聴覚機能をもつ音楽ロボットのためのアーキテクチャの設計とビートカウントロボットへの適用

○水本武志† 武田龍† 吉井和佳‡ 高橋徹† 駒谷和範† 尾形哲也† 奥乃博†
† 京都大学大学院情報学研究科 ‡ 産業技術総合研究所

Design of the Architecture for Musical Robots and Its Application for Beat Counting Robot

* Takeshi Mizumoto†, Ryu Takeda†, Kazuyoshi Yoshii‡,
Toru Takahashi†, Kazunori Komatani†, Tetsuya Ogata†, Hiroshi G. Okuno†
† Graduate Schoold of Informatics, Kyoto University
‡ National Institute of Advanced Industrial Science and Technology

Abstract— In this paper, we designed the general architecture for music robots and applied it to a beat counting robot. Conventional studies have two problems: robots were developed independently and self-generated sounds (e.g., motor noises and a singing voice) were assumed to be ignorable. Hence, (1) a common architecture is necessary to build music robots systematically and (2) auditory functions are essential to suppress self generated sound. The resulting robot recognizes, predicts musical beats with Real-time Beat Tracking System, and suppresses its own counting voice using an adaptive filter based on Independent Component Analysis. Experimental results shows that suppressing robot's own voice improves the accuracy of beat prediction.

Key Words: Robot Audition, Music Robot, Architecture, Self-generated sound

1. はじめに

人間の感情は音楽と密接に関連しているので, 音楽を通した両者とのインタラクションは, 両者の関係を豊かにすると期待される. 本稿では, 音楽と関連する機能を持つロボット全般を音楽ロボットと定義する. 音楽ロボットは, 従来の音声対話に加えて, 新しいロボットと人間の通信チャネルになると考えられる. TOYOTA の Partner Robot や小菅らの MS-DanceR [1] などは音楽ロボットの制御について工夫されているが, 聴覚機能については触れられていない. しかし, 聴覚機能は音楽を通したロボットと人間のインタラクションを実現する為に重要である. なぜなら, ロボットも人間と同じ音楽を聞いた上で行動しなければ, 音楽を媒介としたインタラクションが行われているとは言えないからである. Solis ら [2] はロボット自身の演奏している楽器音を聞き, フィードバックすることで正確な音高の演奏をしている. このロボットの聴覚機能の対象はロボットが生成した演奏音であり, 他者の演奏した音楽ではなかった.

我々は, ロボットが音楽を理解する能力は音楽認識能力と音楽表現能力から構成されると定義する. この 2 つの能力はバランスをとる必要がある. 音楽認識能力が低ければインタラクションを行えず, 音楽表現能力が低ければ, ロボットを観察している人間はロボットの能力を低く見積もってしまうからである. 本定義により, ロボットの音楽理解能力は観察を通したチューリングテスト [3] で測定すると言えよう.

演奏された音楽を聞くという観点において重要な課題となるのは, 自己生成音である. 本稿扱う自己生成音とはロボット自身が出した音全般である. モータ音はモータの改良によって小さくできる一方, 楽器演奏音や歌声などは, 観客に届く必要があるので, ロボットが聞く自己生成音は大きくなり, 本来聞くべき音楽音響信号の S/N 比は低下する. 従って, 音楽ロボットでは自己生成音への対処は不可避である. 琴坂ら [4] や小嶋ら [5] は, 太鼓や音楽を聞くロボットの報告を行っているが, マイクを別室に置くなどの方法で, 自己生成音の問題を回避していた.

