

指一本によるピアノ演奏システム：sfp

奥平 啓太[†]

片寄 晴弘^{†‡}

[†] 関西学院大学理工学部

[‡] 科学技術振興事業団 さきがけ研究 21

{keita, katayose}@ksc.kwansei.ac.jp, <http://ist.ksc.kwansei.ac.jp/~katayose/>

本論文では、指一本の打鍵動作（MIDI 鍵盤でもパソコンのキーボードでも動作する）で、テンポ・音量など指揮的な演奏表現感覚を楽しむためのトイシステム sfp について述べる。sfp では、プロセス制御等によく利用されている PID 制御の考え方を導入するとともに、演奏プラン（表情）を含んだ演奏データを用いてスケジューリングを行っている。これらに基づき、1）ピアノ演奏あるいは指揮的な演奏モードの選択、2）さまざまな演奏応答性の設定、3）拍以下の単位の微妙な演奏表現、を実現している。本論文では、sfp の概要とデザインについて紹介し、指一本の打鍵操作によるテンポ変化と「間」の弾き分けインタフェースについて述べる。

A Performance System using one Finger: sfp

Keita Okudaira[†]

Haruhiro Katayose^{†‡}

[†]Kwansei Gakuin University

[‡]PRESTO, JST

This paper reports a amusement system: sfp, which enables us to enjoy music performances with one finger tap. The scheduler of sfp is based on the PID controller, and uses score information which contains detail note-level expression. sfp provides 1) mode selection (piano mode or conducting mode), 2) parameter setting toward various performance taste, and 3) delicate note expression within a beat. This paper describes the conceptual basis and the design of sfp, and the interfaces which distinguish tempo change and pause for the inter-onset-interval given by a performer.

1 はじめに

音楽演奏においては、表現を行う前にまず、楽器の演奏法を習得する必要がある。「こんな演奏表現を行って見たい」という意図を持っていたとしても、技術が伴わずできないということも少なくない。実際、小中高等学校の集団教育においても、技術の習得に大部分の時間が割かれ、表現に対する教育は必ずしも充実したものではなかった。この反省から、最近では、表現力や創造性を養成するための授業内容が検討され、計算機システムの利用も始まっている。我々は、単純な打鍵動作で、テンポ・音量など指揮的な演奏表現感覚を楽しむためのアミューズメントシステム TFP を開発し、音楽教育¹、や演奏分析研究に利用してきた [1]。TFP は、指一本の拍打で、テンポ、全体的な音量を制御し、「間」を挿入したい場合には、別の鍵盤を押さえる時間で表現するように設計されていた。TFP 自体は単純で使い易い

点も多かったが、メロディーの音符列に対応した拍打を行うことができず、「間」の表現を行う際に、2本の指を使わないといけななどの制約があった。本論文では、PID 制御の考え方を導入し、演奏上のインタフェースを根本的に向上させた演奏システム sfp について述べる。以下、システムの概要（第2章）、PID 制御に基づくスケジューリング（第3章）、演奏表現のインターフェイス（第4章）について述べ、第5章で検討を行う。

2 システムの概要

sfp は、指一本の打鍵操作による演奏インタフェース、PID 制御に基づく演奏スケジューリング、演奏プランの利用、を特徴とする演奏支援システムである。

具体的には、1）単純打鍵操作によるテンポ、音量；「間」に関する演奏表現、2）演奏データの利用による拍以下の微細な表現、3）メロディーのビートを刻むことでの伴奏付き演奏、等が可能である。図1にシステムの概略を示す。演奏データには楽譜情報の他、各音の打鍵・離鍵、音量の

¹ もちろん、楽器を弾いたり、描画技術を高めることも、芸術の大きな楽しみの一つである。その教育的な効果についても否定する立場にはない。情報処理技術により、多様な表現技術に触れることが可能になるのであれば、教育の一選択として利用すべきという立場である。

