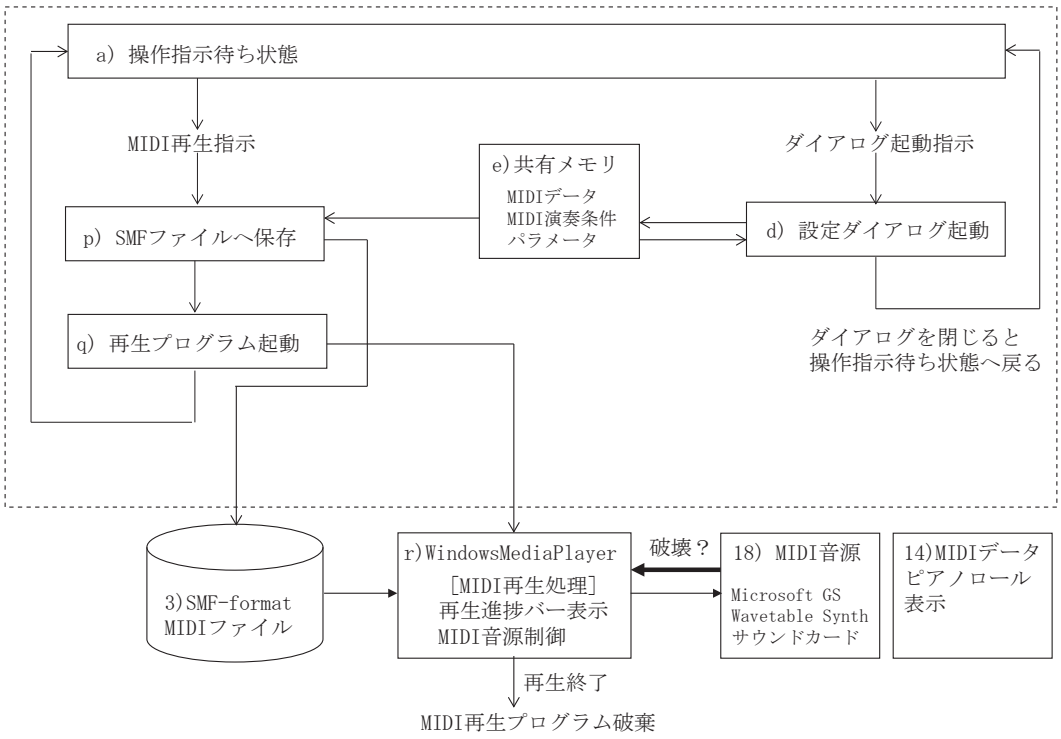

図１０　MIDI サウンド再生系を WindowsMediaPlayer を用いて実装した例

はじめに、前節で述べた、問題が解消しなかった図９の構成による実験結果から、同じ MCIWnd 関数群を用いて、WAV サウンド再生では問題が発生せず、MIDI サウンド再生では問題が発生している。両者の相違の１つに、「18）MIDI 音源」を使用しているか否かがあることから、「18）MIDI 音源」が主原因になっている可能性が考えられる。一方、問題が解消した図１０の構成で使用している "Windows Media Player" も、前述の "Microsoft GS Wavetable Synth" というソフトウェア MIDI 音源を参照している。"Windows Media

Player” の MIDI 再生では特に問題は生じていない。ただし、MIDI 再生後にダイアログを起動したりしていないため、本ツールと同様な問題が生じるかは不明である。従って、この結果からは、「18）MIDI 音源」の実行に伴う「e）共有メモリ」へのアクセスが原因であるかは不明である。

次に、国産の商用音楽ツールとして代表的な（株）インターネットの “Singer Song Writer Start”[14] を ”Windows10 Anniversary Update” を行ったパソコンにインストールし動作テストを行った。本ソフトには Roland 社の ”VSC-1” と称するソフトウェア MIDI 音源が同梱されており、始めにこれを用いてテストした。その結果、同梱されている MIDI 音源 ”VSC-1” では再生音にノイズが混ざる上に、再生中に音途切れが生じ、まともな MIDI 再生が行えなかった。ただし、ダイアログ等の起動によりツールがダウンするような問題は生じなかった。ところが、同ツールにおいて、MIDI 音源の設定を本ツールで使用しているものと同じ ”Microsoft GS Wavetable Synth” に切り替えて MIDI 再生を試みたところ、適切に MIDI 再生が行え、前述のような再生音品質の問題は生じなかった。この場合も、MIDI 再生後にダイアログ等を起動しても本ツールのような問題は生じなかった。従って、”Windows10 Anniversary Update” により、本ツールのようにツールがダウンする現象は見られないものの、使用する「18）MIDI 音源」により再生音品質の問題が生じる場合があり、「18）MIDI 音源」の実行に伴う「e）共有メモリ」へのアクセスが原因である可能性が高くなった。

