RE-001

リズムチューナー：アノテーション情報を用いない リアルタイム発音検出によるリズム練習支援システム

RhythmTuner: A Rhythm Practice Support System using Real-time Onset Detection without Annotation

山田 昌尚†，松尾 章弘†，峯 恭子‡，土江田 織枝†
Masanao YAMADA, Akihiro MATSUO, Kyoko MINE, Orie DOEDA

1. まえがき

音楽演奏能力を向上させるうえで，音高とリズムを適切にコントロールできるようになることは基礎的かつ重要な要素のひとつである．音高やリズムを向上させる練習ではピアノやキーボード，メトロノームなどを用いて演奏者自身が判断し修正できるようになることが望ましいが，初心者にはそのような判断は難しい．また，ある程度演奏に習熟した中級者にとっても，演奏しながらその場で演奏内容の詳細を判断することは難しいため，演奏を録音して聞き直すことがしばしば行われている．このような練習を補助する手段として，音高に関してはチューナーが安価で普及しており，演奏者が自身の演奏を視覚的，定量的に知ることができる．しかしリズムに関してはそのような外部的な補助手段で客観的に可視化する方法は一般化していない．

そこで本研究では，楽器練習者が演奏したリズムをリアルタイムに可視化することでリズムトレーニングを支援するシステムを提案し，その評価を行う．対象とする楽器としては管楽器等の単音で演奏するものを考え，システムへの入力は音響信号とする．特に，任意の楽譜を練習できるように，アノテーション情報を使用せずにパラメータを設定する方法について提案し，実験を行う．

2. 関連研究と本研究の位置づけ

楽器演奏を対象とした練習システムに関する研究はこれまでにいくつか行われている．楽器の種類を限定しない練習支援システムとして Interactive Music Tuition System (IMUTUS) [1] は初心者が簡単な曲をトレーニングするためのもので，録音，再生，演奏評価を行うことができる．またこのシステムはリアルタイム信号処理を用いた楽譜追跡と，オフラインでの発音検出を用いた詳細なフィードバックを行うことができる．IMUTUS はその後 Virtual European Music School (VEMUS) プロジェクト [2] に発展した．VEMUS は個人練習に加えてグループ学習や遠隔学習の機能を提供するものである．特定の楽器に関する研究としてはピアノ [3]，ドラム [4]，ヴァイオリン [5]，チェロ [6]などを対象としたものがあるほか，商用ソフトウエアも存在している [7]．これらの先行研究には，本研究が着目するリズム練習の要素を含むものもあるが，入力インターフェースに MIDI を使用したり，練習の対象を特定の曲に限定している．

これに対して筆者らはこれまで音響信号を入力として単音楽器による演奏を取り扱うことができるシステムを提案してきた [8, 9]．また本システムの特徴のひとつであるリズムに関するリアルタイム処理は上記の先行研究にみられない特長である．一方，これまでの我々の研究においては，発音検出に用いるパラメータをアノテーション情報にもとづいて決定する必要があるという問題があった．そこで本論文では，アノテーション情報なしで発音検出に用いるパラメータ設定を行う方法を提案し，その実験結果を述べる．これにより，任意の演奏を扱うことができる練習支援システムとしての実現性を大きく前進させることになる．以降，本システムを「リズムチューナー」と呼ぶこととする．

3. システムの構成と動作

図 1 にリズムチューナーの構成を示す．構築環境は Windows である．ユーザからシステムへの情報入力およびシステムからの表示は Processing で行い，電子メトロノーム音の生成とマイク入力からの信号処理に ChucK を用いる．ChucK はプリンストン大学で開発されたリアルタイム音楽処理用のプログラミング言語である．後述する OSC を用いることができる点や，音の生成，信号処理を並列処理できることから ChucK を用いた．Processing と ChucK の間で，ユーザが指定したテンポ情報のほかメトロノーム音や発音検出のタイミングをリアルタイムで送受信するために Open Sound Control (OSC) を用いる．OSC は電子楽器およびコンピュータ間で音楽演奏データ等を送受信するための通信プロトコルである．OSC では URL 形式でデータを送るため，各種の情報を容易に区別して情報伝達をできる．

