学術・技術論文

ロボットを対象としたビートトラッキング法の提案とその音楽ロボットへの応用

村田和真[*1] 中臺一博[*1*2] 武田龍[*3]
奥乃博[*3] 長谷川雄二[*2] 辻野広司[*2]

Musical Beat-Tracking for Robots and Its Application to A Music Robot

Kazumasa Murata[*1], Kazuhiro Nakadai[*1*2], Ryu Takeda[*3],
Hiroshi G. Okuno[*3], Yuji Hasegawa[*2] *and* Hiroshi Tsujino[*2]

Human-robot interaction through music in real environments is essential for robots, because such a robot makes people enjoyable. To deal with real music signals by using robot's own ears, we propose a beat-tracking algorithm for a robot based on semi-blind independent component analysis (SB-ICA) and spectro-temporal pattern matching (STPM). SB-ICA suppresses a self-generating sound such as singing or scatting which heavily affects beat-tracking due to its periodicity. STPM provides quick adaptation to beat changes because it is able to use a shorter matching window than conventional beat-tracking methods based on self-correlation functions. We thus developed a music robot which steps, sings, and scats according to musical beats based on the proposed beat-tracking method. The experimental results using the music robot showed highly noise-robust beat-tracking even when the robot was singing or scatting, and quick adaptation to beat changes like a human clapping sound whose tempo is always changing.

Key Words: Beat Tracking, Music Robot, Human-robot Interaction, Noise Robustness, Robot Audition

1. はじめに

近年，ヒューマノイドやホームロボットなど人とソーシャルインタラクションを行うロボットの研究が盛んに行われている．実際に，ロボットが日常環境でソーシャルインタラクションを行うためには，自らの耳で音を聴き，それに応じて行動する能力が必要である．音声コミュニケーション能力はいうまでもなく，人が演奏した音楽を聴き，その音楽に応じて歌ったり踊ったりする音楽インタラクション能力は，自然で豊かなインタラクションを実現する上で重要である．本稿では，音楽インタラクションにおける最も基本的な情報の一つであるビート（拍）に着目し，ロボットへの適用で重要となる高速追従性と安定性を兼ね備えたSTPMリアルタイムビートトラッキング（spectro-temporal pattern matching based real-time beat tracking）法を提案する．また，提案手法で抽出したビート情報を用いて，歌唱や口ずさみといった声の生成と足踏みなどの動作を同期させ人とインタラクションを行う「音楽ロボット」の構築についても併せて述べる．

2. ロボットを対象としたビートトラッキングの課題

ロボットに適用可能なビートトラッキングを実現するためには，次の二つの課題が挙げられる．

(1) ロボット搭載マイクの利用と雑音抑制
(2) ビートトラッキングの安定性・適応性の向上

以下の節で，これらの課題について，詳しく述べる．

2.1 ロボット搭載マイクの利用と雑音抑制

人・ロボットインタラクションのメディアとして，音楽への関心は高く，音楽に合わせて歌や踊りなどの挙動が生成できるロボットが複数報告されている．例えば，MIDI信号に合わせてダンスを行うWABIAN†，複数体で同期して，歌い踊ることができるSony QRIO††，人と社交ダンスを踊るMSDanceR [8] [15] [16]，モーションキャプチャデータを利用して，踊りの模倣を行うHRP-2 [12] が挙げられる．しかし，これらのロボットは，実際には音楽を聴いているわけではなく，事前にプログラムしたとおりの動作を行っている．このため，人とのインタラクティブ

原稿受付 2008年12月20日
*1東京工業大学大学院 情報理工学研究科
*2(株) ホンダ・リサーチ・インスティチュート・ジャパン
*3京都大学大学院 情報学研究科
*1Graduate School of Information Science and Engineering, Tokyo Institute of Technology
*2Honda Research Institute Japan Co., Ltd.
*3Graduate School of Informatics, Kyoto University
■ 本論文は提案性で評価されました．

†http://www.takanishi.mech.waseda.ac.jp/research/
††http://www.sony.net/SonyInfo/QRIO/

なセッションを行うような状況に不向きである．ロボット搭載マイクを用いて音楽情報を抽出するロボットの報告も少数だが存在し，ドラムを介して人とセッションを行うロボット [9] や，音楽ビートに合わせて素早く愛らしい動作を行う Keepon [10] が挙げられる．しかし，これらのロボットは入力の音楽音響信号に雑音が含まれていない，もしくは，雑音が無視できる程度に入力音楽音響信号が大きいことを前提としている．一般的な環境では，雑音の混入は不可避であるため，環境雑音が大きい場合や，自己発声音が生じる場合などは問題が生じる．特に自己発声音の原因となる音源は対象音源よりもロボット搭載マイクに近い位置にあるため，その影響は大きい．こうした問題は，主に「ロボット聴覚」研究 [11] で扱われているが，その多くは音声を対象にしており，対象音源として音楽を扱ったものは筆者らの知る限り見当たらない．しかし，音楽ロボットでは，歌声などの自己発声音が周期性を有しており，これがビートトラッキングに影響を与えてしまう．こうした音楽ロボット特有の問題を解決するためには，高精度な自己発声音の抑制が必要である．

2.2 ビートトラッキングの安定性・適応性の向上

人の演奏のテンポは常に一定ではないこと，曲の途中でテンポが変わりうることから，ロボットを対象としたビートトラッキングには，テンポ変化への高い追従性能が求められる．一方で，MIDI 信号などテンポが比較的一定である場合は，安定してテンポが抽出できることが望ましい．安定的にテンポを抽出するためには，ビートトラッキングのテンポ推定で用いる時間窓を長くすればよいが，逆にテンポ変化への追従性は悪化してしまう．一般にテンポ変化への追従性と安定性はトレードオフ関係にあり，この両立は大きな課題である．音楽情報処理分野では，ビートトラッキング研究が盛んに行われてきたが，安定性を重視，もしくは複雑なビート構造の推定に研究の視点が置かれてきたため，テンポ変化を考慮していない報告が多い [4] [14]．テンポ変化への対応を考慮した手法としては，古くは，上述のトレードオフを制御するため Decay パラメータを導入した手法が挙げられる [2]．この手法は，譜面データに対してその有効性を示しているが，状況に応じて，最適な Decay パラメータを設定する必要があり，このままロボットに適用することは難しい．また，近年では，テンポの揺れや変化に対応するため，確率的手法を用いた報告も見られる [1] [5] [7]．しかし，確率的手法は，曲やジャンルに応じた最適な確率分布を事前に与える必要があり，ロボットへの適用では大きな制約である．