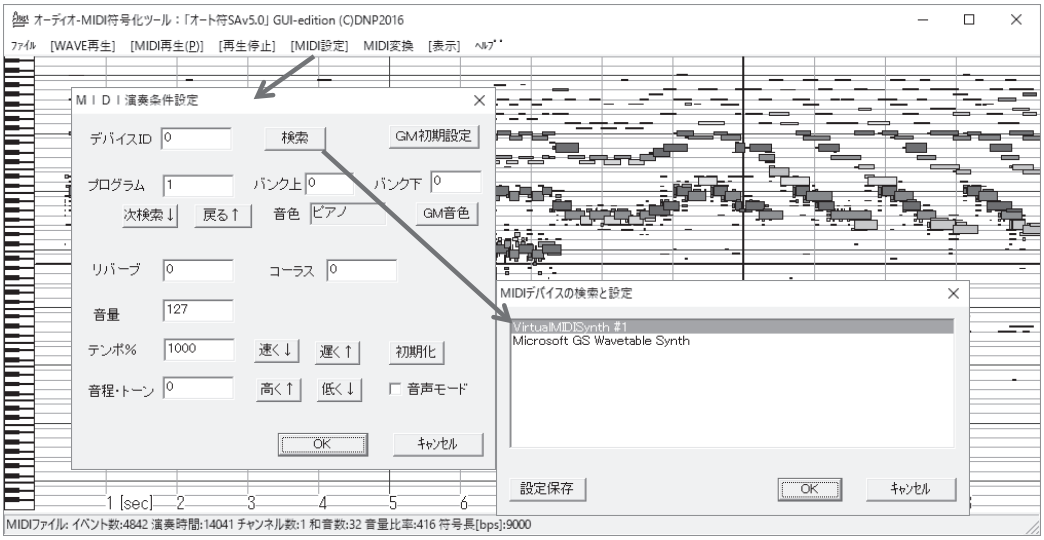

図１１　追加したソフトウェア MIDI 音源の本ツール側での設定例

最後に、本ツール側でアクセスする MIDI 音源を ”Microsoft GS Wavetable Synth” 以外に設定すると問題が改善するかを確認した。前述の MIDI 音源 ”VSC-1” は、“Singer Song Writer Start” からでないとアクセスできないため、”Windows10 Anniversary Update” を行ったパソコンに商用のソフトウェア MIDI 音源を追加インストールし、本ツールおよび前述の商用ツールを用いて動作テストを行った。ソフトウェア MIDI 音源のドライバーとしてはフリーウェア CoolSoft 社 “virtualMIDISynth”[15] を用いた。このソフトでは種々のサードパーティ提供のサウンドフォントをインストールすることができ、試し

にフリーのサウンドフォントである MediaFire 社 “SGM-V2.01”[16] をインストールして実験を行った。まず、前述の “Singer Song Writer Start” を用いて追加インストールされた MIDI 音源に設定して MIDI 再生を試みたところ、適切に MIDI 再生が行え、前述の ”VSC-1” のような再生音品質の問題は生じなかった。続いて、図１１に示すように、本ツール側で追加インストールされた MIDI 音源を設定し、図５の構成で動作テストを行ったところ、前述のツールがダウンする不具合は解消された。この状態で、MIDI 音源を ”Microsoft GS Wavetable Synth” に再度切り替えて MIDI 再生を試したところ、前述の不具合が再現された。即ち、本ツールにおいて「18）MIDI 音源」を ”Microsoft GS Wavetable Synth” 以外に変更すれば、不具合は解消することから、前述の “Singer Song Writer Start” とは不具合の性質が異なるものの、「18）MIDI 音源」が主原因である可能性が濃厚になった。以上のことから、主原因は「c）MIDI 再生スレッド」そのものではなく、「c）MIDI 再生スレッド」がアクセスしている「18）MIDI 音源」に関連するドライバーソフトウェアにあり、「18）MIDI 音源」の実行に伴い「e）共有メモリ」上のリソースが破壊され、再生品質に問題が生じたり、ツールがダウンするという問題が発生するという結論になった。