マイク入力からの A/D 変換およびスピーカ出力への D/A 変換には，フリーの ASIO エミュレーションドライバであ

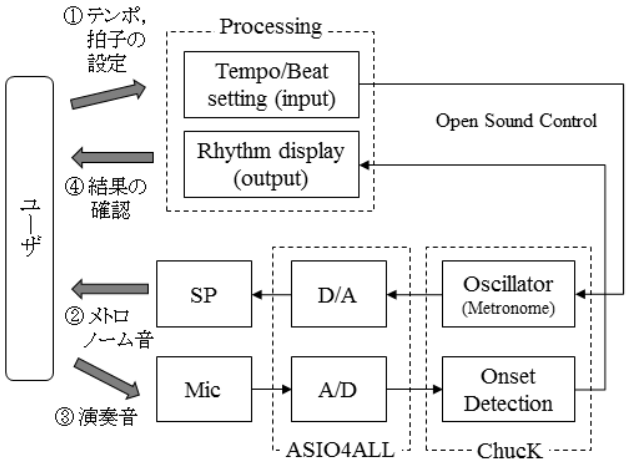


図 1　リズムチューナーのシステム構成

† 釧路工業高等専門学校　National Institute of Technology, Kushiro College
‡ 大阪大谷大学　Osaka Ohtani University

る ASIO4ALL を使用する．これは A/D 変換に Windows 標準のオーディオエンジンを使用すると 100～200ms のレイテンシーが生じるためである．モノラル入力で A/D 変換のサンプリング周波数を 44.1kHz とし，この信号を 1024 サンプルのフレーム幅でハニング窓をかけて短時間フーリエ変換 (STFT) により時間周波数情報に変換した．STFT のホップ幅は 10 ms であり，これがシステムの時間解像度となる．この信号を次節で述べる方法で発音検出する．

システムの動作は以下のとおりである．

① ユーザはまず演奏するテンポと拍子を設定する．
② 設定したテンポと拍子に従って，システムはメトロノーム音を生成しスピーカから音を鳴らす．このとき強拍と弱拍は周波数で区別する．強拍は 880 Hz，弱拍は 440 Hz とした．
③ ユーザはメトロノーム音を聴きながら楽器を演奏する．システムは演奏された音をマイクから取り込んで発音検出する．このときシステムが再生したメトロノーム音とユーザの演奏した音が重なって発音検出が適切に行われない場合はイヤホンなどを使用する．
④ 検出した発音タイミングをメトロノーム音のタイミングとともにリアルタイムでディスプレイに表示する．

実際の表示形式については 5.2 節で実験結果とともに述べる．

4. 発音検出としきい値設定の方法

リズムチューナーの中心となる発音検出の方法と，アノテーション情報を用いずにしきい値を設定する方法について以下に述べる．

4.1.発音検出の方法

音響信号から発音のタイミングを得る発音検出は一般的に，前処理，検出関数，ピーク抽出の 3 段階からなる [10]. 第 1 段階の前処理としては正規化を施す．第 2 段階の検出関数として，周波数ごとの信号スペクトル強度を利用するスペクトルフラックスや High Frequency Content (HFC)，位相の変化を利用する方法など，各種が提案されている．本研究では，周波数ごとの信号スペクトル強度の変化が大きい場合に発音となるピークが現れるスペクトルフラックスを使用する．

まず，音響信号 $x(t)$ を短時間フーリエ変換 (Short Time Fourier Transform, STFT) を用いて時間周波数領域でのスペクトログラム $X(n,k)$ を表すと次のようになる．

$$X(n,k) = \sum_{m=1}^{N-1} x(nh+m)\, w(m)\, e^{-\frac{2j\pi mk}{N}} \qquad (1)$$

ここで n はスペクトログラム上の時間，k は周波数，N は FFT フレームサイズ，h は STFT のホップサイズ，$w(t)$ は窓関数である．次に，検出関数として次式で表されるスペクトルフラックスを求める．

$$SF(n) = \sum_{k=1}^{\frac{N}{2}-1} \max(0, |X(n,k)| - |X(n-1,k)|) \qquad (2)$$

ここでNは FFT フレームのデータ数，nは時間，kは周波数であり，$X(n,k)$ はスペクトログラム (時間周波数信号) を表す．0 とのmaxをとることでスペクトル強度が増加する場合のみを対象としている．