2.3 課題解決のアプローチ

本稿では，上記の課題に対し，セミブラインド独立成分分析（Semi-blind Independent Component Analysis，SB-ICA），および時間周波数パターンマッチング（Spectro-Temporal Pattern Maching，STPM）を導入した自己発声音抑制付き STPM リアルタイムビートトラッキング法（以降，単に STPM リアルタイムビートトラッキングと呼ぶ）を提案し，その解決を試みる．さらに，提案した STPM リアルタイムビートトラッキング手法の有効性と有用性を実証するため，Honda ASIMO に提案手法を適用し，音楽ロボットを構築した．構築した音楽ロボットは，ロボット自身に備えられたマイクを用いて抽出したビートに合わせて，足踏みを行い，同時に口ずさんだり，歌声を出力したりすることが可能である．

2.3.1 SB-ICA による自己発声音抑制

SB-ICA は，ロボット発話中のユーザ割り込み発話（バージイン）に対応するため，我々が開発した手法である [18]．この手法は一般的な適応フィルタベースの手法と比較して，非ガウス性雑音にも有効であり，雑音区間検出器も不要である点で優れている．また，発声音を既知音源として入力に用いるため，雑音除去性能が高い．そこで，高精度な雑音除去性能が必要な歌声や口ずさみなどの自己発声音抑制に本手法を適用する．本稿では，音楽ロボット特有の問題にフォーカスするため，環境雑音やロボット動作音といったロボット聴覚分野で研究が行われている問題は扱わないが，マイクロホンアレイを導入することによって解決可能であると考える．SB-ICA は，マルチチャンネル入力にも簡単に拡張できるため，マイクロホンアレイを導入した場合にも本手法は有効である．

2.3.2 STPM によるテンポ推定

一般的なビートトラッキングでは，時間領域，もしくは各周波数ラインの一次元的な自己相関からビート間隔（テンポ）を推定する．この際に安定性を重視して，自己相関関数の窓長を長くとることが多いが，そのぶん，追従性が損なわれる．STPM は，画像処理で用いられる手法を利用して，この制約を緩和する．具体的には，パワースペクトログラム上で Sobel フィルタを適用して，オンセット強調と定常雑音抑制を行った後，正規化相互相関によるパターンマッチングを行う．ここで，パターンマッチング前に正規化を施すことにより，雑音が平均化され定常雑音がさらに抑制される．また，マッチングは時間周波数領域で二次元的に行う．これらにより，雑音に対するテンポ推定のロバスト性を保ったまま，テンポ変化への高速な追従性を確保することができる．代表的な実時間ビートトラッカー [4] では，テンポ推定に 6〜10 秒程度の窓長を持った自己相関関数を用いているのに対し，STPM では，1 秒程度の窓長を実現し，ビートトラッキングにおけるテンポ推定の安定性と追従性のトレードオフを緩和することができている．また，窓長は同時に検出可能なテンポの最小値を規定する．我々が使用したロボットの足踏みのテンポは，ハードウェア的な制約から，60〜120 [M.M.][†] の範囲である．この最小値である 60 [M.M.] に対応して窓長を 1 秒に設定した．原理的には，1 秒以上の窓長を用いれば，60 [M.M.] 以下のテンポ検出も可能である．

3. STPM リアルタイムビートトラッキング

提案する STPM リアルタイムビートトラッキングアルゴリズムの概要を **Fig. 1** に示す．自己発声音抑制，テンポ推定，ビート時刻推定の三つのモジュールからなっており，入力信号に対してビート時刻とテンポを出力する．

3.1 自己発声音抑制

自己発声音抑制には，2 ch 入力の SB-ICA [18] を用いる．一つめの入力チャネルには，ロボット搭載マイクで収音された音楽音響信号が入力される．この入力には，自己発声音などの雑音

[†] Mälzel's Metronome：一分あたりの四分音符の数．例えばテンポ 60 M.M. であれば，四分音符の長さは 1,000 [ms] である．

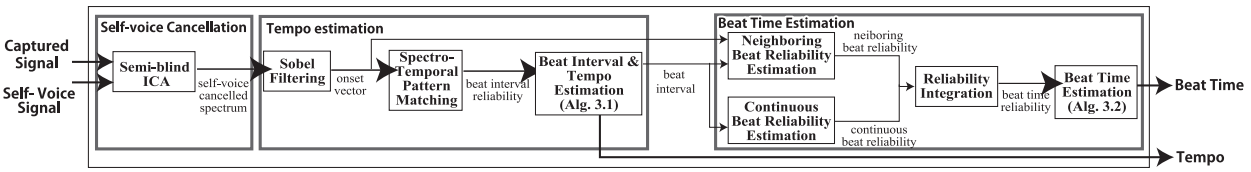


Fig. 1 Overview of SPTM real-time beat-tracking algorithm

が混入している．二つめの入力チャネルには，歌声や口ずさみといったロボット自己発声音が入力される．システムが生成する信号を直接入力しているため，周囲雑音は混入していない．また，発声用スピーカからロボット搭載マイクまでの伝達系の影響も受けていない信号であるため，クリーンな信号ではあるが，ロボット搭載マイクで収音された信号に含まれる自己発声音とは異なっている．これらの信号は，44.1 [kHz]，16 [bit] で同期してサンプリングされた後，窓長 4,096 [pts]，シフト長 512 [pts] で短時間フーリエ変換（STFT）を用いた周波数解析を行う．各チャネルに対して得られたスペクトルをそれぞれ，$Y(t,\omega)$，$S(t,\omega)$ とする．ここで t，ω はそれぞれ，時間フレームと周波数を表すインデックスである．このとき，自己発声音抑圧後の信号 $p(t,\omega)$ は，SB-ICA を用いれば，以下のように求めることができる（簡単のため，ω は省略）．

$$\begin{pmatrix} p(t) \\ S(t) \\ \vdots \\ S(t-M) \end{pmatrix} = \begin{pmatrix} A & W(0) & \cdots & W(M) \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} Y(t) \\ S(t) \\ \vdots \\ S(t-M) \end{pmatrix} \tag{1}$$

なお，M は残響を考慮するためのフレーム数であり，ここでは，発声用スピーカからロボット搭載マイクまでの伝達系によって M フレームに渡って残響が生じることを仮定している．実験的に $M = 8$ と定めた．A，W は分離フィルタであり，SB-ICA では，適応的にこれらの推定を行う．このように，入力，出力に既知信号 S を用いていること，伝達系による残響を考慮して自己発声音抑圧を行っていることから，従来手法と比較して高精度な抑圧を可能としている．