５．本ツールの MIDI サウンド再生系の不具合回避策とソフトウェア改修

前節により、不具合の原因は概ね明らかになり、本ツールの場合は「18）MIDI 音源」の設定を ”Microsoft GS Wavetable Synth” から “virtualMIDISynth” などサードパーティの MIDI 音源に変更すれば、本ツール自体に特に手を加えなくても不具合を回避できることが判明した。また、マイクロソフト社の MIDI 音源ドライバーが不具合の原因になっている可能性が高いため、次期 Windows　OS のアップグレードにより修正される可能性がある。しかし、当面はマイクロソフト社の MIDI 音源を使用せざるを得ないのと、“virtualMIDISynth” 以外のサードパーティの MIDI 音源でも ”Microsoft GS Wavetable Synth” と同様な不具合が発生する可能性はあるため、マイクロソフト社の MIDI 音源を用いても不具合を回避できるように本ツールに対して改修を行う方針にした。

ソフトウェア改修を進めるにあたりヒントとなるのが、４．１節で述べた図１０の「c）MIDI 再生スレッド」を外部プログラムで実装し直した構成である。前述の通り、図１０の ”Windows Media Player” を用いた構成にすれば、”Microsoft GS Wavetable Synth” の MIDI 音源を用いても不具合は発生しなかった。しかし、外部プログラムを起動するにあたり、「3）SMF-format MIDI ファイル」を事前に作成しておく必要があり、MIDI サウンド再生時に、「22）MIDI 演奏条件音源指定」により音色やテンポ等を変更できるという本ツールの特徴を実現しにくくなる。また、外部プログラムの ”Windows Media Player” は、本ツールのメイン側と切り離されているため、図５の構成のように、再生進捗メッセージがメイン側の「a）操作指示待ち状態」に届かず、「a）操作指示待ち状態」から外部プログラムに対して再生停止指示を行うことができない。即ち、再生進捗管理や再生停止指示は外部プログラム側単独で行うことになり、再生進捗は単純なバー表示になってしまい、「14）MIDI データピアノロール表示」を再生に同期させるという本ツールの設計思想が実現できなくなる。

そこで、外部プログラムとして ”Windows Media Player” の代わりに、本ツール自身を用いる方針にした。即ち、MIDI サウンド再生を行う時だけ、本ツールが二重に起動され

た状態になり、GUI 環境は変化しないため、従来通りの操作で図５と同様な機能を実現しながら不具合を解消できる。図１２は、図１０の構成で「q）再生プログラム起動」により起動される外部プログラムとして「r）Windows Media Player」の代わりに、「r）サブ側プログラム」に変更したものである。「r）サブ側プログラム」の構成は基本的に本ツールと同一であるが、「r）サブ側プログラム」から「e）共有メモリ」にはアクセスできないため、図１０の「3）SMF-format MIDI ファイル」と同様に、「e）共有メモリ」に記録されている MIDI データと MIDI 演奏条件パラメータを「p）一時ファイルへ保存」し、「5）一時ファイル」経由で渡す必要がある。ただし、図１０のような「3）SMF-format MIDI ファイル」では、本ツールで定義している MIDI 演奏条件パラメータを全て保存できないため、「5）一時ファイル」形式は独自に定義した。具体的には、独自の「4）TXT 形式 MIDI データ」に MIDI 演奏条件パラメータを付加したものである。MIDI 演奏条件パラメータは、図１１の左側のダイアログボックスで示される設定内容で、デバイス ID、プログラム（楽器音色コード）、バンク上下（拡張音色コード）、リバーブ、コーラス、音量、テンポ、音程・トーンからなる。

