

MIDI 符号化ツール「オート符」を用いた音素 MIDI コードの設計と楽器音による音声合成機 能の実現

茂出木敏雄[†]

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」では、音声信号を与えると、標準的な MIDI 音源で近似的に音声を再現可能な MIDI データを生成できる。本ツールを応用して、日本語約 70 音節が単独に録音された音響信号を準備し、各々を MIDI コードに変換した音節 MIDI コードのデータベースを構築し、更に、各音節を子音部と母音部の MIDI コードに分離した 20 種の音素 MIDI コードのデータベースを構築した。本稿では、先ツールに対して時間分解能を改善する手法について提案し、カナテキスト入力により MIDI 楽器音を基本とした音声合成機能を実現するための音素 MIDI コードのデータベース設計ツールについての開発事例を報告する。

Design of Phoneme MIDI Codes Using the MIDI Encoding Tool “Auto-F” and Realizing Voice Synthesizing Functions Based on Musical Sounds

Toshio Modegi[†]

Using our previously developed audio to MIDI code converter tool “Auto-F”, from given vocal acoustic signals we can create MIDI data, which enable to playback the voice-like signals with a standard MIDI synthesizer. Applying this tool, we are constructing a MIDI database, which consists of previously converted simple structured MIDI codes for a set of Japanese syllable signals and also separated consonant and vowel phoneme signals. In this paper, we propose an improved MIDI converter tool which can produce temporally higher-resolution MIDI codes. And we propose a design tool of a set of phoneme MIDI-code database in order to realize a novel voice synthesizing system based on harmonically synthesizing musical sounds, which can generate MIDI data and playback voice signals with a MIDI synthesizer by giving Japanese plain (kana) texts.

1. はじめに

筆者らは与えられた音響信号に対して一般化調和解析を用いて平均律音階のスケールで高精度な周波数解析を行い、MIDI データ形式に自動変換する技術の開発を進めてきた[2][3]。本技術は「オート符@SA」という名称で汎用的な音響解析ツールとしてまとめ、2001 年より財団法人デジタルコンテンツ協会のホームページより無償配布を進めており、主として採譜業務の支援等に活用いただいている[5]。本解析ツールは、特に和音解析精度が高く、音声信号に適用すると解析されたフォルマント成分が MIDI 形式に和音近似され、一般的な MIDI 音源を用いてボーカルが再現できるという特徴をもつ。

これに対して、和音演奏可能な玩具楽器や、複数のアコースティック楽器等をアンサンブル演奏することにより、ボーカルが再現できないかという要望が多く寄せられるようになり検討を進めてきた。ボーカルを再現する音声合成技術としては、これまで文献[1]のようにヒトの音声に近づけることを目標に研究が推進されてきており、不自然さの面で課題は残されているものの、部分的に産業応用されるレベルに到達している。一方、特にエンターテインメント等の分野においては、セリフは聞き取れる必要性はあるが、ヒトとは全く異なる、例えば動物を模倣した声質などリアリティとは逆の声質で再現することも要望されている。そこで、筆者らは種々の楽器音を模倣した声質で音声合成を実現する手法についてトライアル研究を進めている。手始めに、71 種の日本語音節を録音した素材を用いて、デュレーションやベロシティを均一にした二連の単純な和音で表現した音節 MIDI データベースを構築し、カナテキスト入力により MIDI 音源や楽器演奏により音声を再現できる MIDI データを合成するシステムの試作を実施した[4]。しかし、聴取可能なセリフについてはかなり限定され、より再生音声の明瞭性が求められている。

本稿では、MIDI 音源を用いて、より明瞭な音声再現を実現できるよう、先に提案した音響解析ツールに対して周波数解析における時間分解能を改善する手法について提案する。続いて文献[4]で作成した日本語 71 音節に関する男声・女声 MIDI データベースに対して、各音節を更に音素データに分離し、日本語 20 種の音素に関する MIDI データベースを構築する方法について提案する。そして、各音素を単一の和音で表現し、与えられたカナテキストより、これら 20 種の和音の組み合わせで音声合成を実現できるようにし、楽器音による合成音声の聴取性について評価したので、その概要を紹介する。