3.2 時間周波数テンプレートマッチングによるテンポ推定

テンポ推定は，スペクトログラムを整形する前処理部と時間周波数テンプレートマッチング（STPM）を用いたビート間隔推定部からなる．

3.2.1 前処理

前処理部では，まず，得られたスペクトル $p(t,\omega)$ に対し，音声認識や音楽認識で用いられるメルフィルタバンク適用し，周波数の次元数を 64 次元に圧縮する．得られたメルスケールでのパワースペクトルを $p_{mel}(t,f)$ とする．ここで f は，メル周波数軸での周波数インデックスを表す．スペクトログラム上で，パワーが急激に上昇している時刻はオンセットである可能性が高く，オンセットとビート時刻やテンポは密接な関係がある．そこで，時間方向のエッジ強調と周波数方向の平滑化を同時に行うことができる Sobel フィルタを適用する．Sobel フィルタは画像処理でエッジ抽出のために用いられる手法であり，$p_{mel}(t,f)$ に Sobel フィルタを適用した場合の出力 $p_{sobel}(t,f)$ は，下記のように表すことができる．

$$\begin{aligned} p_{sobel}(t,f) = &-p_{mel}(t-1,f+1) + p_{mel}(t+1,f+1) \\ &- p_{mel}(t-1,f-1) + p_{mel}(t+1,f-1) \\ &- 2p_{mel}(t-1,f) + 2p_{mel}(t+1,f) \end{aligned} \tag{2}$$

さらに，ビート時刻に対応するパワーの立ち上がり部のみを抽出するため，以下の処理を行い，各フレームごとに 62 次元の（$f = 1, 2, ..., 62$）オンセットベクトル $d(t,f)$ を定義する．

$$d(t,f) = \begin{cases} p_{sobel}(t,f) & \text{if } \mathrm{p_{sobel}(t,f)} > 0, \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

3.2.2 ビート間隔推定

隣り合う二つのビートの間隔を「ビート間隔」と定義する．得られたオンセットベクトルを用いてビート間隔の推定を行う．この際に正規化相互相関関数（NCC）を用いた時間周波数パターンマッチング（STPM）を行う．NCC の出力をビート間隔信頼度 $R(t,i)$ と呼ぶことにし，下記のように定義する．

$$R(t,i) = \frac{\sum_{j=1}^{F_w} \sum_{k=0}^{P_w-1} d(t-k,j)d(t-i-k,j)}{\sqrt{\sum_{j=1}^{F_w} \sum_{k=0}^{P_w-1} d(t-k,j)^2 \sum_{j=1}^{F_w} \sum_{k=0}^{P_w-1} d(t-i-k,j)^2}} \tag{4}$$

ここで，F_w は，オンセットベクトルのマッチングで用いる次元数であり，本稿では，62 次元すべてを用いた．また，P_w はパターンマッチングの窓長で i はシフトパラメータである．式（4）の分母に示される正規化項は信号処理でいう白色化に相当するため，Sobel フィルタでの雑音抑制に加えて定常雑音の抑制効果があること，ビート間隔信頼度の算出にに時間情報に加えて周波数情報も同時に利用していることから，STPM では雑音に対する処理の安定性を確保したまま，P_w の値を小さくすることが可能である．

次にビート間隔信頼度 $R(t,i)$ から実際のビート間隔を推定するため，まず，ローカルピークを抽出する．

$$R_{pk}(t,i) = \begin{cases} R(t,i) & \text{if } \mathrm{R(t,i-1) < R(t,i)} \\ & \text{\& } R(t,i) > R(t,i+1), \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

$R_{pk}(t,i)$ の上位二つの値を抽出し，その値に対応する i を $R_{pk}(t,i)$ の値の大きいほうからそれぞれ，$I_1(t)$，$I_2(t)$ とす

Algorithm 3.1: ESTBEATINTERVAL(I_1, I_2)

$I_c \leftarrow I_1$
if $\alpha R_{pk}(t, I_1) < R_{pk}(t, I_2)$
then
  $I_d \leftarrow |I_1 - I_2|$
  for $n \leftarrow 2$ **to** N_{max}
  do
    if $|I_1 - nI_d| < \delta$ **or** $|I_2 - nI_d| < \delta$
    then
      $I_c \leftarrow nI_d$
      break
$I \leftarrow$ ESTBEATINTERVAL($I_c, I(t-1)$)
return (I)

る．抽出した二つのピークの信頼度の差が大きい場合は，$I_1(t)$ をビート間隔候補 $I_c(t)$ とする．ただし，これらの差が小さい場合には，裏拍などの影響で，$I_1(t)$ が必ずしもビート間隔とはいえない．特に，整数分の整数倍（1/2，2/1，5/4，3/4，2/3，4/3 など）が間違いとして検出されやすいことを考慮し，$I_1(t)$，$I_2(t)$ の差を用いたビート間隔候補 $I_c(t)$ の推定を行う．具体的には，$I_1(t)$ と $I_2(t)$ の差を I_d とし，$|I_1(t) - nI_d|$ もしくは，$|I_2(t) - nI_d|$ が閾値以下の場合，nId を Ic とした．この際，n は，2 から N_{max} まで探索を行う．次に，$I_1(t)$，$I_2(t)$ から得られるビート間隔候補 $I_c(t)$ と前フレームのビート間隔 $I(t-1)$ を用いて，上記と同様の処理を行い，最終的なビート間隔 $I(t)$ の推定を行う．ビート間隔 $I(t)$ を推定するアルゴリズムを Algorithm 3.1 にまとめる．なお，α，δ は，それぞれ，実験的に 0.7，5 とした．また，N_{max} は，四分音符までを考慮して 4 とした．