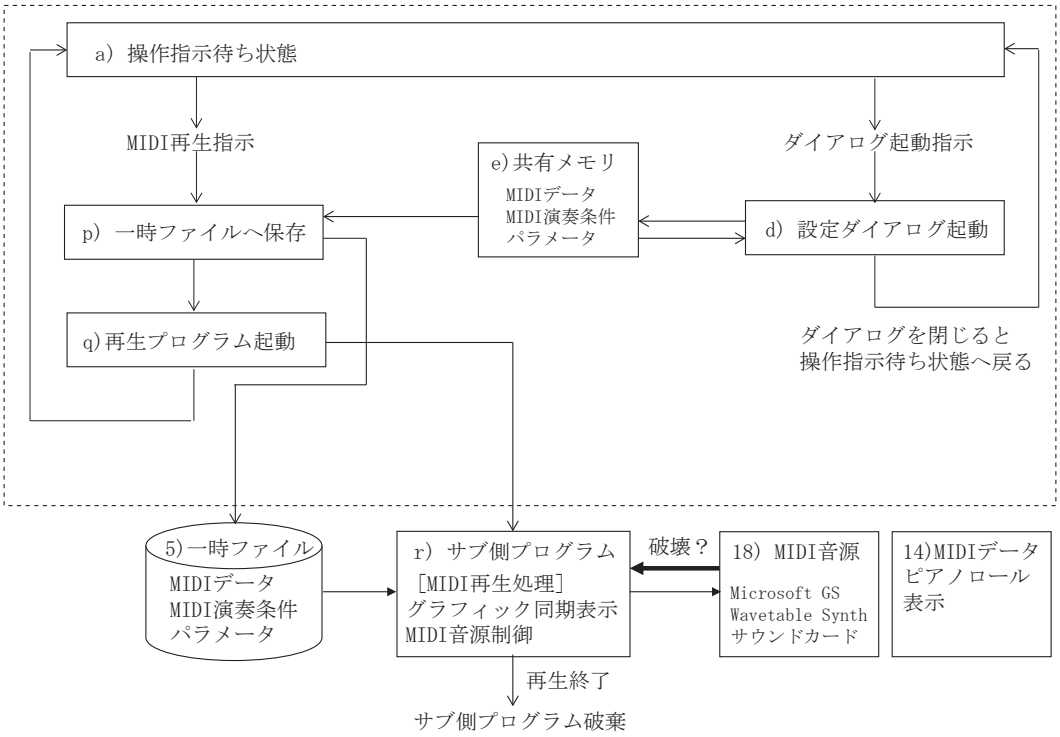

図１２　MIDI サウンド再生系のソフトウェア改修（メイン側プログラム）

その後、「q）再生プログラム起動」により「r）サブ側プログラム」が起動されるが、「q）再生プログラム起動」については、４．１節で述べたように、Win32API の ShellExecute（）関数で「r）サブ側プログラム」の実行形式ファイルを Open させれば良い。また、「q）再生プログラム起動」を実行する際、GetWindowRect（）関数を用いてメイン側のウィンドウサイズと位置を取得し、「r）サブ側プログラム」のウィンドウ生成時に MoveWindow（）関数を用いて指示するようにした。これにより、「r）サブ側プログラ

ム」のウィンドウがメイン側のウィンドウにピタリと重なるように起動させることができ、プログラムが二重に起動されてもGUI画面は変わらないためシームレスな操作が行える。図１３に「r）サブ側プログラム」起動時のGUI画面を示す。図１３では、サブ側プログラムをメイン側プログラムより故意にずらして示しているが、サブ側プログラム起動時には、メイン側プログラムのウィンドウが見えないように、メイン側プログラムの上に重なる。