2. 既提案の音響信号の MIDI 符号化ツールの概要と改良手法

[†] 大日本印刷株式会社 情報コミュニケーション研究開発センター
Media Technology Research Center, Dai Nippon Printing Co., Ltd. (E-mail: Modegi-T@mail.dnp.co.jp)

図1左側の縦方向のフロー(1)~(5)は、筆者らが先に開発した MIDI 符号化処理の主要構成を示し[3]、右側の(6)~(8)は本稿で追加提案する改良手法を示す。はじめに、与えられたソース音響信号より周波数解析対象のフレームを抽出するが、後続フレームへのシフト幅はソース音響信号の周波数変動を大まかに検出しながら適応的に設定するようにしている。続いて、一般化調和解析手法に基づき平均律音階の半音（ノートナンバー）単位に非線形な周波数次元で周波数解析を行うが、周波数が高くなるにつれ、半音間隔が粗くなるため、周波数ごとに半音間を微分音（副周波数）に分割して解析を行うようにしている。続いて、時間的に隣接する同一の主周波数の解析成分（単音成分）を連結し音符としてまとめ、MIDI イベント形式で符号化する。最後に標準的な MIDI 音源で再生可能な和音数とビットレートになるように、MIDI イベントデータを削減する。この際、各 MIDI イベントデータには、ペロシティとデュレーション（ノートオン時刻からノートオフ時刻までの期間）情報をもたせており、これらの積が低い MIDI イベントデータを削減対象とするようにしている。

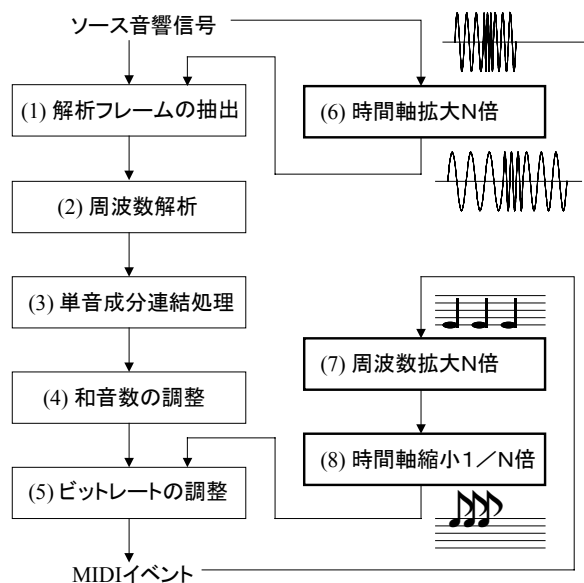


図1 既提案の MIDI 符号化処理の構成と追加改良構成

(1) 解析フレームの抽出

音響解析では解析フレームを時間軸方向に移動させながら、信号全体の解析を行うが、この際のフレーム長とシフト幅の設定方法について以下述べる。

周波数分解能はフレーム長により変化し、経験上ソース音響信号のサンプリング周波数が 44.1[kHz]の場合、低域部まで忠実に解析するためには 4096 サンプル以上必要である。解析時の解析フレーム長は解析周波数ごとに可変にするが、抽出するフレーム長は固定値として、例えば $T=4096$ を与える。

一方フレームシフト幅は、小さくするほど時間分解能が向上するが計算時間も増大する。そして、解析対象信号が単調である箇所に対して、必要以上にフレームシフトを細かくすると、後述する単音成分連結処理で支障をきたす。そこで、効率的な計算および高精度な単音成分連結処理のためにも、フレームシフト幅は解析対象フレームごとに変化させ、最適値を設定する方法が望ましい。

始めに、与えられた音響信号の全領域に対してゼロ交差点検出を行い、ゼロ交差点間隔の粗密または自己相関解析により周波数変化が顕著なゼロ交差点を選別する。基本的には、この変化が急峻なゼロ交差点に解析フレームの先頭を設定するが、例えば、最小シフト幅を $T/8$ に、最大シフト幅を $T/2$ に設定し、シフト幅が一定以上に細くなり過ぎたり、延び過ぎたりしないように調整を行う。