テンポ $I_m(t)$ は，推定した T_I フレーム分のビート間隔群に対する中央値として下記のように定義する．

$$I_m(t) = \mathrm{median}(I(t_i)) \quad (t_i = t, t-1, \cdots, t-T_I) \tag{6}$$

なお，T_I は，実験的に 13 フレーム（約 150 [ms]）とした．

3.3 ビート時刻推定

ビート時刻を推定するため，ビート時刻信頼度を算出する．ビート時刻信頼度を算出するため，「近接ビート信頼度」と「連続ビート信頼度」を導入する．近接ビート信頼度は，あるフレームとそのビート間隔前のフレームがともにビート時刻である信頼度であり，連続ビート信頼度は各時刻において推定されたビート間隔で，ビートが連続的に存在しているかを示す信頼度である．処理フレーム t ごとにフレーム $t-i$ とその 1 ビート間隔分前のフレーム $t-i-I(t)$ が共にビート時刻である信頼度，つまり近接ビート信頼度 $S_c(t, t-i)$ を，オンセットベクトルを用いて以下のように定義する．

$$\begin{aligned} S_c(t, t-i) &= F_s(t-i) + F_s(t-i-I(t)), \quad (0 \le i \le I(t)) \\ F_s(t) &= \sum_{f=1}^{F_w} d(t, f) \end{aligned} \tag{7}$$

また，処理フレーム t におけるフレーム $t-i$ の連続ビート信頼度 $S_r(t, t-i)$ を，近接ビート信頼度を用いて以下のように定義する．

Algorithm 3.2: ESTBEATTIME($t, T(n), S(t), I(t)$)

procedure SEARCHPEAKS($S(t), t, t_r, N_{max}$)
  search N_{max} peaks in $S(t)$ within $t \pm \frac{1}{2}t_r$.
comment: $t[i]$ is an array of peak times.
  N_p is the size of the array.
return ($t[i], N_p$)

main
$T(n+1) \leftarrow nil$
if $t > T(n) + \frac{3}{4}I(t)$
then
  ($t[i], N_p$)
    $\leftarrow$ SEARCHPEAKS($S(t), T(n), I(t), 3$)
  if $N_p > 0$
  then
    $i_{min} \leftarrow \underset{i}{\mathrm{argmin}}(|T(n) + I(t) - t[i]|)$
    $T(n+1) \leftarrow t[i_{min}]$
  else $T(n+1) \leftarrow T(n) + I(t)$
return ($T(n+1)$)

$$\begin{aligned} S_r(t, t-i) &= \sum_{m}^{N_{S_r}} S_c(T_p(t, m), i), \quad (0 \le i \le I(t)) \\ T_p(t, m) &= \begin{cases} t & (m = 0) \\ T_p(t, m-1) - I(T_p(t, m-1)) & (m \ge 1) \end{cases} \end{aligned} \tag{8}$$

$T_p(t, m)$ はフレーム t を基準として m 個前のビート時刻で，N_{S_r} は連続ビート信頼度を評価する際に考慮するビート数である．連続ビート信頼度は複数のビート列が見つかった場合に，どのビート列が最も信頼できるかを判定する際に有効である．処理フレーム t におけるフレーム $t-i$ のビート時刻信頼度を近接ビート信頼度と連続ビート信頼度を用いて以下のように定義する．

$$S'(t, t-i) = S_c(t, t-i) S_r(t, t-i) \tag{9}$$

S' 間の時間的重なりを考慮するため，以下の加算平均を用いて最終的なビート時刻信頼度 $S(t)$ を得る．

$$S(t) = \frac{1}{N_{S'}(t)} \sum_{t_i \in S'_t(t)} S'(t_i, t) \tag{10}$$

なお，$S'_t(t)$，$N_{S'}(t)$ は，フレーム t で値を持つ S' の集合，およびその集合の要素数を表す．

次に，ビート時刻信頼度 $S(t)$ を用いてビート時刻を推定する．n 番目のビート時刻 $T(n)$ が得られており，$T(n+1)$ を推定するものとする．現在の処理フレームが，$T(n)$ にビート間隔 $I(t)$ の 3/4 倍を加えた時刻を超えていれば，ビート信頼度信頼度から $t \pm I(t)/2$ の範囲内で，最大三つのピークを抽出する．ピークが存在すれば，その時刻の中で，最も，$T(n) + I(t)$ に近いピークを $T(n+1)$ とする．ピークが存在しない場合は，$T(n) + I(t)$ を $T(n+1)$ とする．アルゴリズムを Algorithm 3.2 に示す．

4. 音楽ロボットの実装

提案した STPM リアルタイムビートトラッキングを用いて構築した音楽ロボットのアーキテクチャを **Fig. 2** に示す．このシステムは主にロボットと三つのサブシステム（リアルタイム

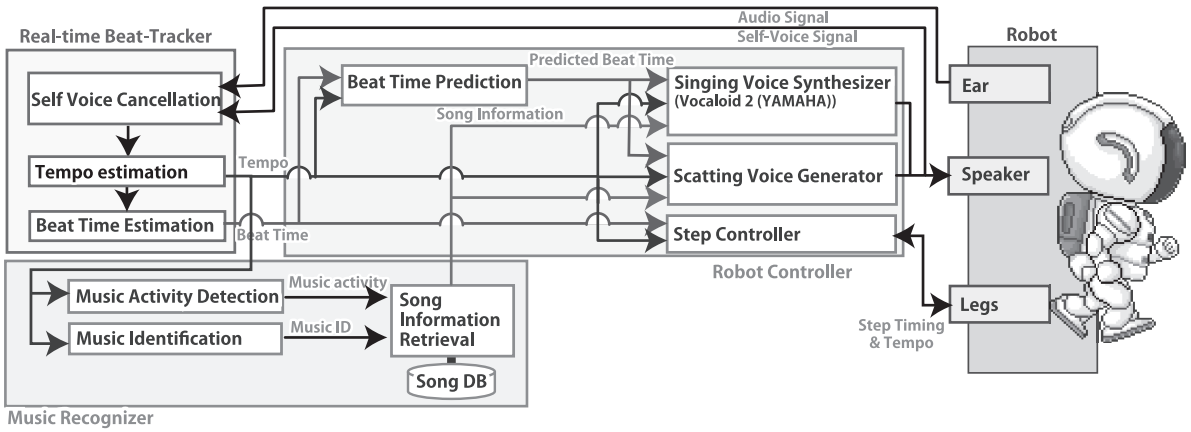


Fig. 2 Architecture of a music robot

ビートトラッカー部，音楽認識部，ロボット制御部）に分けられる．

4.1 ロボットおよびハードウェア構成

ロボットには，Honda ASIMO を用いた．音楽信号収音用に，頭部にマイクを 1 本搭載しており，胸部には発声用にスピーカを内蔵している．ASIMO の動作は，本稿では足踏みのみを扱った．足踏み制御は足踏み間隔のみで行うインタフェースとなっている．また，ハードウェア的な制約から，足踏み間隔は 500〜1,000 [ms] の範囲となっている．これは音楽のテンポに換算すると 61〜120 [M.M.] である．