我々は, 音楽ロボットにおける 3 つの課題

1. 音楽認識能力
2. 音楽表現能力
3. 自己生成音の抑圧

を統一的に扱うアーキテクチャを設計し, その適用例としてビートカウントロボットを開発した. ビートカウントロボットとは, ロボットが聞いた音楽のビートを予測し, 四分音符の位置にあわせて "1, 2, 3, 4, 1, 2, ..." と, 数えるロボットである. ビートの認識には, リアルタイムビートトラッキング [6] を用い, 表現には, 事前に録音した音声を適切なタイミングで発声する. また, STRAIGHT [7] を用いて録音した波形の時間長を伸縮させ, 音声のスピードで音楽のテンポも表現する. 実験で自己生成音の効果を評価した結果, 自己生成音の抑圧によってビート予測精度が改善することを確認した.

2. 音楽ロボットのアーキテクチャ

我々は, 1 章で挙げた 3 つの問題を統一的に扱うために, 音楽ロボット一般に適用可能なアーキテクチャを

設計した．設計にあたっては，Levelt の提案したモデル “A Blueprint for the Speaker” [8] を基礎にした．これは，人間が発話を意図してから実際に発話する一連の過程をモデル化したものである．モデルに人間が自分の声を聞くという要素が入っており，我々が扱う問題と類似しているので，基礎として採用した．本モデルは 3 つの要素，Conceptualizer，Formulator，Speech Comprehension から構成される．まず，Conceptualizer が発話内容のプランニングを行う．次に，Formulator が辞書を用いて言語に変換し，さらに唇の動きを生成する．そして，Speech Comprehension が自己発話を認識し，自らの意図と発話をモニタリングする．Levelt らは自己発話はすべて認識するとしていたが，内藤らの報告によると，自分自身の声を聞くとき，脳は高次の言語理解を行う聴覚連合野で処理せず，一次聴覚野のみが処理している [9]．従って，自分の声は (1) 低次の信号処理と，(2) 高次の意味の理解の 2 段階で処理していると考えられる．

我々が設計した音楽ロボットのアーキテクチャを図 1 に示す．本アーキテクチャは，2 つの要素から成る．

1. 音楽認識モジュール
2. 音楽表現モジュール

音楽表現モジュールの処理　最初に Conceptualizer で表現に関する知識 (例 歌詞，振り付け) を元に，ロボットが行うべき表現のプランニングを行う．次に，Formulator で音楽的知識に基づいて動作系列を生成する．ここで，音楽的知識を聞こえてきた音楽と不協和音にならないような音程や，タイミングとずれない動作を生成するための制約条件として用いる．最後に，動作系列を実際の物理信号 (例 モータへの信号，スピーカへの波形) に変換してロボットの身体へ送る．同時に，音楽表現の内部表現を音楽認識モジュールへも送信する．

音楽認識モジュールの処理　ロボットは自分の耳で自己生成音と音楽の混合音を聞く．次に，音楽表現の内部表現を手がかりとして，混合音を音楽と自己生成音に分離する．分離された音楽は音楽認識器へ送り，音楽の情報 (例 ビート，リズム，基本周波数) を抽出し，Conceptualizer へ送る．同様に，分離された自己生成音も Conceptualizer へ送る．なぜなら，自己生成音はロボットの内部状態を知る手がかりとなるからである．

このように，音楽表現モジュールと音楽認識モジュールが相互に通信することで，聴覚機能を持った音楽ロボットを実現する．

3. ビート カウント ロボット

3·1 アーキテクチャの適用

本節では，音楽ロボットの一般的なアーキテクチャをビートカウントロボットに適用する (図 2)．そのための変更点は次の 5 点である．

1. 音楽表現には声を使用．
   自己生成音をロボットの発声のみと仮定することで，自己生成音の影響を限定．具体的には，“表現に関する知識”(図 1) を “発声波形の集合”(図 2) に，“ロボットの身体”(図 1) を “発声器官 (スピーカ)”(図 2) に，“自己生成音”(図 1) を “自発声”(図 2) に置換．