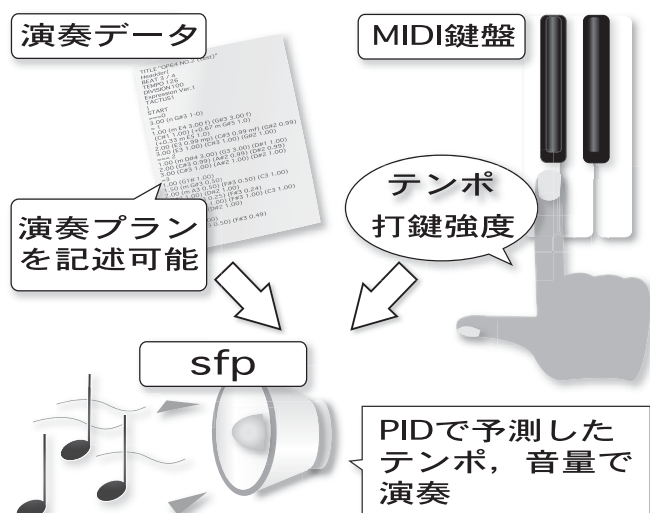


図 1: sfp の概要

```
Op64.txt
TITLE "Op.64" /* "楽曲名" */
Header{ /* 演奏開始時の各種情報 */
  BEAT 3 / 4 /* 拍子 */
  TEMPO 126 /* 四分音符/minのテンポ */
  DIVISION 100 /* 拍子の分母あたりの分解能 */
  TACTUS 1 /* 1拍あたりの打鍵回数 */
}
START /* ここから音符情報 */
=0
3-000 (m G#3 1-000 mp)
=1
1-000 (m E4 3-000 mf) (G#3 3-000) (C#1 1-000) (G#5 1-000 p +0-002)
(E5 1-000 p +0-004 %0-003)
2-000 (E3 0-099) (C#3 0-099) (G#2 0-099)
3-000 (E3 1-000) (C#3 1-000) (G#2 1-000)
=2
Body{ /* 2小節目からの各種情報 */
  BEAT 3 / 4 /* 拍子 (変わらない場合は省略可) */
  TEMPO 126 /* 四分音符/minのテンポ */
  TACTUS 2
}
1-000 (m D#4 3-000 mf %0-010) (G3 3-000) (D#1 1-000)
2-000 (C#3 0-099) (A#2 0-099) (D#2 0-099)
3-000 (C#3 1-000) (A#2 1-000 v63) (D#2 1-000 v60)
=3
1-000 (m G1# 1-000 mf)

/***** 中略 *****/

=32
1-000 (G#3 2-000) (m C#3 1-000) (C#2 1-000)
END /* ここまで */
```

図 2: 楽譜として与えるデータ例

微妙な表現を記載できるようにしている．演奏データを用いることで，演奏者は既存の演奏を下敷きとして用いることができる．

PID 制御に基づく予測的なスケジューリングを行っており，追従性の調整，自由な拍打，遅れ補償ができるようになっている．なお，ペダルの操作に関しては，スルーで演奏に反映するようにしている．

以下，システムを理解する上で中心的な概念となるテンポと拍打，演奏データについて説明する．

2.1 テンポと拍打

1. テンポ

テンポは楽曲を演奏する際の速度を意味しており，1分あたりの単位音符の演奏する数 (bpm) を用いて表される．単位音符には，4分音符が割り当てられることが多いが，リズムの表現にあわせ，他の音符が用いられることもある．

2. タクトスと拍打

タクトスは，指揮での腕の動き²によって計算される時間のことである．ここでは，規準拍内に打鍵する回数をタクトスと定義している．例えば，規準拍が4分音符の場合，タクトスを1とすることで，4分音符一つに一回の拍打を対応させることになる．4つ並んだ16分音符すべてに拍打を対応させたい場合はタクトスを4と設定する．

sfp では，以上のように定義されたテンポ，タクトスに基づいて拍打に関する処理を行っている．さらに，タクトスに関しては，倍の刻み（裏拍打ち）を解釈するような機

構を組み込んでおり，細かい演奏制御を行う際に，利用できるようにしている．

2.2 演奏データ

楽譜として与えるデータを図2に示す．Headerでは楽曲の全体的な情報（初期状態）を記述している．BEAT, TEMPO, DIVISION, TACTUSはそれぞれ，楽曲の拍子，テンポ，規準拍あたりの分解能，規準拍あたりの打鍵回数を表している．これらの情報については，Bodyとして，楽譜データの任意の場所に埋め込めるようにしている．これにより，変拍子，部分的なタクトスの変更にも対応できるようにしている．

sfpの基本的な使い方では，テンポ，音量については演奏者が与えるようにしている．音符に関する基本的な情報は，ピッチ（キー）と，拍単位で記述される発音時刻と音価（音符の長さ）である．これに加え，deviation項として，音量（velocity値），発音時刻と音価のズレを記述できるようにしている．実際の演奏例あるいは音楽解釈機構を用いて演奏プランを用意することで，単純な打鍵動作だけでは表現できない拍単位の演奏ニュアンスを付加することが可能となる．