リアルタイムビートトラッカー部と音楽認識部は，MacOSX 上で C++ で実装されており，Core2Duo を搭載した 1 台の PC 上で動作している．この PC には，2 ch 信号が入力される．頭部マイクで収音される信号を 1 番目のチャネルに，発声用のスピーカに出力される信号を分岐したものを 2 番目のチャネルに入力している．また，ロボット制御部のうち，足踏み制御，口ずさみ音声出力処理については，リアルタイムビートトラッカー部と同じ PC 上で動作する．ただし，歌唱機能については，商用のソフトを用いている関係上，Windows PC を用いた．

4.2 リアルタイムビートトラッカー部

リアルタイムビートトラッカー部には，提案手法である STPM リアルタイムビートトラッキングを用いている．ただし，前述のロボットのハードウェア的制約を加味し，テンポ推定をロボットの動作範囲である 61 [M.M.] から 120 [M.M.] に制限している．

4.3 音楽認識部

歌唱機能を実現するためには，ビート情報のほかに曲名，歌詞，譜面，歌い出しのタイミングといった情報が必要である．こうした情報も入力音楽音響信号から認識し，認識した情報をもとに自動的に検索できることが望ましい．そこで，構築した音楽ロボットでは，歌の開始・終了のタイミングを検出するため，1）テンポ情報に基づく音楽区間検出機能を実装した．さらに，歌詞，譜面情報を取得するため，2）テンポ情報に基づく曲名同定と曲名をキーとした曲情報検索機能を実装した．これらの機能は，テンポ情報のみを用いて実現されているため，ビートが推定できれば利用可能であるとともに，アルゴリズムが単純であるため，処理遅れが少ないという利点がある．

4.3.1 音楽区間検出

音楽区間検出では，音楽区間は，安定したビート間隔が得られると仮定する．現在の処理フレームを t としたとき，過去 A_w 個のフレームのうち，$|I(x) - I(t)| < \alpha$ を満たすフレーム x の総数を N_x とする．なお，α はビート間隔の許容誤差を表す．ビート間隔安定度 S を

$$S = \frac{N_x}{A_w} \tag{11}$$

と定義する．実験的に $A_w = 300$（約 3.5 [s] に相当），$\alpha = 5$（58 [ms] に相当）を用い，安定度 S が 0.8 以上の場合に音楽区間であるとした．

4.3.2 曲名同定と曲情報検索

曲名同定では，同じテンポを持つ曲は存在しないと仮定し，推定されたテンポに最も近いテンポを持つ曲の ID を出力する．ただし，推定テンポに近いテンポを持つ曲が存在しない場合には，「unknown」に対応する曲 ID を出力する．この仮定は，曲数が多い場合には成立しにくくなる．より多くの曲が扱えるようスケーラビリティを向上するためには，ピッチや複雑なビート構造など他の情報も利用する必要があろう．しかし，本稿では提案したビートトラッキング法の有効性・有用性を検証することを主眼にしており，この問題は今後の課題とする．

次に得られた曲 ID をキーとして曲情報データベースから歌詞情報と譜面情報を取得する．曲 ID が unknown の場合には，歌唱を行うことはできないため，代わりに，口ずさみを行うよう命令をロボット制御部に送出する．

4.4 ロボット制御部

ロボット制御部はリアルタイムビートトラッカーによって検出されたビート時刻とビート間隔，音楽認識によって検索された曲情報を用いて，ビート時刻に同期して足踏み，歌唱，口ずさみを行う．

4.4.1 ロボットの挙動

ロボットの挙動は，歌唱，口ずさみ，足踏みを扱っている．歌唱機能の歌声合成エンジンには VOCALOID2 [6] を用いた．VOCALOID2 は，MIDI を拡張することで，音符の発音長，音階，発音記号入力することを可能にしており，これらの情報を用いて自然な歌声を合成することができる．実際には，曲に同

期して歌を歌うことができるように，曲情報検索で得られる譜面情報のうち，音の時刻と長さを抽出したビートに応じて変換した後，歌声合成エンジンに送出している．口ずさみは，ビート時刻に合わせて「ずん」「ちゃ」と口ずさむ機能である．ビート時刻に合わせて自然に聞こえるよう口ずさむためには「ずん」や「ちゃ」の各音とビート時刻の同期のタイミングが重要である．そこで「ずん」,「ちゃ」から抽出したオンセット時刻ベクトルの各値の合計値のピークを「ずん」「ちゃ」のビート時刻とする．この各音でのビート時刻と，音楽のビート時刻を合わせて発音する．ロボットの足踏みに関しては，足の接地時刻とビート時刻，ビート間隔を用いた簡単なフィードバック則を用いて制御した [17].

4.4.2　ビート時刻予測

リアルタイムビートトラッカー部では，オンライン処理が可能であるが，一定の遅延が発生する．主な遅延は，式（10）に示すビート時刻信頼度の算出と Algorithm 3.2 に示すビート時刻推定である．フレーム t のビート時刻信頼度 $S(t)$ を算出する際には，すべての t_i が揃うまで待つ必要がある．t_i の最大値は，$t+max(I(t_i))$ で規定される．$I(t_i)$ の最大値は，アルゴリズム上 60 [M.M.] に対応するフレーム数であるため，NCC の窓長と同じ 1 [s] となる．ビート時刻推定では，$t=T(n)+3/4I(t)$ の際のピーク抽出に，$T(n)+3/2I(t)$ までビート時刻信頼度が必要である．つまり，フレーム t のビート時刻信頼度が得られてから，$3/4I(t)$ だけ待つ必要がある．この最大値は，0.75 [s] である．また，SB-ICA で M フレーム分，Sobel フィルタで 1 フレーム分の遅延が生じるため，リアルタイムビートトラッカー部では約 2 [s] の避けられない遅延が発生する．さらに，ロボット制御部では，歌唱生成部が約 0.3 [s] の遅延が発生するため，合計で約 2.3 [s] の遅延が発生する．このため，リアルタイムで歌唱機能を実現するには，この遅延に対するビート時刻予測が必要である．本稿では，単純にテンポを外挿することで，ビート時刻予測を実現する．具体的には，$T(n)$ をフレーム t までに推定されたうちの最新のビート時刻としたとき，フレーム t より未来のビート時刻に対応するフレームのうち最も t に近いフレーム T' を以下のように定義する．