第 3 段階のピーク抽出では，しきい値を超える局所最大値 (local maxima) を検出し，その時刻を発音時刻とする．しきい値として次式による動的しきい値を用いる．

$$TH(n) = \delta + \lambda \cdot \mathrm{median}(SF(n-v_1{:}n-v_2)) + \alpha \cdot \mathrm{mean}(SF(n-v_1{:}n-v_2)) \qquad (3)$$

ここで δ はしきい値の定数項，λ および α はそれぞれ中央値，平均値に対する重みであり，v_1, v_2 は動的しきい値の対象幅を表す．このしきい値を用いて，検出関数 $DF(n)$ および発音 $OD(n)$ を次のように求める．

$$DF(n) = SF(n) - TH(n) \qquad (4)$$

$$OD(n) = \begin{cases} 1, & DF(n) > 0 \ and \ \underset{w_1<m<w_2}{\mathrm{argmax}}\ DF(m) = n \\ 0, & otherwise \end{cases} \qquad (5)$$

ここで w_1, w_2 は local maxima の探索範囲である．リズム練習支援システムとしては，(5) 式の $OD(n) = 1$ となる時刻 n を演奏者へのフィードバックとして表示する．

4.2.しきい値設定（キャリブレーション）の方法

発音検出手法の性能について検討する場合，一般的には検出した発音と，別に用意したアノテーションデータを比較して true positive , false positive, false negative をカウントし，F 値を求めて評価基準とする．前記のしきい値設定に関しても，著者らはこれまでアノテーションデータを用いた精度の検証を行い，タンギング奏法およびレガート奏法のいずれにおいても 95%以上の F 値が得られることを確認している [8]．このとき，しきい値の算出に用いるパラメータ δ, λ, α についてはアノテーションデータに対して高い F 値が得られるように設定しており，その自動的な決定方法についても検討を行った [9].

しかし，リズムチューナーでは任意の演奏パターンを入力とするため，それぞれの入力にアノテーションデータを用意することは現実的でない．そこで，アノテーション情報なしでしきい値の算出に用いるパラメータを決定するため，実際の練習を行う前にシステムの使用者に合わせてこのパラメータ設定を行うためのモードを設けることにした．これをキャリブレーションと呼ぶことにする．このキャリブレーションにおいては，システムを使用する演奏者が一定時間の間に任意のいくつかの音を演奏し，その時に自分が発音した音数をシステムに入力する．システムは，入力された音響信号から求めた発音検出数がユーザの指定した発音数とできるだけ近くなるように，しきい値算出に用いるパラメータを設定する．つまり，アノテーションデータなしでのしきい値算出パラメータ設定の問題を，発音検出数を評価関数とする最適値探索問題として考える．

ここで (3) 式において，しきい値算出パラメータである δ, λ, α をすべて設定するのは煩雑である一方，$TH(n)$ を求めるうえでは δ が固定しきい値としておおまかな水準を規定し，λ と α の項が区間 $[n-v_1{:}n-v_2]$ の中央値と平均値を動的しきい値として組み込んでいることから，λ と α のいずれかを用いることでも動的しきい値の効果が得られる．そこで，λ と α のいずれかのみを用いることとして，それぞれを使用した場合の検出率を調べたところ λ を用い

た方が効果的であったため，以降の議論は $\alpha = 0$ とし，δ と λ を対象として行う．

具体的な最適値探索として，山登り法と同様の手法を用いる．まず $\lambda = 0$ と仮定して，2 分法を用いて発音検出数がユーザの指定値に最も近くなる δ を求める．次に，λ を 0～1 の範囲で 20 等分し，その中で発音検出数がユーザの指定した発音数に最も近い λ と α の組を用いることとした．このとき，λ を増加させた分，δ を減少させることとする．