3 PID制御に基づくスケジューリング

sfpでは，プロセス制御によく用いられるPID制御を用いて，演奏のスケジューリングを行っている．この章では，PID制御，演奏スケジューリングについて述べる．

² 指揮においては，拍の刻みの粒度はダイナミックに変えられている．

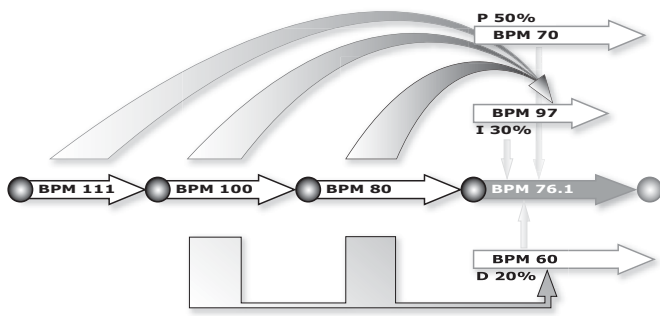


図 3: PID 制御によるテンポの計算

3.1 PID 制御

PID 制御 [2] において、P 値（比例動作）は、制御対象の目標値と現在の値との偏差に基づいて操作量を決定する。偏差が大きければ、制御対象を目標値に早く近付けるために操作量を大きくし、偏差が小さければ、この状態を維持しようと小さくする。I 値（積分動作）は、偏差の積分値を反映し、定常偏差を改善することにより、収束をはかる。D 値（微分動作）は、偏差の変化量に基づいて、偏差の動向を予測するような制御を行い、未来の予測より動作の収束時間を改善する。この 3 つの要素にそれぞれ適当な重みを配分し、操作量を決定することによって、制御対象の現状を目標に、より速く近づけていくのが PID 制御の狙いである。

3.2 sfp における PID 制御

sfp での PID 制御は、人間とシステムのインタラクションを扱う点において特殊であり、厳密には上記の定義とは異なるが、基本的な操作イメージは同じものとして理解できる。ここで、1) P 値とは、deviation 項を含む演奏データ（システムが弾きたい演奏）、2) I 値は、演奏者の、指定タクト分の演奏データ履歴（テンポ、音量）平均、3) D 値は、演奏者の、現在の拍と、その直前の拍の演奏データの変化量から算出される予測値、に対応する。以下、テンポ制御を例に、PID 制御に基づくスケジューリングについて説明する。

図 3 は、演奏者の過去 3 拍のテンポから、次拍のテンポを予測する例である。P 値は、楽譜に記述された BPM³ であり、I 値は、過去 3 拍の BPM の平均値である。D 値は、過去 2 拍の BPM の変化量から計算される次拍の BPM の予測値である。図 3 の例では、演奏者が徐々にテンポを遅くしながら演奏しているので、次拍はさらに遅いテンポで演奏すると予測される。これら 3 つの値に重みを割り振ることで、システムが出力する演奏のテンポを決定する。