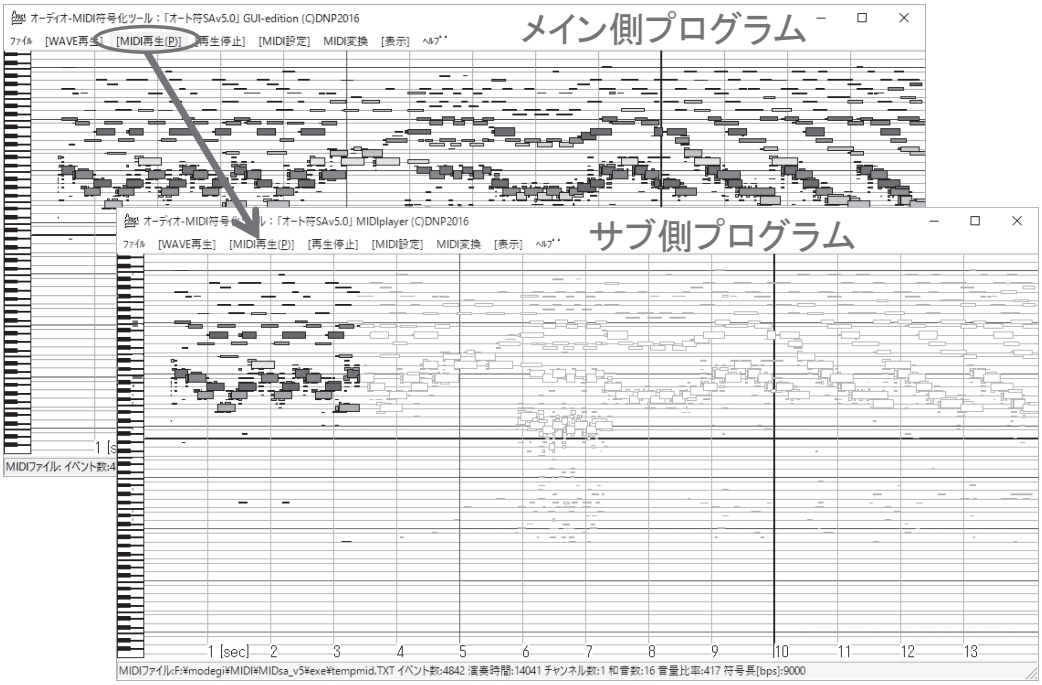

図１３　本ツール改修後のMIDIサウンド再生時のGUI画面

図１４は「r）サブ側プログラム」の構成を示し、基本的には図１２と同様な構成で、双方に「a）操作指示待ち状態」と「e）共有メモリ」が存在するが、プログラムが異なるため、これらは独立した機能である。「r）サブ側プログラム」が起動されると、SendMessage（）関数により「r）サブ側プログラム」の「a）操作指示待ち状態」にMIDI再生指示が自動的に送られ、「s）一時ファイル読込み」により前述の「5）一時ファイル」を読み込んで「e）共有メモリ」に記録する処理を行う。続いて、「b）MIDI再生スレッド起動」を実行し、以降は３．２節で述べた流れで「c）MIDI再生スレッド」によるMIDI再生処理が実行される。「r）サブ側プログラム」では図５と同様に再生停止指示などの対話操作が行えるが、再生終了となって「c）MIDI再生スレッド」が破棄されると、「r）サブ側プログラム」自体も処理を終了しウィンドウを破棄するようになっている点が図５の構成と異なる。「c）MIDI再生スレッド」により「r）サブ側プログラム」の「e）共有メモリ」の内容が破壊されている可能性があるため、「r）サブ側プログラム」は破棄してしまう。