この方法において，最初の δ の選択に 2 分法を用いることができる理由を述べる．(3) 式において λ を固定すれば，δ を増やすと $TH(n)$ が大きくなり，従って (4) 式の $DF(n)$ は小さくなる．これにより (5) 式の $DF(n) > 0$ が成立することが少なくなるため，結果的に $OD(n) = 1$ として検出される発音数が減少することから，発音検出数は δ の増加に対して一様減少となる．このことから 2 分法を用いて発音検出数がユーザの指定した発音数に最も近い δ を求めることができる．また，次の段階で λ を変化させて探索するのは動的しきい値の効果を得るためであり，この探索範囲は，これまでの研究から λ の値は 0～1 程度の値で動作することが経験的にわかっているため設定したものである．

5. 実験

前節で述べたアノテーション情報なしでしきい値算出パラメータを設定するキャリブレーションの妥当性を確認する実験と，そこで設定したパラメータを用いてリズム練習支援システムとしての被験者実験を行った．

5.1. しきい値設定に関する実験

アノテーション情報を用いないしきい値設定について検討するため，しきい値 $TH(n)$ を算出するパラメータと発音検出数の関係について調べた． システムが検出した発音数からユーザが指定した発音数を引いた値を「発音検出数差」と呼ぶことにする．発音検出数差が 0 であれば，システムが検出した発音数とユーザが指定した発音数が等しいからしきい値が適切に設定されているといえる．

実験として，フルート，トランペット，ホルンの 3 種類の楽器で録音した音の発音検出を実施した．録音したパターンは変ロ長調で 1 オクターブ音階をタンギングで演奏したもの (16 音，約 5 秒) である．前節で述べた(3), (4)式のパラメータは $v_1 = 100$ ms, $v_2 = 0$, $w_1 = 50$ ms とした．図 2 にホルンの場合の，しきい値算出パラメータ δ, λ に対する発音検出数差の結果を示す．この図から，前節で述べたとおり λ が一定であれば δ に対して発音検出数差は一様減少していることが確認できる．また，λ を増加させるにつれて発音検出数差が 0 となる位置は δ を減少させることで得られることもこの図からわかる．4.2 節で述べたパラメータ探索は図 2 において，まず $\lambda = 0$ となる奥側のライン上で 2 分法を用いて発音検出数差が 0 に近い点を探し，次にそこから λ を変化させて，それぞれの λ に対して発音検出数差が 0 に近くなる δ を求めて最適値を得ることになる．最初に δ で 2 分法を用いることにより，パラメータ空間全体を探索する必要がなくなるため効率を良くしている．図 2 のような，しきい値算出パラメータに対する発音検出数の関係は他の楽器やレガート奏法の場合でも同様の結果が得られており，しきい値算出パラメータの設定方法が妥当であるといえる．

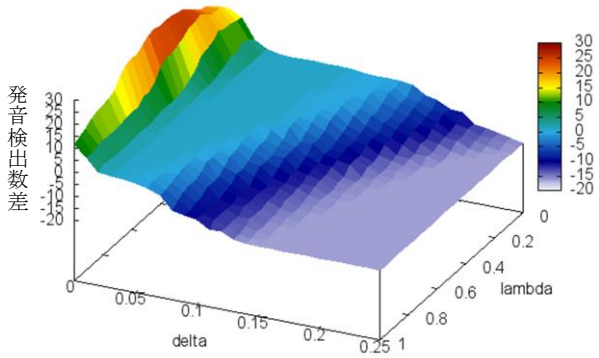


図 2　しきい値算出パラメータに対する発音検出数差

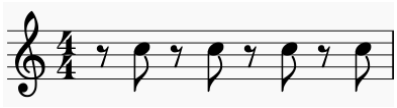


図 3　実験で演奏した楽譜

表 1　被験者実験によるシステム使用前後の発音平均位置と標準偏差

被験者	平均値		標準偏差	
	使用前	使用後	使用前	使用後
A	62.0	55.9	4.0	2.4
B	54.6	52.4	3.0	2.2
C	56.2	46.7	4.1	11.1
D	49.6	49.9	4.9	2.8
E	53.0	50.6	2.8	2.6