(2) 周波数解析

前節で述べた方法により、サンプリング周波数 f_s の原音響信号より p 番目に抽出された解析フレームのサンプル配列を $x(i)$ ($0 \leq i \leq T-1$) とする。周波数解析は、 n ($0 \leq n \leq 127$) を MIDI のノートナンバーとして 128 種の解析周波数 $f(n)=440 \cdot 2^{(n-69)/12}$ の調和関数を基本にした一般化調和解析により行う。しかし、周波数が高くなるにつれ、ノートナンバー間の周波数間隔が広がるため、特に $n > 60$ では解析精度が低下してしまう。そこで、ノートナンバー間を以下のように M 個の微分音に分割した $128M$ 種の調和関数を用いて解析を行う。

$$f(n,m)=440 \cdot 2^{(n-69+m/M)/12} \quad (1)$$

はじめに、ノートナンバー分の強度配列 $E(n)$ ($0 \leq n \leq 127$) と副周波数配列 $S(n)$ を定義し、初期値を全て 0 とする。

(a) $0 \leq n \leq 127$ のおよび $0 \leq m \leq M-1$ に対して以下式で相関計算を行い、 $E(n,m)$ を最大にする (n_{max}, m_{max}) を求める。式(2)において $T(n)$ は解析フレーム長で、フレーム長 T を超えない範囲で調和関数の周期の最大の整数倍になるよう設定し、 k を適当な整数値として、 $T(n)=k/f(n,m)$ で与える。

$$A(n,m)=1/T(n) \cdot \sum_{i=0, T(n)-1} x(i) \sin(2\pi f(n,m)i/f_s) \cdot$$

$$B(n,m)=1/T(n)\bullet\sum_{i=0,T(n)-1}x(i)\cos(2\pi f(n,m)if_s).$$

$$E(n,m)=A(n,m)^2+B(n,m)^2. \quad (2)$$

(b) 上記決定した $A(n_{max},m_{max})$ および $B(n_{max},m_{max})$ を用いて、以下式でサンプル配列 $x(i)$ の全ての要素 ($0\leq i\leq T-1$) を更新する。

$$x(i)=x(i)-A(n_{max},m_{max})\bullet\sin(2\pi f(n_{max},m_{max})if_s)-B(n_{max},m_{max})\bullet\cos(2\pi f(n_{max},m_{max})if_s). \quad (3)$$

(c) $E(n_{max})\leftarrow E(n_{max})+E(n_{max},m_{max})$ 、 $S(n_{max})=m_{max}$ として、再度(a)の処理に戻り、 $0\leq n\leq 127$ の全ての $E(n)$ および $S(n)$ の値が決定するまで(a)から(c)までの処理を繰り返す。

処理負荷を軽減するため、 M の値についてはノートナンバーに基づいて可変に設定し、例えば解析する周波数間隔が 100 [Hz] 程度になるようにする。そして、ノートナンバー 60 以下は分割せず $M=1$ にする。また、精度は若干落ちるが、初回(a)の処理で $S(n)$ を決定し、2 回目以降の(a)の処理は $m=S(n)$ に固定して行い、微分音解析を省略する方法もとれる。

また、上記(a)の段階で、既に同一ノートナンバーに対して副周波数が異なる信号成分が複数回に渡って解析される可能性があるが、 $E(n)$ と $S(n)$ に既に値がセットされている場合は $E(n,m)$ の最大値の選定候補から除外する方法もとれる。

(3) 単音成分連結処理

p 番目と $p+1$ 番目の解析フレームにより周波数解析されたノートナンバー n の単音成分を [時刻(p), 時間長(p), 主周波数 n , 副周波数 $S(n)$, 強度 $E(n)$] と [時刻($p+1$), 時間長($p+1$), 主周波数 n , 副周波数 $S(n)$, 強度 $E(n)$] とする。時刻(p)および時刻($p+1$)は各々解析フレームの第 1 サンプルの原音響信号上の絶対サンプルアドレスをサンプリング周波数で除算することで得られる。時間長(p)は時刻($p+1$)−時刻(p)で、時間長($p+1$)は時刻($p+2$)−時刻($p+1$)で与えられる。

時間的に隣接するこれら 2 つ単音成分に対して、副周波数を考慮した周波数の差が所定値 N_{dif} 未満で、双方の強度が所定のしきい値 L_{min} 以上でかつ双方の強度の差 L_{dif} が所定値以下で両者の連続性が認められる場合、後続単音成分を前方単音成分に連結統合する。ただし、連結後の主周波数・副周波数・強度は大きい方の単音成分の各値を採用し、時間長は双方の和、即ち時刻($p+2$)−時刻(p)で与える。