³ BPM はテンポの表現法で、1 分間あたりのビート数である。図 1 の矢印の長さはその逆数として表されている。

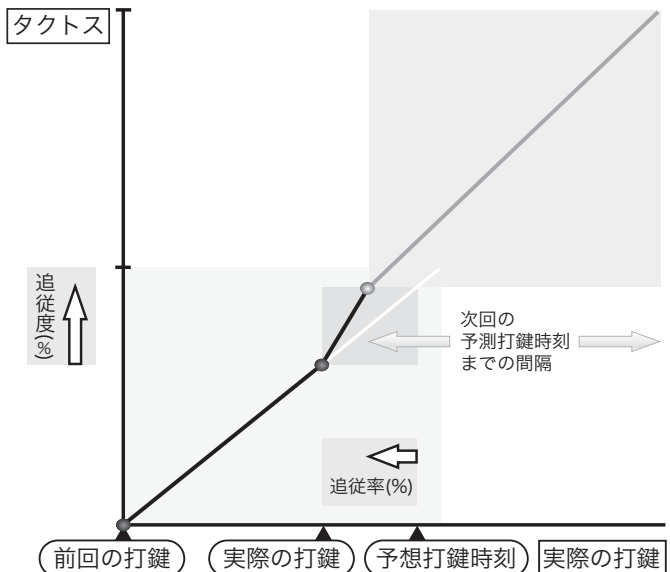


図 4: スケジューリングの様子

P 値の割合を大きくとれば、演奏データを重視した演奏を行うので、ピアニストを指揮しているのに近い演奏感覚が得られる。I 値を重視すれば、自身の打鍵操作にゆっくりと追従する演奏感覚、D 値を重視すれば、局所的な変化によく反応する操作感が得られる。

3.3 スケジューリングの実際

システム実行時には、上記の PID 制御の重みに関するパラメータの他に、I 制御（平均）を求める際に参照するタクトス値、追従性に関わるパラメータ（以下）を与える。

3.3.1 追従度と追従率

図 4 にスケジューリングの様子を示す。ここで、横軸は実時間、縦軸はタクトスの区切りである。傾きはテンポをあらわし、傾きが大きいほどテンポは速くなる。

sfp では、拍打の検出直後に、PID 制御を用いて次のタクトスに割り当てられた音符の発音（消音も含む）までのスケジューリングを行う。この処理だけでは、システムの発音（予想打鍵時刻）と演奏者による実際の拍打時刻がずれてしまうことがあるため、その修正を行う処理を実装している。ここでは、ズレの修正の早さに関連したパラメータとして、追従度と追従率を与えるようにしている。追従率は、予想打鍵時刻前に演奏者が打鍵した際に、どの程度打鍵した拍まで追いついて演奏するかを示す割合である。追従度は、追従度分の演奏幅が、どれだけの割合で追いつくかを示した割合で、追従率、追従度とも、百分率で表している。

追従率を 100 に設定すると、予想打鍵時刻より先に打鍵が行われた場合に、即時にその拍打に対応した音符の発消

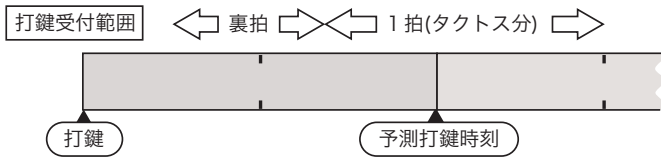


図 5: 打鍵受付範囲

音がなされ、TFP のスケジューラとほぼ同様の動作をすることになる。

3.3.2 拍打の検出

拍打の検出の様子を図 5 に示す。拍打は、PID 制御により計算されるテンポ（予想される一拍の長さ）を規準として検出がなされる。その間の $1/4$ の時刻までは、演奏ミスや他の鍵盤に指が触れた場合などの事態を想定し、拍打を受け付けられないものとする。 $1/4$ から $2/3$ までは、裏拍打ちがなされたものとしての拍打を検出する。 $2/3$ から $3/2$ までは、タクトスに対応する拍打がなされたものとして検出を行う。このように打鍵受付範囲を設けることで、最大で、テンポを 3 分の 2 まで遅くすることや 2 倍にまで速くすることが可能となっている。

3.3.3 テンポの計算

以上のような拍打に関する定義に基づき、以下のようにテンポの計算を行う。

P 値, I 値, D 値それぞれの重みを E_P, E_I, E_D 。また、演奏データのテンポを $BPM_P[\text{bpm}]$, I (平均) として参照する小節数を α とすると、 n 回目の打鍵間隔 $IOI_n[\text{s}]$ は、

$$IOI_n = \left\{ E_P \frac{60}{BPM_P} + E_I \sum_{k=n-\alpha}^{n-1} IOI_k + E_D (2IOI_{n-1} - IOI_{n-2}) \right\} \times \frac{1}{E_P + E_I + E_D}$$