（1 拍の長さを 100 としている）

5.2. システムを使用した被験者実験

これまでに述べたしきい値算出パラメータの設定方法によりリズムチューナーを用いて被験者実験を行った．実験で演奏した内容は図 3 で示すように，拍の頭に音がなく裏拍での発音が連続するリズムである．これは「後打ち」または「裏打ち」と呼ばれ，行進曲などで頻繁に使用される一方で，そのリズムを正確に演奏することが難しく，しばしば練習項目としてとりあげられるものである．

被験者は 16 歳～19 歳の 5 人の吹奏楽経験者で，楽器演奏歴は 1～7 年である．まずリズムチューナー使用の前に，被験者ごとにしきい値パラメータの設定を行った．表 1 に，システムを使用開始時と 5 分間のリズムチューナー使用後について，それぞれ 16 拍分演奏したときの発音位置の平均値と標準偏差を示す．演奏テンポは 100 BPM で，16 拍分の音高は同一とし，被験者が演奏しやすい音高で実施した．演奏楽器は被験者 A, B がホルン，被験者 C がトロンボーン，被験者 D がクラリネット，被験者 E がバスーンである．表 1 では 1 拍の長さを 100 と表しているので後打ちの正しい発音位置は 50 となる．表 1 において，被験者 C を除いて，システム使用後は平均値が 50 に近づくとともに標準偏差も小さくなっていることから，正確なタイミングで安定して演奏できるようになっていることがわかる．

被験者 C は，システム使用後の測定で画面表示を意識しすぎて演奏が乱れてしまった様子が実験時に観察された．

図 4 に被験者 A がリズムチューナーを使用したときの画面表示を示す．図 4 (a) はシステムを使って練習する前，(b) はシステムを 5 分間使用した後である．画面右上の「感度設定」と表示されているボタンが，4.2 節で述べたキャリブレーションモードに入るボタンである．このボタンを押すとシステムが 10 秒間のカウントダウンを行うので，その間に演奏者がいくつかの音を演奏し，その発音数をキーボードから入力する．この発音数をもとにしきい値のパラメータ設定を行う．感度設定ボタンの下にはメトロノーム音の有無の切り替えと，テンポおよび拍子を入力するテキストボックスがある．画面左側で 4 段になっているのは，演奏に対する発音検出を表示するもので，1 段が 1 小節に対応する．図 4 では 4 拍分が表示されており，この拍数は左側の Beat テキストボックスで入力した値によって変化する．システムの使用中は，現在時刻を表すバーが左から右に移動して，メトロノームの視覚的役割を果たす．図 4 では，4 小節目 4 拍目にあたる場所にこのバーが表示されている．このバーの移動とともに，発音が検出された位置に赤い点を打っていく．図 4 (b) で 1 小節目 2 拍目と 3 小節目 3 拍目に点がないが，これは演奏としては発音があったがシステムが検出しなかった false negative である．このように部分的な誤検出はあるものの，全体としてシステム使用前にやや遅れ気味だった発音が，システム使用後は改善していることが視覚的に確認できる．参考として，被験者から「視覚情報があることで練習に集中しやすい」などのコメントが得られた．

検出精度に関しては，被験者 5 名全体で発音 160 個に対して false positive と false negative がそれぞれ 5 個ずつであったため，F 値が 96.8%となった．これは，筆者らのこれまでの研究でアノテーションデータを使用してしきい値算出パラメータを設定した場合と同等の水準となっているため，本論文で提案したアノテーションデータを用いないパラメータ設定方法の妥当性を示しているといえる．

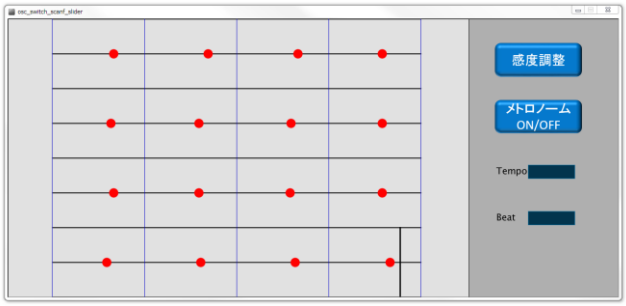
6. まとめ

リアルタイム発音検出を用いたリズム練習支援システムとして，事前のアノテーション情報なしで発音検出パラメータを設定する方法の提案と実装をリズムチューナーとし，その被験者実験を実施した．システムを使用する演奏者の音に合わせたしきい値検出パラメータのキャリブレーションを簡便な方法で実現することで，アノテーション情報がない場合でも，従来のアノテーション情報を用いて実験した場合と同等の検出精度が得られた．これによって，リアルタイムで任意の演奏パターンを扱うリズム練習支援システムとしての実用性が大きく高まったといえる．