$$T' = \begin{cases} T_{\mathrm{tmp}} & \text{if } \mathrm{T_{tmp}} \geq \frac{3}{2}\mathrm{I_m(t)} + \mathrm{t} \\ T_{\mathrm{tmp}} + I_m(t) & \text{otherwise.} \end{cases} \tag{12}$$

$$T_{\mathrm{tmp}} = T(n) + I_m(t) + (t - T(n)) - \{(t - T(n)) \mod I_m(t)\}$$

なお，足踏み生成に関しては，足踏み間隔しか制御できないことから，予測機能を用いる代わりに，前述のように簡単なフィードバック制御を用いている．

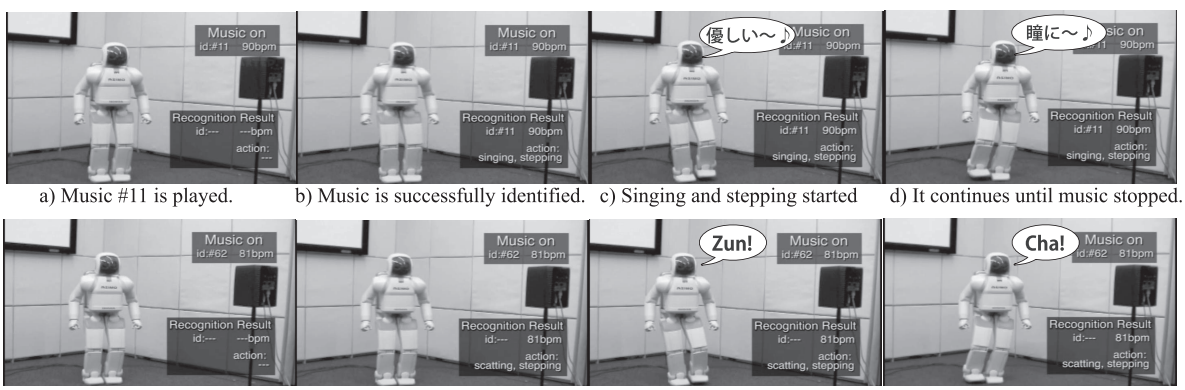
5. 評　　価

構築した音楽ロボットの動作例を **Fig. 3** に示す．正面のスピーカから，音楽（インストルメンタル）が流れ，ロボット搭載マイクの入力からその音楽のビートを検出する．検出したビートから曲名が判定できれば，ビートに合わせて足踏みを行いながら，歌唱を行う（Fig. 3 a)〜d)）．判定できない場合は，口ずさみ音声を出力する（Fig. 3 e)〜h)）．この音楽ロボットのビートトラッキング性能を以下のように評価した．

実験 1　ビートトラッキングの基本性能
実験 2　テンポ変化への追従速度
実験 3　ビート予測のノイズロバスト性能

実験 1 用の評価データとして，市販の CD のように様々な楽器音と歌声を含んでいる RWC 音楽データベース（RWC-MDB-P-2001）[3] のポピュラー音楽全 100 曲を使用した．それぞれの曲は，正解ビート時刻を簡単に取得するため，MIDI データを用いて生成した．ただし，MIDI データの情報は，得られたビート時刻の評価のためのみに用い，ビートトラッキング時には用いないものとした．各曲の開始 30〜90 秒までの 60 秒間を評価データとして用い，提案手法と自己相関関数ベースの手法 [17] でビート検出正解率（r_{rme}）を測定した．ビート検出正解率の算出では，推定ビート時刻と正解ビート時刻の差が ±100 [ms] 以内に収まっている場合に成功したものとした．この値は，二つの音の立ち上がりタイミングが ±100 [ms] 以内の場合，一つの音に知覚されるという時間知覚研究の知見 [13] を考慮して決定した．具体的な定義を下記に示す．

$$r_{\mathrm{e}} = \frac{N_{\mathrm{e}}}{N_{\mathrm{d}}} \times 100. \tag{13}$$



a) Music #11 is played.　b) Music is successfully identified.　c) Singing and stepping started　d) It continues until music stopped.

e) Music #62 is played. (#62 is not in the music DB.)　f) Tempo is successfully estimated, but music identification failed.　g) Scatting and stepping started　h) It continues until music stopped.

Fig. 3　Snapshots of music robot behavior

なお，N_e は推定成功ビート数，N_d は検出ビート総数である．

実験 2 用の評価データとして RWC-MDB-P-2001 から生演奏録音の 3 曲を選び，テンポ変化を含む音楽音響信号を作成した．具体的には，No. 11，No. 18，No. 62 を選択し（テンポはそれぞれ 90，112，81 [M.M.]），これらを No.18–No.11–No.18–No.62 の順に 60 秒ずつ区切りでつなげることで 4 分の音楽信号を作成した．この評価データを用いて 5 種類の状態でのビートトラッキングの遅延を提案手法と自己相関関数ベースの手法で計測した．ビートトラッキングの遅延は実際にテンポが変化してからシステムがテンポ変化に追従するまでの時間とする．5 種類のうち 2 種類は ASIMO の電源を off にして口ずさみを行った場合と行わなかった場合，他の 3 種類は ASIMO の電源を on にして足踏みをしながら口ずさみを行った場合，行わなかった場合，歌った場合である．

実験 3 用の評価データとして，RWC-MDB-P-2001 の No. 62 の MIDI データを用いて生成したテンポ一定の曲を用いた．ただし，実験 1 と同様，MIDI のデータはビート時刻の検証のみに用いる．5 種類のうち 3 種類は ASIMO の電源を切った状態で，1 種類は口ずさみなしで自己発声音抑制あり，他の 2 種類はともに口ずさみありだが，自己発声音抑制の有無に違いがある．他の 2 種類は ASIMO の電源を入れ足踏みをしながら口ずさみを行っている状態で自己発声音抑制を行った場合と行わなかった場合である．評価指標には，ビート予測成功率（r_{rmp}）を用いた．

$$r_p = \frac{N_p}{N_c} \times 100. \qquad (14)$$

なお，N_p は予測成功ビート数，N_c は正解ビート総数である．

実験はいずれも残響時間 0.2 秒（RT_{20}），4 [m] × 7 [m] の部屋で行い，音楽用音源にはスピーカ（GENELEC 1029A）を用いた．ロボットとスピーカの距離は 1.5 [m] である．