同一ノートナンバーにおける時系列の連結処理は、不連続が認められるまで後続する複数の単音成分に対して繰り返し行い、最終的に統合された [時刻(p), 時間長(p), 主周波数 n , 副周波数 $S(n)$, 強度 $E(n)$] に対して、2 つの MIDI ノートイベントに変換する。時刻(p) で、ノートナンバー n のノートオン・イベントを発行し、ベロシティ値

は $E(n)$ の最大値を E_{max} として、 $128\bullet\{E(n)/E_{max}\}^{1/4}$ で与える。時刻については、Standard MIDI File では、直前イベントとの相対時刻 (デルタタイム) で与える必要があり、その時刻単位は任意の整数値で定義でき、例えば、1/1536 [sec] の単位に変換して与える。そして、時刻(p) + 時間長(p) $\bullet C$ で、ノートナンバー n のノートオフ・イベントを発行する。 C は $0\leq C\leq 1$ の実数で、使用する MIDI 音源の音色に依存するが、MIDI 音源の余韻を考慮して早めにノートオフ指示をするためである。 $C=1$ にしても MIDI 音源の処理上は問題ないが、後続音と部分的に重なる場合がある。

上記連結条件のしきい値の具体的な設定値は、通常、 $N_{dif}=8/25$ [単位：ノートナンバー]、 $L_{min}=1$ [単位 128 段階ベロシティ]、 $L_{dif}=10$ [単位 128 段階ベロシティ] である。

(4) 和音数の調整

MIDI 符号に変換する段階で、MIDI 音源で処理可能な同時発音数についても考慮する必要がある。時間軸方向に発音期間中 (ノートオン状態) のノートイベントの個数を連続的にカウントし、例えば 32 和音を超えている箇所が見つかった場合は、各々対になるノートオフイベントを近傍区間内で探索し、各ノートイベント対のベロシティ値とデュレーション値 (ノートオフ時刻−ノートオン時刻) の積 (エネルギー値) で優先度を評価し、指定和音数以下になるよう優先度の低いノートイベント対を局所的に削除する処理を行う。この際、ベロシティ値またはデュレーション値のいずれかが所定の下限值より低い場合、優先度に関係無く削除する処理も加える。

(5) ビットレートの調整

MIDI 符号に変換する段階で、MIDI 音源で処理可能なビットレートについても考慮する必要がある。時間軸方向に例えば 1 秒間隔にノートオンまたはノートオフイベントの個数をカウントし、各々の符号長を平均 5 バイト (40bits) とし MIDI 音源で処理可能な最大ビットレートを 9000 [bps] とすると、1 秒間あたりイベント数が 9000/40=225 個を超えている区間が見つかった場合は、その区間に存在するノートオンまたはノートオフイベントと各々対になるノートオフまたはノートオンイベントを近傍区間内で探索し、各ノートイベント対のベロシティ値とデュレーション値 (ノートオフ時刻−ノートオン時刻) の積 (エネルギー値) で優先度を評価し、指定イベント個数 (225) になるよう優先度の低いノートイベント対を局所的に削除する処理を行う。この際、ベロシティ値またはデュレーション値のいずれかが所定の下限值より低い場合、優先度に関係無く削除する処理も加える。

(6) 時間軸拡大 N 倍

上記(1)を実行する前に、与えられた音響波形信号に対して、時間軸方向に N 倍 (例えば、 $N=4$) だけ線形補間により拡大する処理を行う。サンプル i に対して原音信号を

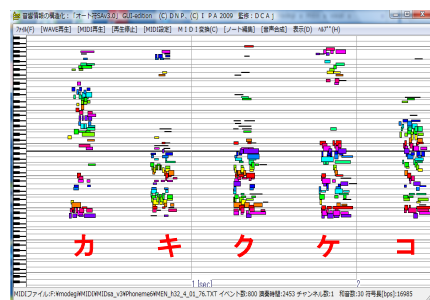
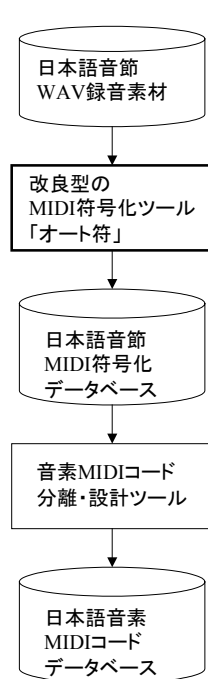
$x(i)$ 、拡大される信号 $x'(i)$ とすると、

$$x'(i \cdot N + k) = (1 - k) \cdot x(i) + k \cdot x(i + 1). \quad (0 \leq k \leq N - 1) \quad (4)$$