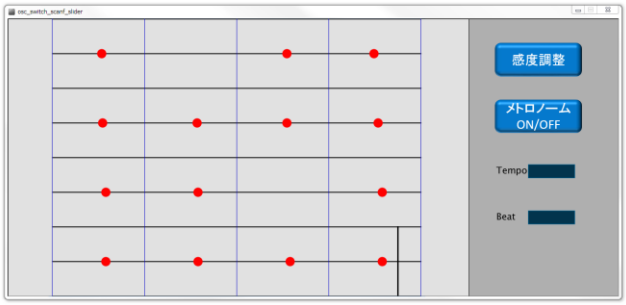
今後の課題として，本論文ではリズムチューナーの使用前後について被験者実験を実施したが，リズム練習支援システムとしての有効性を検証するためには，リズムチューナーを使用しない統制群を置いた実験を実施して対応のない 2 群の平均値検定を行うか，同一人物でリズムチューナーを使用しない状態と使用した状態で実験し対応のある平均値差検定を実施する必要がある．

謝辞

本研究は JSPS 科研費 16K01094 の助成を受けたものです．



(a) システム使用前



(b) システム使用後

図 4 リズムチューナーの実行画面（被験者 A）

参考文献

[1] S. Raptis, A. Chalamandaris, A. Baxevanis, A. Askenfelt, E. Schoonderwaldt, K. Hansen, D. Fober, S. Letz, and Y. Orlarey, “IMUTUS - an Effective Practicing Environment for Music Tuition”, Proceedings of International Computer Music Conference, pp. 383-386, 2005.

[2] G. Tambouratzis, K. Perifanos, I. Vougari, A. Askenfelt, S. Granqvist, K. F. Hansen, Y. Orlarey, D. Fober, and S. Letz, “VEMUS: An integrated platform to support music tuition tasks,” 8th IEEE International Conference on Advanced Learning Technologies, pp. 972-976, 2008.

[3] 竹川佳成, 寺田努, 塚本昌彦, “リズム学習を考慮したピアノ演奏学習支援システムの設計と実装,” 情報処理学会論文誌,第 54 巻, 第 4 号, pp. 1383-1392 , 2013.

[4] Y. Konishi, and M. Miura, “Estimating musical score and proficiency at playing drums,” Proceedings of International Congress on Acoustics, 2010.

[5] J. Wang, S. Wang, W. Chen, K. Chang, and H. Chen, “Real-time pitch training system for violin learners,” IEEE International Conference on Multimedia and Expo Workshopsx, pp. 163-168, 2012.

[6] M. Yamasaki, and M. Miura, “Proficiency estimation for audio of cello performances,” Proceedings of Forum Acusticum, 2014.

[7] M. Konecki, “Self-paced computer aided learning of music instruments,” Proceedings of International Convention on Information and Communication Technology, Electronics and Microelectronics, pp. 910-914, 2015.

[8] Masanao Yamada, Akihiro Matsuo, "Development of rhythm practice supporting system with real-time onset detection," Proceedings. of International Conference on Advances in Electrical Engineering, pp. 176-179, 2015.

[9] 松尾章弘, 土江田織枝, 山田昌尚, “リアルタイム発音検出のための動的しきい値自動最適化,” 情報処理学会第 78 回全国大会, 第 2 分冊, pp. 437-438, 2016.

[10] P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. Sandler, “A tutorial on onset detection in music signals,” IEEE Transactions on Speech and Audio Processing, Vol. 13 No. 5, pp. 1035-1047, 2005.