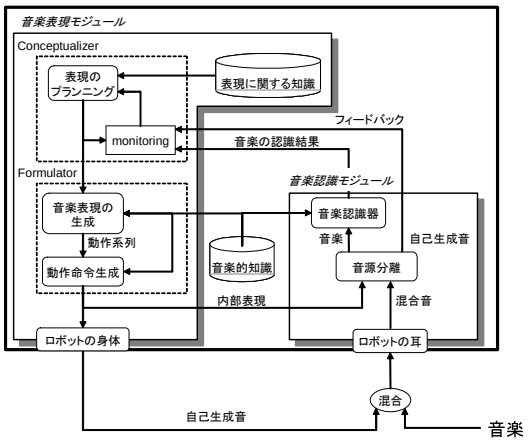

Fig.1 音楽ロボットの一般的なアーキテクチャ

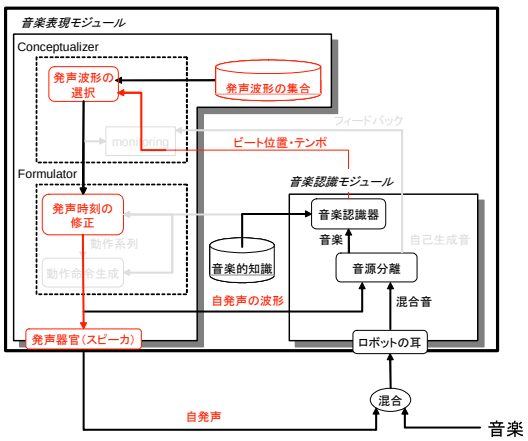

Fig.2 ビートカウントロボットへの適用例

2. 適切な発声を選択．
   ロボットの表現は，事前に音声波形を用意し，認識結果に応じて選択．また，発声波形の特性に応じた発声時刻も修正．具体的には，“表現のプランニング”(図 1) を “発声波形の選択”(図 2) に，“音楽表現の生成”(図 1) を “発生時刻の修正”(図 2) に置換．“動作命令生成”(図 1) は除去．
3. 自己生成音の波形は既知と仮定．
   前項より，仮定は成立するので，”内部表現”(図 1) を “自発声の波形”(図 2) に置換．
4. 分離された音楽のみを使用．
   本ロボットの自己生成音はロボット自身の声であるので，低次の信号処理を行うのが妥当であり，自己生成音をモニタリングするためのフィードバックは不要．具体的には，“フィードバック”(図 1) を除去．
5. 音楽の認識対象はビートの位置とテンポ．
   音楽の基本的な要素がビートであるので，ビートが認識対象．具体的には，“音楽の認識結果”(図 1) を “ビート位置・テンポ”(図 2) に置換．

3·2 ビートトラッキングによる認識

音楽の認識には，後藤らのリアルタイムビートトラッキング [6] を用いた．後藤らの手法は，音楽音響信号を入力とし，4 拍子を仮定することで，音響信号のテンポや含まれる楽器の仮定を置かずに音響信号のビート構造を出力する．ビート構造とは，音響信号の四分音符の

位置と, その四分音符の小節内の位置で構成される情報である.

リアルタイムビートトラッキングは次の 2 つのステージから成る: (1) 周波数分析ステージと (2) ビート予測ステージ. 周波数分析ステージで音響信号のオンセット時刻をパワースペクトルから検出し, ビート予測ステージで異なる予測戦略を持つマルチエージェントシステムによって次のビート時刻を予測する.

3·2.1 周波数分析ステージ

まず, 音響信号の短時間フーリエ変換 (STFT) を行う. 窓関数は 4096[points] のハニング窓を用い, 512[points] のシフト幅とした. サンプリングレートは 44.1[kHz] である. 次に, STFT された音響信号のパワーからオンセット成分を抽出する. オンセット成分は, 式 1 で定義され, パワーの立ち上がりの速度によって決まる値である.