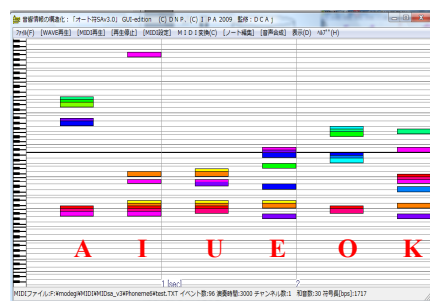
(7) 周波数拡大 N 倍

上記(6)を実行した場合、上記(5)を終了後に本処理を加える。全てのノートイベント(ノートオンまたはノートオフイベント)のノートナンバー値に $12 \cdot \log_2 N$ だけ加算する。例えば、 $N=4$ の場合 24 半音 (2 オクターブ) だけ全体的に音高を上げる。

(8) 時間軸縮小 1/N 倍



(A) オーディオの MIDI 符号化ツール「オート符」



(B) 音素 MIDI コード設計ツール

図2 日本語音素 MIDI 符号データベースの構築方法

上記(6)を実行した場合、上記(7)を終了後に本処理を加える。全てのノートイベントの時刻(ノートオン時刻またはノートオフ時刻)に $1/N$ を乗算して縮小する。この処理を行うと、時間あたりのノートイベント数が N 倍に増大するため、上記(5)のビットレートの調整を再度実行する。

3. 提案する日本語音素 MIDI コード設計ツールの機能

前節で述べた方法により、音声を録音した音響信号を与え、時間分解能 $1/1536[\text{sec}]$ 、和音数 32 の標準設定で MIDI 符号化を実行すれば、MIDI 音源で音声を再生可能な MIDI データを得ることができる。例えば、GM 規格の MIDI 音源でプログラム番号(54, Voice-Ooh)を設定して再生すれば、より音声らしい再生が可能である。これに対して、既存の音声合成ツールと同様にテキスト入力で音声再生用の MIDI データを出力する方法について以下述べる。

文献[4]で提案した日本語音節の和音モデルに基づき音素 MIDI コードのデータベースを設計する手順を図2に記す。まず、濁音、半濁音、撥音を含む全 71 音節の男声、女声の録音音声素材を収集する。

続いて、既開発のオーディオの MIDI 符号化ツール「オート符@SA」を用い、高精細な MIDI コードに自動変換する。変換結果例として、同図(A)に「カキケコ」の音節サンプルに対して適用した結果を示す。横軸は時間で縦軸は周波数で、各四角形が音符(ノートオンとノートオフ・イベントの対)を示し、横幅はデュレーション、縦幅はベロシティの表示も兼用している。

まず、音節ごとに 32 重音以下の各種デュレーションおよびベロシティ情報をもつ和音に変換し、その結果が同図(A)である。続いて、各々を2つの音素成分に分離し、5つの母音音素「A, I, U, E, O」と共通する1つの子音音素「K」にまとめ、各音素に対してデュレーションおよびベロシティが均一で8重音以下の単一の和音になるよう整形化を行った結果が同図(B)である。本例では全ての音素を単一の8和音に均一化し、ベロシティは全て127にする。これらの音素データを組み合わせて音節を構成する場合、音節のデュレーションと後続音素との間隔を各々標準区間長として、例えば $0.25[\text{sec}]$ とすると、子音音節の場合、第1和音は $1/3$ 区間長、第2和音は $2/3$ 区間長に設定する。

表1に MIDI データに自動変換された71種の日本語音節成分どうしを掛け合わせて、20種の音素成分に変換する方法を示す。まず、母音音素「A」・・・「O」は表1の第1列目に単独で存在するが、精度を向上させるため、同行の子音12音節を含めた13音節成分どうしのAND演算で変換する方法をとるようにした。一方、表1の子音音素「K」・・・「W」は同列の5音節(YとWは $3 \cdot 2$ 音節)に共通して含まれるた