5.1 実験結果

実験 1 の結果を **Fig. 4** に示す．結果は，入力曲のテンポに応じて，60 [bpm]〜10 [bpm] の範囲ごとにクラスタリングを行い，各範囲ごとに平均ビート検出正解率を算出した．横軸は，クラスタリングを行ったテンポ範囲を示している．縦軸は各範囲ごとのビート検出正解率を示している．図から，ほぼ，すべてのテンポ範囲で，提案法が優れていることが分かる．また，テンポが遅い場合，従来法，提案法ともに，ビート検出正解率の低下がみられる．これは，テンポの遅い曲は，ドラムなどテンポ抽出の鍵となる楽器が少ない音楽であったためである．これに対して，ドラムなどのテンポ抽出が容易な楽器が含まれる 110 [bpm] 以上の曲では，良好なビート検出正解率が得られている．

実験 2 の各条件における遅延平均時間を **Table 1** に示す．また，**Fig. 5** は ASIMO 電源が OFF の場合の推定結果例である．総じて，提案手法は従来手法に比べテンポ変化への適応が速いことが分かる．Table 1 では，提案手法が従来手法より口ずさみのない場合で 10 倍程度，ある場合は，20 倍程度追従が高速であることが分かる．また，Fig. 5 の 100 秒近くでビートトラッキングが乱れているのは，ビート時刻にオンセットがない部分が評価データに一時的に存在しているからである．このため提案手法では一時的に，従来手法では長期に渡って結果が不安定になる．提案手法を用いた音楽ロボットは，音楽区間検出機能を有しており，ビートが抽出できない区間は非音楽区間であると判定されるため，音楽ロボットでは，このような部分の影響は少ない．

実験 3 の結果を **Table 2** に示す．“Correct” はビートトラッキングシステムが正しいビート（表拍）を予測した事を示し，“Up-Beat Error” は間違えて裏拍を予測した事を示す．これは自己発声音がその周期性のためにビートトラッキングに影響を与えていること，および自己発声音抑制がこうしたノイズに効果的に働くことを示している．

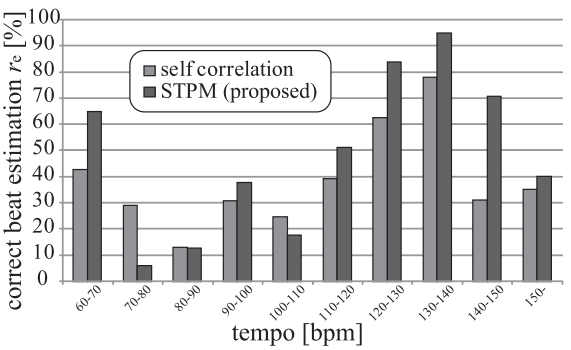


Fig. 4 Performance of beat-tracking

Table 1 Tracking delay for tempo changes [s]

	ASIMO power off		ASIMO power on (with stepping)		
	w/o scatting	w/ scatting	w/o scatting	w/ scatting	w/ singing
STPM	1.31	1.31	1.29	1.29	1.29
Self-correlation	11.24	29.91	14.66	20.43	N/A

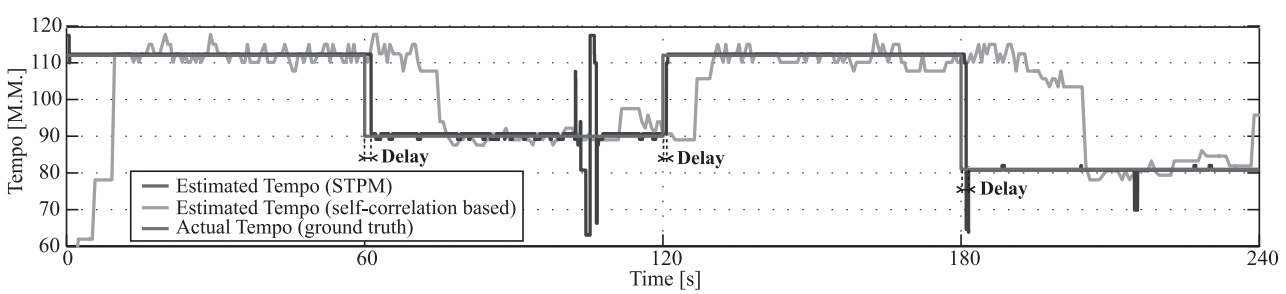


Fig. 5 Result of using music of tempo chage

Table 2 Beat prediction success rate r_{p} [%]

	ASIMO power off			ASIMO power on (with step)	
	w/o scatting	w/ scatting		w/ scatting	
		w/ self-voice cancel	w/o self-voice cancel	w/ self-voice cancel	w/o self-voice cancel
Correct	73%	76%	54%	77%	54%
Up-Beat Error	3%	1%	22%	2%	28%

6. おわりに

人・ロボット音楽インタラクションの実現を目指して，雑音に対するロバスト性と，テンポ変化に対する高速追従性を兼ね備えた，STPM リアルタイムビートトラッキングを提案した．また，口ずさみや歌などの周期性を伴った自己発声音の影響を削減するためにセミブラインド ICA による自己発声音抑制を併せて導入した．さらに，提案した手法を Honda ASIMO に実装し，ロボット搭載マイクを用いて収音した信号からリアルタイムでビートを抽出し，そのビート情報に基づき，足踏み，口ずさみ，歌唱を行う音楽ロボットを構築し，対雑音ロバスト性とテンポ変化に対する高速追従性を確認した．ロボットの挙動の高度化，ロボットの自己発声音以外の自己発生音や環境雑音の抑制，高度な音楽認識機能の導入など，実用的な音楽ロボットの実現には，課題が残されている．しかし，これまで，ロボット分野では，ロボットの耳を用いて音楽を含む音響信号を処理する研究の歴史は浅く，音楽情報処理の分野では，ロボットのような物理的な身体を伴うエージェントやユーザインタフェースに対する研究が少なかった．こうした音楽ロボットの実現を目指した研究は，単にアプリケーションを構築するだけではなく，音楽情報処理とロボティクスを融合した新たな研究領域の創出が期待できよう．

謝 辞 ご助言をいただいた東京工業大学 井村順一教授，早川朋久准教授に感謝する．

参考文献

[1] A. Cemgil and B. Kappen: “Monte carlo methods for tempo tracking and rhythm quantization,” Journal of Artificial Intelligence Research, vol.18, pp.45–81, 2003.

[2] R. Dannenberg and B. Mont-Reynaud: “Following an improvisation in real time,” Proceedings of the International Computer Music Conference, pp.241–258, 1987.