$$d(t,\omega) = \begin{cases} \max(p(t,\omega), p(t+1,\omega)) - PrevPow, \\ \quad \text{if } \min(p(t,\omega), p(t+1,\omega)) > PrevPow, \\ 0, \text{ otherwise,} \end{cases}$$

$$\text{where } PrevPow = \max(p(t-1,\omega), p(t-1,\omega \pm 1)).$$

ただし, $d(t,\omega)$ はオンセット成分, $p(t,\omega)$ はパワースペクトルである. オンセット成分 $d(t,\omega)$ を 7 つの周波数帯 (0-125[Hz], 125-250[Hz], 250-500[Hz], 500-1000[Hz], 1-2[kHz], 2-4[kHz], 4-11[kHz]) に分割し, それぞれの総和を求める. こうして 7 次元のオンセット成分の信頼度が得られる. その各次元のピークをオンセットとし, その時刻の信頼度をオンセットの信頼度とする.

3·2.2 ビート予測ステージ

前節で得られたオンセット時刻とその信頼度をマルチエージェントシステムに与える. 各エージェントは, 異なる戦略で入力を解釈し, それぞれが次のビート予測を出力する. その予測をエージェントの確信度を元に統合し, システムの出力とする. 戦略のパラメータは次の 3 つである.

1. 注目周波数
   注目する周波数帯域を重みで指定.
2. 自己相関区間
   ビート間隔を求めるための, オンセットの自己相関関数を求める窓幅を指定.
3. ピークの選択
   半拍の予測誤りを防ぐために使用.

エージェントは, 次の 2 つの情報も用いる.

1. コード変化度
   エージェントが予測しているビート間隔で $p(t,\omega)$ を分割し, その変化度をコードの変化度とする.
2. ドラムパターン
   典型的なドラムパターンを事前に用意しておく. 音響信号のドラムのオンセットを検出し, パターンとマッチングすることで現在のドラムパターンを推定する. なお, ドラムの有無の判定も行う.

各エージェントの予測結果と実際のオンセットの信頼度からエージェントの確信度を更新する. 最も確信度の高いエージェントの予測をシステムの予測とする.

3·3 自発声の抑圧

3·3.1 問題の所在

自己生成音の波形が既知という仮定は, エコーキャンセルと類似した問題設定である. この問題に対しては, ICA に基づく適応フィルタ [10] を用いる. 典型的な手法として用いられている NLMS (Normalized Least Mean Square) はノイズに対してロバストではなく, しかも本稿の問題設定ではノイズは音楽であるので, 常にノイズが存在することになり, NLMS を用いることはできない. 一方, ICA に基づく適応フィルタは学習則に非線形関数を含んでおり, ノイズのパワーが大きくてもフィルタ係数が発散しにくいので採用した.

3·3.2 ICA に基づく適応フィルタ

ICA に基づく適応フィルタは, 1ch の未知信号と既知信号の混合音から既知信号を分離する手法である [10]. 以下に概要を述べる.

未知信号を窓幅 T, シフト幅 T で STFT し, f フレーム目, ω のスペクトルを $S(f,\omega)$ で表す. このとき, 未知信号は空間の残響によって後続の M フレーム目まで干渉する. すなわち, 後続への干渉 $S(\omega, f-1), S(\omega, f-2), ..., S(\omega, f-M)$ を仮想的な別音源とみなす.

分離フィルタの学習則は次の通りである.

$$\mathbf{w}(\omega, f{+}1) = \mathbf{w}(\omega, f) + \mu_1 \phi_{\hat{N}(\omega)}\left(\hat{N}(\omega, f)\right) \bar{\mathbf{S}}(\omega, f) \quad (1)$$

ここで, μ_1 は学習係数, $\bar{\mathbf{S}}$ は $\mathbf{S}$ の複素共役, $\hat{N}(\omega, f)$ は推定されたノイズのスペクトログラム, $\mathbf{w}$ は分離フィルタベクトルである.