表1 日本語71音節より20音素を分離するための変換テーブル

		K	S	T	N	H	M	R	G	Z	D	B	P	Y	W
A	ア	カ-A	サ	タ	ナ	ハ	マ	ラ	ガ	ザ	ダ	バ	パ	ヤ	ワ
I	イ	キ-I	シ	チ	ニ	ヒ	ミ	リ	ギ	ジ	ヂ	ビ	ピ		
U	ウ	ク-U	ス	ツ	ヌ	フ	ム	ル	グ	ズ	ヅ	ブ	プ	ユ	
E	エ	ケ-E	セ	テ	ネ	ヘ	メ	レ	ゲ	ゼ	デ	ベ	ペ		
O	オ	コ-O	ソ	ト	ノ	ホ	モ	ロ	ゴ	ゾ	ド	ボ	ポ	ヨ	ヲ
n			ン												

め、各々5音節成分どうしのAND演算で変換する。その際、各々に含まれている母音音素成分をあらかじめ削除した上でAND演算を行う。そのため、母音音素成分は先に決定しておく必要がある。撥音の母音音素成分「n」については「ン」音節単独で変換し、子音音素成分「N」と合成して撥音音節を合成するものとする。

具体的なAND演算を行う方法は次の通りである。複数のMIDIデータに自動変換された音節データより、表1に基づいて、共通の音素を含む要素を抽出し、各音節区間で発音されているノートイベントのベロシティ値とデュレーション値との積をエネルギー値とし、ノートナンバーごとにエネルギー値の総和を求める。子音音素を決定する場合は、各々含まれている母音音素に対してあらかじめ決定された母音音素のノートナンバーに対応する成分に所定の小さな値を乗算して減衰させておき、母音音素と重複するノートナンバーが抽出されないようにする。続いて複数の音節データどうしで対応するノートナンバーのエネルギー総和値を乗算する。そして、得られたエネルギー総和値の乗算値が高い順にノートナンバーを指定和音数(例. 8)だけ選択し、指定のベロシティ値(例. 127)とデュレーション値(例. 0.25sec)を与えた指定個数のノートイベントに自動変換する。

4. おわりに

前節で述べた方法を用いて、濁音、半濁音、撥音を含む全71音節の男声、女声の2セットの録音データに対して、時間分解能1/768sec、32和音構成で改良型「オート符」ツールを適用し図2(A)で示されるようなMIDIデータに変換した。続いて、全71音節の男声、女声の2セットの32和音構成のMIDIデータを用いて、全20音素の各々8和音構成で、図2(B)で示されるような音素MIDIコードに変換し、男声と女声の2セットのデータベースを作成した。その結果を表2・3に示す。図中の各音名表記はMIDIノートナンバー60をC3とし、全て8和音で構成され、強さ・長さは均一であるが、子音音節を合成する際は、子音音素である第1和音は短めに、母音音素である第2和音は長めに演奏することを想定している。

男声および女声の「こんにちば」に対して、表2・3の音素MIDIコードのデータベースを参照しながら再生可能なMIDIデータを合成し、市販の楽譜編集ツールを用いて五線譜に自動変換した結果を図3に示す。同図に示されるように、比較的判読性のある五線譜に変換できることが確認でき、文献[4]の提案手法に比べ、個々の音節の明瞭性は向上し、GM規格のMIDI音源でプログラム番号(1, GrandPiano)の設定でも音声らしきメッセージは聴取でき、音色をプログラム番号(54, Voice-Ooh)に設定して再生すれば、より音声らしい再生が可能であることは確認できた。

ただし、現段階では単音音節を聴取するのは困難で、短い音節の既知の単語を聴取できるレベルである。今後は、音声の聴取性を評価する手法を検討し、外国語音声への対応を含めて、音素MIDIコードのデータベースの再構築を進める予定である。

参考文献

- 1) 古井貞照:「電子・情報工学入門シリーズ2 音響・音声工学」, 近代科学社, 第13章, pp.161-173 (March 1996).
- 2) Toshio Moegei: "Very low bit-rate audio coding technique using MIDI representation," Proceedings of the ACM 11th international workshop on Network and operating systems support for digital audio and video, pp.167-176, New York, USA, (June 2001).
- 3) 茂出木敏雄, "音響信号の平均律音階に基づく汎用解析ツール「オート符」の開発," 電気学会・電子情報システム部門誌, Vol.123-C, No.10, pp.1768-1775, (October 2003).
- 4) 茂出木敏雄, "MIDI符号化ツール「オート符」を用いた音素MIDIコードの設計と楽器音による音声合成機能の実現," 電気学会・2009年電子情報システム部門大会・論文集, GS9-6, pp.1274-1281, (September 2009).