である。そして、 $n-1$ 回目の打鍵後のタクトス間の演奏が全て終了している場合、 n 回目の打鍵後のテンポは、これを $BPM_n[\text{bpm}]$ とすると、

$$BPM_n = \frac{1}{IOI_n}$$

である。

演奏中の拍が全て演奏される前に次の拍が打鍵された場合、追従度を AL , 追従率を AR とすると、 $n-1$ 回目の打鍵後の演奏分の残りの間隔の n 回後の打鍵直後の追従度で定められた分のテンポ $BPM_{An}[\text{bpm}]$ は、追従率が 100 より小さい場合、

$$BPM_{An} = \frac{AL}{100 - AR} BPM_{n-1}$$

となる。

演奏中の拍が全て演奏される前に次の拍が打鍵された場合、 $n-1$ 回目の打鍵後の演奏分の残りの間隔を $IOI_{odd}[\text{s}]$, 追従度を AL , 追従率を AR とすると、 $n-1$ 回目の打鍵後の演奏分の残りの間隔の n 回後の打鍵直後の追従度で定められた分のテンポ $BPM_{An}[\text{bpm}]$ は、追従率が 100 より小さい場合、

$$BPM_{An} = \frac{AL}{100 - AR} BPM_{n-1}$$

となる。

4 演奏表現のインターフェイス

sfp では、PID の各要素の重みや追従性に関するパラメータを変えることによって、さまざまな応答感を得ることができる。加えて、一本の指の打鍵操作での「間」の表現、予測に基づく自走演奏、などの演奏インタフェースを提供している。

4.1 「間」の表現

「間」とは、次の拍の先頭を演奏する前に、ためる（時間的間隔をあける）ことを言う。「間」が挿入された際には、それを「間」と判断し、テンポ計算から除外しなければ、次拍のスケジューリングが大きく乱れてしまう。sfp では、図 5 に示した「打鍵受付範囲」、「予測打鍵時刻」を利用した次の 2 種類のタイプの「間」の挿入インタフェースを選べるようにしている。

鍵盤 off で間タイプ 打鍵受付範囲を超えても次の打鍵が確認されなかった場合に、これを「間」と判断する。システムは演奏を次の拍の先頭の直前で停止した状態となり、次の打鍵を確認すると同時に演奏を再開する。

鍵盤 on で間タイプ 鍵盤を離すタイミングを積極的に利用するタイプである。鍵盤を押し込んだままで、PID 制御で予測された次の打鍵時刻である「予測打鍵時刻」を過ぎた場合に、これを「間」と判断する。システムは演奏を次の拍の先頭の直前で停止した状態となり、鍵盤をはなして次の打鍵を確認すると同時に演奏を再開する。

システムが「間」を判断した場合、その拍の打鍵間隔は、PID 制御での I 値, D 値を算出する際の参照リストから除外される。裏拍での「間」を表現は認めておらず、タクトスで指定した区間の最後まで演奏がなされる⁴。

⁴ これらの操作により、タクトスの値をうまく設定することで、思わぬところで「間」が入ってしまうといったミスを防ぐことができる。

表 1: 各種モードの変数設定

モード	出だしの演奏	鍵盤 on で間	鍵盤 off で間
ピアノ	○	○	×
バトン	×	×	○
伴奏	○	×	×

4.2 自走演奏

PID 制御を用いた予測スケジューリングのメリットとして、打鍵を待たずに演奏を進める自走演奏への応用が有る。デフォルトでは自走区間を 1 小節に設定している⁵が、任意の小節数の自走演奏が設定できるようになっている。

鍵盤 off で間タイプ 鍵盤を押し放しにした状態で予測打鍵時刻を迎えた時には、システムはこの予測打鍵時刻丁度に打鍵がなされたものとみなして、以降の演奏を続ける。このまま演奏者が、鍵盤を押し放しの状態を続けた場合、演奏者が設定した小節数、自走演奏が行われる。

鍵盤 on で間タイプ 予想打鍵時刻を過ぎても演奏者が次の打鍵をしない場合、自走演奏が求められているのか、あるいは、テンポを遅くすることが求められているかをシステムが判断する必要がある。このような時には、次の拍の冒頭からタクトスの半分までテンポを維持し、“出だしの演奏”として演奏するようにした。もし、この拍の打鍵が予想打鍵時刻より遅い場合には、出だしの演奏分をシステムが“かぶって”演奏したものとみなし、打鍵後は新たに計算されたテンポに基づいてスケジューリングが行なう。演奏者が打鍵受付範囲をこえても打鍵を行わなかった場合は、自走演奏が求めたと判断し、予測打鍵時刻丁度に（演奏者の）打鍵があったものとして処理し、引き続き次の拍まで演奏を行うようにしている。