オンライン学習アルゴリズムは次の通りである.

$$\hat{N}(f) = Y(f) - \mathbf{S}(f)^T \mathbf{w}(f), \quad (2)$$

$$\hat{N}_n(f) = \alpha(f)\hat{N}(f), \quad (3)$$

$$\mathbf{w}(f{+}1) = \mathbf{w}(f) + \mu_1 \phi_{N_n}(\hat{N}_n(f))\bar{\mathbf{S}}_{\mathbf{n}}(f), \quad (4)$$

$$\alpha(f{+}1) = \alpha(f) + \mu_2[1 - \phi_{N_n}(\hat{N}_n(f))\bar{\hat{N}}_n(f)]\alpha(f) \quad (5)$$

ここで, $\alpha(f)$ は正の正規化係数, 非線形関数は $\phi(x) = \tanh(|x|)e^{j\theta(x)}$ とする.

3·4 発声による表現

3·4.1 発声内容

ビートカウントロボットでは, ビートトラックによって得られたビート構造を “1, 2, 3, 4” と発声することで表現する. 発声波形は事前に 16[kHz] でサンプリングした. また, STRAIGHT [7] を用いてサンプリングした波形の時間長を 2 倍と 1/2 倍に伸縮させ, 合計 12 種類の音声波形を用意した.

発声内容はビート予測結果に応じて変化させ, 時間長はビート間隔を閾値で判定することで変化させた.

3·4.2 発声タイミング

ロボットが発声するタイミングは, 基本的にはビート予測結果に従う. ただし, 発声内容によってその特性, 例えばアクセントの位置が異なるので, 発声内容に応じた発声タイミングの修正が必要である. そこで, ビートトラッキングと同じ方法で発声波形のオンセットを求め, 信頼度が閾値 θ を超える最初の時刻を発声のビー

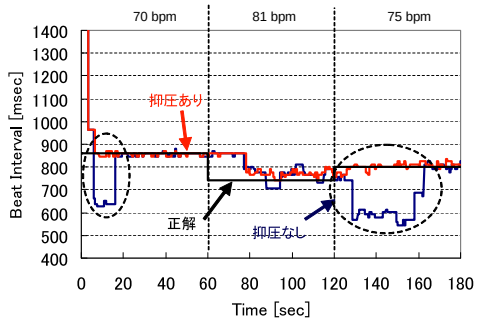

Fig.3 条件 1: 周期的なカウント

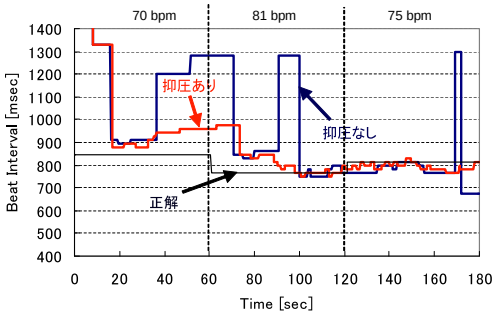

Fig.4 条件 2: 非周期的なカウント

トと定義した. 本稿では, $\theta = 0.5$ とした. この発声のビート時刻と予測されたビート時刻が合うように発声時刻を修正した.

4. 評価実験

自己生成音の抑圧の有無によるビートカウントロボットの認識能力の変化を評価する実験を行った.

4·1 実験条件

ロボットには, Robovie-R2 を用い, 1 チャネルのマイクロフォンを装着した. 実験に用いた音楽音響信号には RWC 音楽データベース [11] からテンポの異なる 3 曲 (RWC-MDB-P-2001-52, 94, 56) を各 1 分ずつ切り取った, 合計 3 分の曲を用いた. 曲のテンポはそれぞれ 70, 81, 75[bpm] である. マイクと音楽を出力するスピーカは 140[cm] の距離があり, マイクロフォンとロボット自身の声を出力するスピーカは 40[cm] の距離がある.

次の 2 条件で実験を行い, それぞれ自己生成音を抑圧の有無による認識能力の有無を比較した.