5) 財団法人デジタルコンテンツ協会 d-CON Support, <http://www.dcaj.org/d-con/frame09.html>
 (「オート符®SA」の無償配布元)。

表2 音素 MIDI コードのデータベース作成事例 (男声)

[A]	[I]	[U]	[E]	[O]	[K]	[S]	[T]	[N]	[H]
F5	A#6	D3	A#3	F#4	F4	B6	D#4	C4	D4
E5	C#3	C#3	A3	F4	A#3	A6	F3	D#3	A3
D#5	A#2	A#2	G#3	E4	C3	C4	D#3	G#2	F3
A4	D2	A2	E3	G#3	B2	B3	B2	G2	E3
G#4	C#2	D2	G#2	G3	A#2	F3	G#2	F#2	D#3
C2	C2	C#2	D2	F#3	G2	F2	F#2	F2	F#2
B1	B1	C2	C#2	C2	C#2	E2	E2	E2	E2
A#1	A#1	B1	A1	B1	A1	D#2	A1	A1	A1

[M]	[R]	[G]	[Z]	[D]	[B]	[P]	[Y]	[W]	[n]
B3	B3	C#4	C#4	D4	C4	C4	C#4	E4	D3
D#3	D#3	B3	C4	F3	B3	A#3	B3	D4	B2
C3	C3	F3	C3	C3	F3	A3	A3	C#4	A2
G2	B2	C3	B2	G#2	D#3	F3	F3	B3	G#2
F#2	G#2	B2	G2	G2	C3	G2	C3	G#3	D2
F2	G2	G2	F#2	F2	B2	E2	G#2	G3	C#2
E2	F2	F#2	E2	E2	G2	D#2	D#2	C2	C2
D#2	A1	E2	D#2	A1	E2	A1	A1	A#1	B1

表3 音素 MIDI コードのデータベース作成事例 (女声)

[A]	[I]	[U]	[E]	[O]	[K]	[S]	[T]	[N]	[H]
B5	A6	E5	G6	B4	D#5	G#6	G#6	C3	F4
A#5	G#6	D#5	D#6	A#4	B4	E4	D6	B2	E4
F#5	G6	D#4	B4	A4	D#4	D#4	G#5	A#2	D#4
D#5	F#6	C#4	F4	G#4	C4	B3	F4	A2	B3
C#5	G3	D#3	E4	D4	A3	E3	E4	G#2	A#3
C5	F3	D3	D4	C#4	D#3	D3	D#4	F2	E3
B4	E3	C#3	C#4	A#3	D3	C#3	E3	C2	D#3
G#4	D#3	A#2	C4	A3	C#3	B2	G#2	A#1	B2

表3・続き 音素 MIDI コードのデータベース作成事例 (女声)

[M]	[R]	[G]	[Z]	[D]	[B]	[P]	[Y]	[W]	[n]
F#4	G#6	G6	A6	G#6	G4	E5	D#6	D#5	B5
C3	D#4	F#6	G#6	D#4	D#4	E4	F#4	D5	A#5
B2	C4	E4	G#5	C4	B3	D#4	E4	B4	C4
A#2	B3	D4	D3	B3	B2	D4	C4	A#4	A#3
G#2	C3	C4	C3	D3	A2	B3	B3	E4	D#3
F#2	B2	B3	A2	C3	G#2	A#3	E3	D#4	D3
F2	A2	D#3	G#2	G2	G2	E3	A2	D4	C3
C#2	G#2	B2	C2	C2	B1	D#3	G#2	C4	B2

男声 こ ん に ち わ

女声 こ ん に ち わ

図3 日本語「こんにちは」MIDI 合成音の五線譜表現事例