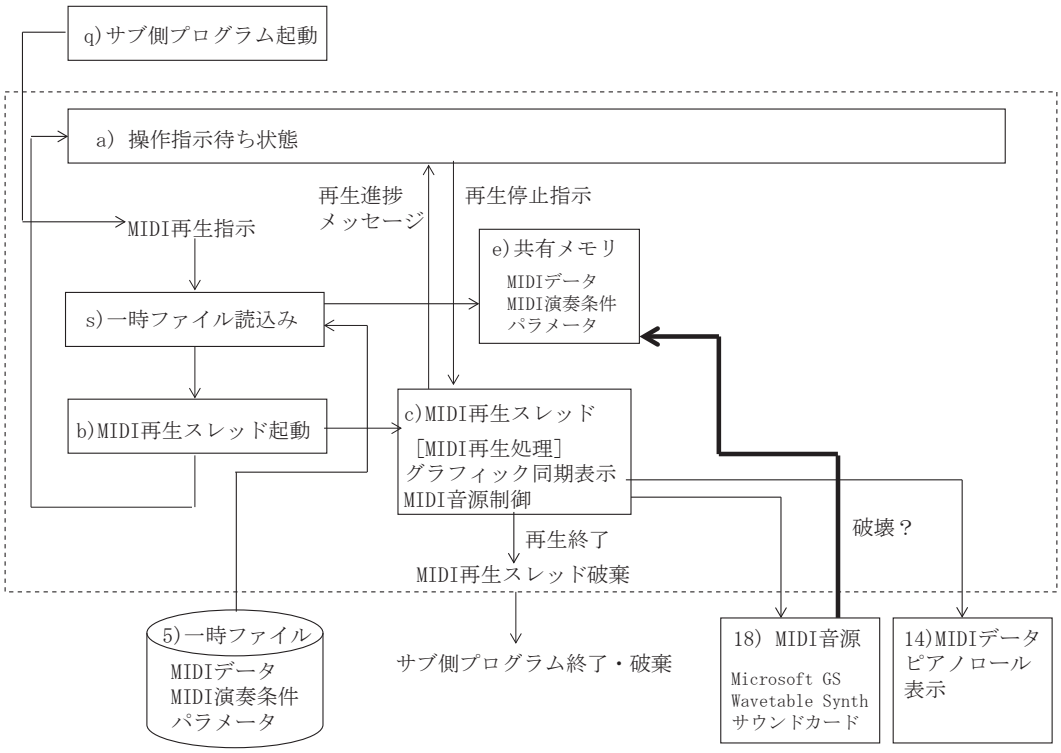

図１４　MIDI サウンド再生系のサブ側プログラムのソフトウェア処理構成

そうすると、「r）サブ側プログラム」のウィンドウの下側に隠れていた、図１２の本ツール・メイン側の「a）操作指示待ち状態」に制御が戻り、引き続き種々のメニュー指示を出すことができる。この時、図１４の「c）MIDI 再生スレッド」により「r）サブ側プログラム」の「e）共有メモリ」の内容が破壊されることがあっても、本ツール・メイン側の「e）共有メモリ」には影響が及ばないため、「d）設定ダイアログ起動」を実行してダイアログを閉じてもツールがダウンするような不具合は発生しない。以上のソフトウェア改修により、”Microsoft GS Wavetable Synth” の MIDI 音源を用いても前述の不具合は発生しなくなった。

６．おわりに

本稿では、音響信号の MIDI 符号化ツール「オート符」の全体構成、符号化処理構成とサウンド再生の構成の各概要を紹介し、”Windows10 Anniversary Update” により生じた MIDI サウンド再生系の不具合の分析結果と回避策について説明した。不具合の現象としては、「c）MIDI 再生スレッド」により MIDI 再生を実行すると、「e）共有メモリ」上のダイアログのリソースが破壊され、ダイアログを起動するとツールがダウンするというものであった。その主原因としては、「c）MIDI 再生スレッド」そのものではなく、「c）MIDI 再生スレッド」がアクセスしている「18）MIDI 音源」、特に WindowsOS 標準添付のソフトウェア MIDI 音源 ”Microsoft GS Wavetable Synth” が関連しているという結論になった。回避策としては、MIDI サウンド再生を行う時だけ、本ツールを再生専用にもう１本一時的に起動させ、MIDI 再生により再生専用側ツールの「e）共有メモリ」のリソースが破壊されても、本ツールのメイン側には影響が及ばないようにした。そして、本ソ

フトウェア改修により、前述の不具合を解消させることができた。

本稿で紹介したツールは、開発着手から丁度 20 年が経過し本年で基本特許 [3] も切れる。その間、音楽制作環境は様変わりしており、まずループと呼ばれる波形形式の音楽フレーズ部品を貼り合わせるだけで、BGM レベルの音楽制作が簡便に行えるループシーケンス・ソフト [17] が登場し、MIDI ベースで個々の音符を打ち込む業務を大幅に削減した。そのため、マルチメディア規格の中で最も長い 34 年の歴史をもつ MIDI 規格 [4] の存在意義を懸念する声も一部出てきたが、平均律音階や和声学を基調とした音楽制作の形態が無くならない限り不滅であると考える。また、VOCALOID[18] に代表される歌声合成ツールが登場し、DTM（Desk Top Music）だけで音楽制作をトータルに行える環境が整ってきた。本ツールの MIDI 音源を用いた歌声再生品質が VOCALOID と比較され、本ツールの意義を疑問視される声もあったが、本ツールは歌声も MIDI ベースで打ち込み可能にすることを追及している点で、今後も、VOCALOID と棲み分けが可能であると考えている。