4.3 各種モード

上記のようなテンポ表現と‘間’の表現に関わるインタフェースを（従来楽器の）奏法感覚の視点でまとめると、表 1 に示すようなモードに整理される。

ピアノ モード sfp の基本となるもので、普段ピアノ演奏をしている人に一番自然に感じてもらうことのできたモードである。このモードでは、基本的にタクトスごとに打鍵することによって演奏を進めていく。演奏者が定めた小節数は打鍵なしに演奏を進めることができる。

バトン モード 指揮経験のある人や、打楽器の演奏を主に行う人に自然と感じてもらうことのできたモードである。

⁵ ‘間’は通常、小節の終わり（あるいは小節の始め）に挿入される。‘間’に関する操作を行ってもほとんどの場合、問題なく動作する。小節途中で‘間’を入れたい特殊なケースでは、自走モードをオフにする。

ピアノモードとは逆で、鍵盤を押しっぱなしにしている間は、演奏者の定めた小節数までどんどん演奏が行なわれ、逆に鍵盤をから指を離すと、タクトス分の演奏を終了して演奏が停止する。

伴奏 モード 演奏データに伴奏に相当するデータを与え、メロディのビートに対応したタクトスを指定しておけばメロディーを弾くことで、伴奏が生成される。このモードでは、楽曲のメロディーに対応して打鍵していくので、鍵盤を弾くように演奏した場合や、押し放しにした場合でも演奏者の設定した小節数の間は自動演奏が行われる。

5 検討

sfp は、インタラクションの要素⁶を持った演奏支援システムである。システムの評価を行うことは重要なことであるが、認識率や処理速度といった客観的な指標で評価することは難しい。ここでは、定性的な評価を示すことにする。

5.1 sfp の使用感

sfp を使ってみた人達の多くの反応は、まずは、「笑う」ということだった。指一本での演奏システムに対する驚き、予想以上の表現性、操作性に対するちょっとした戸惑いが笑いとなって現れるようである。そして、多くの人が再度、演奏を試みようとした。

ほとんどの人が演奏を繰り返すうちに、こつを修得し、より上手な演奏ができるようになった。音楽経験者の方が、上達度が高く、指揮経験者の場合、数回の操作で、情緒豊かな表現を行えるようになった。sfp は演奏支援システムではあるが、技能の上達を楽しむことができるという点で、楽器としての特性を持っているといえる。

各種パラメータの設定変更に応じた操作感の変化は、使用したものの全員が感じることができた。嗜好するパラメータ設定の組合せは人によって異なる。筆者（片寄）の場合、鍵盤 on で間タイプを用いて、指一本の打鍵で‘間’とテンポ変化の弾き分け表現が出来た時には不思議な喜びがあった。これらの状況については、<http://ist.ksc.kwansei.ac.jp/~katayose/sfp/> に所収しているので参照されたい。

5.2 応用性

TFP は、高校での演奏表現教育、演奏ルールの学習システムのためのデータ入力ツールとして利用されてきた実績が有る。sfp は、パラメータの設定によっては、TFP と同じ動作をするようになる。‘間’の表現機構は異なるが、上記の用途に対し、TFP に替わるシステムとして使用するこ

⁶ ここでは、相互作用、自律性、感応性を考える。

とが出来る．さらに，sfp は，PID 制御に基づく予測，裏拍打ちを許すことで，TFP と比べて，指揮に近い操作感が得られるようになっている．初期的な指揮感覚の養成システムとしての使用が想定される．

sfp には，メロディ入力に対する簡易自動伴奏システムとしての使用法がある．PID 制御により，予測を伴って演奏のスケジューリングを行っているため，遅れ時間を保証してやることで，発音より先だって打鍵データを送信してやる必要の有る（アナログ型）自動ピアノや自然楽器によるメロディ入力⁷に対しても対応できる．