[3] M. Goto, H. Hashiguchi, T. Nishimura and R. Oka: “Rwc music database: Popular, classical, and jazz music databases,” Int. Conf. Music Information Retrieval, pp.287–288, 2002.

[4] M. Goto and Y. Muraoka: “A real-time beat tracking system for audio signals,” Proceedings of the International Computer Music Conference, pp.171–174, 1995.

[5] S.W. Hainsworth and M.D. Macleod: “Particle filtering applied to musical tempo tracking,” EURASIP Journal on Applied Signal Processing, vol.15, pp.2385–2395, 2004.

[6] H. Kenmochi and H. Ohshita: “Vocaloid—commercial singing synthesizer based on sample concatenation,” INTERSPEECH-2007, pp.4011–4010, 2007.

[7] A.P. Klapuri, A.J. Eronen and J.T. Astola: “Analysis of the meter of acoustic musical signals,” IEEE Trans. Audio, Speech, and Language Processing, vol.14, no.1, 2006.

[8] K. Kosuge, T. Hayashi, Y. Hirata and R. Tobiyama: “Dance partner robot—ms dancer—,” Proc. of IEEE/RSJ Int’l Conf. on Intelligent Robots and Systems (IROS-2003), pp.1743–1750, 2003.

[9] S. Kotosaka and S. Schaal: “Synchronized robot drumming by neural oscillators,” Proc. of Int’l Sympo. Adaptive Motion of Animals and Machines, 2000.

[10] M.P. Michalowski, S. Sabanovic and H. Kozima: “A dancing robot for rhythmic social interaction,” Proc. of ACM/IEEE Int’l Conf. on Human-Robot Interaction (HRI 2007), pp.89–96. IEEE, 2007.

[11] K. Nakadai, T. Lourens, H.G. Okuno and H. Kitano: “Active audition for humanoid,” Proc. of 17th National Conference on Artificial Intelligence (AAAI-2000), pp.832–839. AAAI, 2000.

[12] A. Nakazawa, S. Nakaoka, K. Ikeuchi and K. Yokoi: “Imitating human dance motions through motion structure analysis,” Proc. of IEEE/RSJ Int’l Conf. on Intelligent Robot s and Systems (IROS-2002), pp.2539–2544, 2002.

[13] R.A. Rasch: “Synchronization in performed ensemble music,” Acustica, vol.43, no.2, pp.121–131, 1979.

[14] E. Scheirer: “Tempo and beat analysis of acoustic musical signals,” Journal of the Acoustical Society of America, vol.103, no.1, pp.588–601, 1998.

[15] T. Takeda, Y. Hirata and K. Kosuge: “Hmm-based dance step estimation for dance partner robot—ms dancer—,” Proc. of IEEE/RSJ Int’l Conf. on Intelligent Robots and Systems (IROS-2005), pp.1602–1607, 2005.

[16] T. Takeda, Y. Hirata, Z. Wang and K. Kosuge: “Hmm-based error detection of dance step selection for dance partner robot—MS DanceR—,” Proc. of IEEE/RSJ Int’l Conf. on Intelligent Robots and Systems (IROS-2006), pp.5631–5636, 2006.

[17] K. Yoshii, K. Nakadai, T. Torii, Y. Hasegawa, H. Tsujino, K. Komatani, T. Ogata and H.G. Okuno: “A biped robot that keeps steps in time with musical beats while listening to music with its own ears,” Proc. of IEEE/RSJ Int’l Conf. on Intelligent Robots and Systems (IROS-2007), pp.1743–1750, 2007.

[18] 武田龍，中臺一博，駒谷和範，尾形哲也，奥乃博：“独立成分分析に基づく適応フィルタのロボット聴覚への適用”，日本ロボット学会誌，vol.26, no.6, pp.529–536, 2008.

村田和真（Kazumasa Murata）

2007 年東京工業大学工学部制御工学科卒業．2009 年同大学大学院情報理工学研究科情報環境学専攻修了．2009 年 4 月より（株）ソニー勤務．音楽ロボットの研究に従事．IEEE-RSJ/IROS-2008 NTF Award Finalist．人工知能学会会員．

（日本ロボット学会正会員）

武田　龍（Ryu Takeda）

2006 年 京都大学工学部情報学科卒業，2008 年同大学情報学研究科知能情報学専攻修士課程修了，現在同大学博士後期課程に在学中．IEEE-RSJ/IROS-2006 Best Paper Nomination Finalist．情報処理学会，IEEE などの会員．

（日本ロボット学会学生会員）

長谷川雄二（Yuji Hasegawa）

1986 年東京都立大学理学部生物学科卒業，同年本田技研工業（株）入社．1987 年（株）本田技術研究所配属．2003 年より（株）ホンダ・リサーチ・インスティチュート・ジャパン，プリンシパル・エンジニア．画像処理研究，神経系の情報処理機構，特に昆虫の視覚情報処理の研究に従事．日本動物学会，日本神経回路学会などの会員．

中臺一博（Kazuhiro Nakadai）

1993 年東京大学工学部電気工学科卒業，1995 年同大学大学院工学系研究科情報工学専攻修了．NTT，NTT コムウェア，JST ERATO 北野プロジェクトを経て，2003 年より（株）ホンダ・リサーチ・インスティチュート・ジャパン，シニア・リサーチャ．博士（工学）．2006〜2008 年東京工業大学大学院情報理工学研究科客員准教授，2009 年より同連携准教授兼務．ロボット聴覚，実時間情報統合，音環境理解の研究に従事．人工知能学会，ヒューマンインタフェース学会，日本音響学会，IEEE などの会員．

（日本ロボット学会正会員）

奥乃　博（Hiroshi G. Okuno）

1972 年東京大学教養学部基礎科学科卒業．NTT，JST 北野プロジェクト，東京理科大学理工学部を経て，2001 年 4 月より京都大学大学院情報学研究科知能情報学専攻教授．博士（工学）．音環境理解，ロボット聴覚，推論機構の研究に従事．

（日本ロボット学会正会員）

辻野広司（Hiroshi Tsujino）

1984 年東京工業大学理学部情報科学科卒業．1986 年同大学大学院情報科学専攻修士課程修了．1987 年（株）本田技術研究所入社．2003 年より（株）ホンダ・リサーチ・インスティチュート・ジャパン，チーフ・リサーチャ．脳型コンピュータ，知能システム，ヒューマンロボットインターフェース，画像認識などの研究に従事．IEEE，SFN，INNS，人工知能学会，日本ソフトウェア科学会などの会員．

（日本ロボット学会正会員）