1. 周期的なカウント: 認識結果に従う
2. 非周期的なカウント: 認識結果を乱数に置換

4·2 実験結果

条件 1 の実験結果を図 3 に示す. 縦軸が予測されたビート間隔, 横軸が時間である. 波線の時点で曲が変わっている. 自己生成音の抑圧を行わない場合, 1 曲目の開始時点と 3 曲目の開始時点でビート予測が失敗している. すなわち, 自己生成音によって曲の変化への追従性能が低下している. それに対して, 自己生成音を抑圧した場合は追従性能が改善している. 2 曲目の, 正解への追従が遅れている部分は, ビートトラックのエージェントの確信度が変化するまでの時間である.

条件 2 の実験結果を図 4 に示す. この場合も, 抑圧によって正解との誤差が改善されているが, 条件 1 の場合よりも収束が遅い. これは, 抑圧しきれなかった非周期的な発声の成分がビートトラックに影響を及ぼしているからであると考えられる. 消し残り成分は条件 1 の場合にも存在するが, その影響は小さかった. これは, 条件 1 の場合はその成分が周期的であるためにビート予測への干渉が限定的であったからだと考えられる.

5. 結論

人間とインタラクション可能な音楽ロボットにはその場の音楽や自己生成音を聞き分けることが必須である. そこで, 我々は聴覚機能を持つ音楽ロボットの一般的なアーキテクチャを設計し, その適用例としてビートカウントロボットを開発した. 実験の結果, 自己生成音の抑圧による認識能力の低下の改善を確認した.

ただし, 本稿では自己生成音が波形が既知と仮定した. この仮定は, ダンスロボットにおけるモータ音や楽器演奏ロボットにおける演奏音などには必ずしも適用できない. 今後, そのようなロボットへの適用と, その場合の自己生成音の抑圧に取り組む予定である.

参考文献

[1] K. Kosuge, T. Hayashi, Y. Hirata, and R. Tobimyama. Dance partner root - ms dancer. In *IROS 2003*, pp. 3459–3464, October 2003.

[2] J. Solis, K. Taniguchi, T. Ninomiya, T. Yamamoto, and A. Takahashi. Development of waseda flutist robot wf-4riv: Implementation of auditory feedback system. In *ICRA 2008*, pp. 3654–3659, May 2008.

[3] A. Turing. Computing machinery and intelligence. *Mind*, Vol. LIX, No. 235, pp. 433–460, October 1950.

[4] S. Kotosaka and S. Shaal. Synchronized robot durumming by neural oscillator. *Journal of RSJ*, Vol. 19, No. 1, pp. 116–123, 2001.

[5] H. Kozima and M. P. Michalowski. Rhythmic synchrony for attractive human-robot interaction. In *Proc. of Entertainment Computing*, October 2007.

[6] M. Goto. An audio-based real-time beat tracking system for music with or without drum-sounds. *Journal of New Music Research*, Vol. 30, No. 2, pp. 159–171, June 2001.

[7] H. Kawahara. STRAIGHT, exploration of the other aspect of vocoder: Perceptually ismorphic decomposition of speech sounds. *Acoust. Sci. and Tech.*, Vol. 27, No. 6, pp. 349–353, 2006.

[8] Willem J. M. Levelt. *Speaking: From Intention to Articulation*. ACL-MIT Press Series in Natural Language Processing. The MIT Press, March 1989.

[9] 本庄巌. 言葉を聞く脳しゃべる脳. 中山書店, May 2000.

[10] R. Takeda, K. Nakadai, K. Komatani, T. Ogata, and H. G. Okuno. Exploiting known sound source signals to improve ica-based robot audition in speech separation and recognition. In *IROS 2007*, pp. 1757–1762, November 2007.

[11] 後藤真孝, 橋口博樹, 西村拓一, 岡隆一. RWC 研究用音楽データベース: ポピュラー音楽データベースと著作権切れ音楽データベース. 情報処理学会 音楽情報科学研究会 研究報告, 第 2001 巻, pp. 35–42, October 2001.