これらの用途とは若干異なるが，ハンディキャップトパーソンの自己表現のためのインタフェース，ボケ防止のアミューズメントなど，福祉的な用途も広がっている．

5.3 類似システムとの比較

sfp に最も類似するシステムは TFP である．上述のように，sfp は，機能面では，ほぼ，TFP をほぼ包含し，スケジューリングとコントロール性，‘間’への対処において，改良がなされている．指一本で，演奏を入力するシステムとしては，他に，CiP [3] がある．CiP は，単旋律の演奏表現の教育を目的にしたシステムで，まずは，ピッチ入力を行い，次にリズムに留意し，メロディビートを任意のキーで打鍵するという修正するというシンプルなインタフェースで構成されている．

機能面では，指揮システム [4, 5]，自動伴奏システム [6, 7] が関連研究となる．

宇佐ら [4] は，実際の指揮の感覚を再現することを目指し，加速度センサによる指揮棒の振りを解析し，HMM によるビートパターンの発見など精緻な作り込みを行った．ブラボーミュージックシリーズ [5] は，PlayStation2 上で動作する指揮者の感覚を楽しむゲームである．こちらは，ボタンを押す際の強弱で曲自体の音量，そのタイミングでテンポを操作し演奏を進めていく．

自動伴奏システムは，演奏者がどの部分の演奏を行っているかを監視し，それに基づき，伴奏データのスケジューリングを行う演奏システムである．Dannenberg は，自動伴奏システムで解決すべき事項を整理し，演奏者のミスに対応するための手法を示した [6]．堀内らは，システムが演奏プランを持つことの重要性，追従性と使用感の関係などについて論じた [7]．

sfp の制御対象は，指揮システムや自動伴奏システムと基本的に同じである．sfp では，PID 制御に基づく操作感の設定，一本指の打鍵による‘間’への対応，演奏データを利用したニュアンスの表現など，演奏インタフェースとしての練り込みを行っており，これらが，他の研究例との差異となっている．

⁷ ピッチ情報を MIDI データに変換して使用する．その際，若干の遅れ時間が発生するので時間補償が必要である．

6 おわりに

本論文では，指一本の打鍵動作（MIDI 鍵盤でもパソコンのキーボードでも動作する）で，テンポ・音量など指揮的な演奏表現感覚を楽しむためのトイシステム sfp について述べてきた．sfp では，プロセス制御等でよく利用されている PID 制御の考え方を導入し，さらに，演奏プラン（表情）を含んだ演奏データを用いてスケジューリングを行っている．これらに基づき，1）さまざまな演奏応答性の設定，2）ピアノ演奏あるいは指揮に近い演奏モードの切り分け，3）拍以下の微妙な演奏表現，を実現している．

指一本の打鍵操作による‘間’が挿入や，裏拍打ち検出などにより，sfp の前身となる TFP と比べ，はるかに自由度の高い演奏が演奏インタフェースを提供できるようになった．

今後は，使用感に関するさらなる調査，演奏モーフィング機能の付加と演奏プラン付演奏データの収集，GUI の整備を進めていく予定である．

謝辞

本研究は，科学技術振興事業団さがけ研究 2-1 の支援を受け実施されました．研究に協力頂きました竹内好宏氏，野池賢二氏，橋田光代氏に感謝します．

参考文献

- [1] 片寄晴弘，竹内好宏，上符裕一，井口征士：TFP の改良と教育利用における評価，情報処理学会音楽情報処理科学研究報告 96-MUS-16，pp.21-25 (1996)
- [2] 須田信英：PID 制御 -システム制御情報ライブラリ-，朝倉書店 (1992)
- [3] 大島千佳，宮川洋平，西本一志：Coloring-in Piano：表情付けに専念できるピアノの提案，情報処理学会音楽情報処理科学研究報告 2001-MUS-42，pp.69-74 (2001)
- [4] 宇佐聡史，持田康典：HMM とファジィを使った指揮認識システム，情報処理学会音楽情報処理科学研究報告 97-MUS-21，pp.37-44 (1997)
- [5] 株式会社ソニー・コンピュータエンタテインメント：ブラボーミュージック，<http://www.scei.co.jp/> (2001)
- [6] R.B.Danneng：An On-Line Algorithm for Real-Time Accompaniment，*Proc. of ICMC*，pp.193-198 (1984)
- [7] 堀内靖雄，田中穂積：自主性を持つ伴奏システム，人工知能学会論文誌，Vol. 10，No. 1，pp.72-79 (1995)