以上、本稿で紹介したツールは、前述の通り過去約 16 年間にわたって筆者が担当した本学・情報表現学科の授業「マルチフィールド体験演習」（旧：基礎演習Ⅰ）の教材としても使用してきたもので、並行して改良開発も進めてきた。その間、WindowsOS は数々のアップグレードがなされたが、その影響を殆ど受けなかった。その意味で、2016 年夏に行われた Windows10 Anniversary Update により生じた問題は初めての体験で、本ツールのソースコードを見直す良い機会にもなり、何とか復旧させることができた。これまで長期に渡り本ツールを継続して活用し、種々の改良を加えることができたのも、本学の学生や教職員の皆様とのインタラクションによるものと考えており、改めて本学関係者の皆様に謝意を示す。また、本稿で紹介した改修版ソフトウェアツール「オート符 version 5」について、教育・研究・その他にご活用されたい場合は C 言語ソースコードを含め提供可能ですので、ご連絡ください（E-mail: Modegi-T@mail.dnp.co.jp）。

引用文献

1）茂出木敏雄:「聴覚芸術への情報学的アプローチと音楽情報処理ツールの開発事例」,『尚美学園大学・芸術情報研究』, 第 18 号 , pp.15-35, Nov. 2010.

2）茂出木敏雄:「オーディオ -MIDI 符号化ツール「オート符」における表情付け解析機能の実装」,『尚美学園大学・芸術情報研究』, 第 20 号 , pp.17-34, Nov. 2011.

3）特許公報 :「音声信号の符号化方法および音声の記録再生装置」, 特許第 3776196 号（出願 : 1997.3.5）.

4）一般社団法人・音楽電子事業協会 http://www.amei.or.jp/, accessed in Jan. 2017.

5）茂出木敏雄・飯作俊一 :「コンピュータミュージックによる医療診断の支援」,『コンピュータサイエンス誌ｂｉｔ』,Vol.29, No.11, 共立出版 , pp.14-23, Nov.1997.

6）Toshio Modegi, “Application of MIDI Technology for General Audio Signal Coding”, Information Systems and Technologies for Network Society, World Scientific Publishing Co. Pte. Ltd., pp.163-168, Oct. 1997.

7）Toshio Modegi and Shun’ichi Iisaku, “Proposals of MIDI coding and its application for audio authoring,” Proceedings of IEEE International Conference on Multimedia Computing and Systems, IEEE Press, pp.305-314, Jun. 1998.

8）Toshio Modegi, “Multi-track MIDI encoding algorithm based on GHA for synthesizing vocal sounds”, Journal of Acoustic Society of Japan, Vol.20, No.4, pp.319-324, Sep. 1999.
9）一般財団法人デジタルコンテンツ協会 , http://www.dcaj.or.jp/, accessed in Jan. 2017.
10）茂出木敏雄：「音響情報の MIDI 符号化ツール「オート符（R)」の開発」、『芸術科学会誌 DiVA』、No.2, 夏目書房（株）, pp.42-48, Dec. 2001.
11）公開特許公報：「音響信号の符号化装置」, 特開 2016-28269 （出願：2015.6.17）.
12）田辺義和：『Windows サウンドプログラミング - 音の知識×プログラミングの知識 -』、（株）翔泳社 , pp.151-256, Apr. 2001.
13）北山洋幸：『WAV プログラミング　C 言語で学ぶ音響処理』、（株）カットシステム , pp.230-300, Sep. 2008.
14）株式会社インターネット　“Singer Song Writer Start”: http://www.ssw.co.jp/products/ ssw/win/ sswstart/index.html, accessed in Jan. 2017.
15）CoolSoft　“virtualMIDISynth”: http://coolsoft.altervista.org/en/virtualmidisynth#soundfonts, accessed in Jan. 2017.
16）MediaFire, SoundFont “SGM-V2.01”: http://www.mediafire.com/file/zo8l3dgf2989266/ SGM-V2.01.7z, accessed in Jan. 2017.
17）MAGIX Software GmbH（Sony Creative Software）“ACID”, http://www. magix-audio.com /us/ acid/ , accessed in Jan. 2017.
18）ヤマハ株式会社 “VOCALOID”, http://jp.yamaha.com/products/music-production/software/ vocaloid/, accessed in Jan. 2017.