

Spinner : 再構成可能なユーザーインタフェースへの シンプルなアプローチ

小林茂 赤松正行

IAMAS (岐阜県立国際情報科学芸術アカデミー)

概要

本研究では、シンプルなアプローチによる再構成可能なユーザーインタフェースである「Spinner」に関して報告する。Spinner は、ダイヤル型の物理コントローラと仮想コントローラから構成される。1つの物理コントローラはPCの画面上の1つの仮想コントローラに対応し、ユーザは仮想コントローラを操作する際、動的に関連付けを変更する(ある物理コントローラを別の仮想コントローラに関連付ける)ことができる。Spinnerは多目的に使えるため、仮想コントローラと物理コントローラの間を必要とする様々な用途に応用できる。このシステムを用いることにより、ユーザは再構成可能なユーザーインタフェースを作ることができる。

Spinner: A Simple Approach to Reconfigurable User Interfaces

Shigeru Kobayashi and Masayuki Akamatsu

IAMAS (International Academy of Media Arts and Sciences)

Abstract

This paper reports on our recent development of a reconfigurable user interface. We created a system that consists of a dial type of controller 'Spinner,' and the virtual controller objects. One physical controller corresponds to one virtual controller on a PC's display device, and a user can freely change the connection on the fly (i.e. associate the physical controller to another virtual controller). 'Spinner' is versatile and can be used for many purposes that need connections between virtual controllers and physical controllers. By using this system, a user can build a reconfigurable user interface.

1 はじめに

PCで動作する音楽ソフトウェアの多くは、多数の仮想コントローラ(例:ダイヤル、スライダ、スイッチなど)を備えている。図1は代表的なソフトウェア音源の1つであるReasonの画面である[1]。この画面上には、ダイヤル、スライダ、スイッチなど数十個の仮想コントローラが配置されている。

通常は、こうしたコントローラをマウスで操作するが、その操作感は必ずしも仮想コントローラから連想されるものと一致しない。特にダイヤル型のコントローラの場合には、マウス操作(左右または上下)と実際のダイヤルの動き(回転)が一致しないためコントロールが難しい。また、マウスの場合には同時に1つの仮想



図 1 Reason の画面例

コントローラしかコントロールできない。

これらの問題を解決するために、多くの楽器メーカーから様々な物理コントローラが発売されている。こうしたコントローラを用いることにより、仮想コントローラを実物のコントローラに近い感覚で操作することが可能になる。

しかしながら、一般的には、こうした物理コントローラの数、画面上の仮想コントローラの数よりかなり少ない。なぜなら、画面上の仮想コントローラは、ソフトウェア的に簡単に追加できるのに対して、物理コントローラは設置場所や製品コストの影響を受けるため、一定の個数にとどまる傾向にあるからである。

この問題を解決するために、多くの製品では、限られた数の物理コントローラを仮想コントローラに関連付ける仕組みを備えている。例えば、物理コントローラの数 が 16 個の場合、16 個までの仮想コントローラとの関連付けを 1 つのセットとし、複数のセットを切り替えできるといったような仕組みである。

しかし、このアプローチにも問題がある。通常、物理コントローラの各コントローラの配置は固定されているため、画面上での配置と、物理的な実際の空間での配置が一致しないものになってしまう。これは、ユーザが慣れることによってある程度解決するが、そのためには一定の時間が必要である。

これに対して、関連付けたパラメータを書き込むことができるようなテンプレートを用意する、それぞれのパラメータごとに小さな LCD を用意してパラメータ名を表示する、といったアプローチがとられているが、いずれも完全な解決には至っていない。

2 関連研究

こうした問題を解決するための 1 つのアプローチは、物理コントローラを再構成可能にすることである。例えば、Buxton らはタブレットとテンプレートを用いた再構成可能なインタフェースを提案している [2]。また、Greenberg らの Phidgets は、カスタマイズしたコントローラを比較的簡単に作れるようにするための環境を提案している [3]。最近の Villar らによる Pin&Play&Perform も、再構成可能なコントローラの提案である [4]。本研究も、シンプルなアプローチによる再生構成可能なインタフェースを提案するものである。

3 本研究でのアプローチ

本研究でのアプローチを簡単に説明すると、次のようになる。

- 物理コントローラは同時にコントロール可能な数（ダイヤル型であれば片手でコントロールできるのは 1 つであるので合計 2 個）だけを用意する
- 仮想コントローラ側は、ユーザが自由に GUI を設計できるような既存の環境を用いる（本研究ではコンピュータ音楽の分野で一般的に用いられている Cycling '74 社の Max/MSP を用いた [5]）

- 事前に静的な関連付けは行わず、仮想コントローラを操作する時に動的に関連付けを行う

物理コントローラの数を制限したのは、システムをシンプルにして低コストで実現できるようにするためである。次に、GUI 構築部分に独自の新規の環境を用いなかったのは、多くのユーザーが既に慣れ親しんでいる環境に、最小限のライブラリを追加するだけで利用できる方が、新規の環境を新しく覚えるよりも良いと考えたためである。最後の点は、最初の点とも関連するが、動的に関連付けを行うことにより、ユーザが関連付けを覚えておく必要がなくなるようにするという狙いである。

実際の操作手順は次のようになる。

1. 操作したいパラメータの仮想コントローラの上に物理コントローラを置く
2. 物理コントローラのノブを押し込んで関連付けを開始する
3. 全ての仮想コントローラが ID 表示モードになり、白と黒の点滅でそれぞれの ID を表示する
4. 物理コントローラ底部のセンサが画面上的パターンを読み取り、ID をデコードする
5. ID をデコードできたら、以降はその ID でパラメータを送信する

ユーザは、新しいパラメータにアクセスするごとに、一連の流れを繰り返す。実際には、これらの一連の操作は 300ms 程度で終了するため、ユーザーはほとんど待ち時間を意識することなく操作を続けることができる。

図 2 は実際に Spinner を使用している様子である。2 つの物理コントローラ（ダイヤル）はインタフェース・ボードに接続され、インタフェース・ボードは USB 経由で PC に接続されている。ユーザは PC の画面上的仮想コントローラに対して、ダイヤル型のコントローラを介してアクセスする。

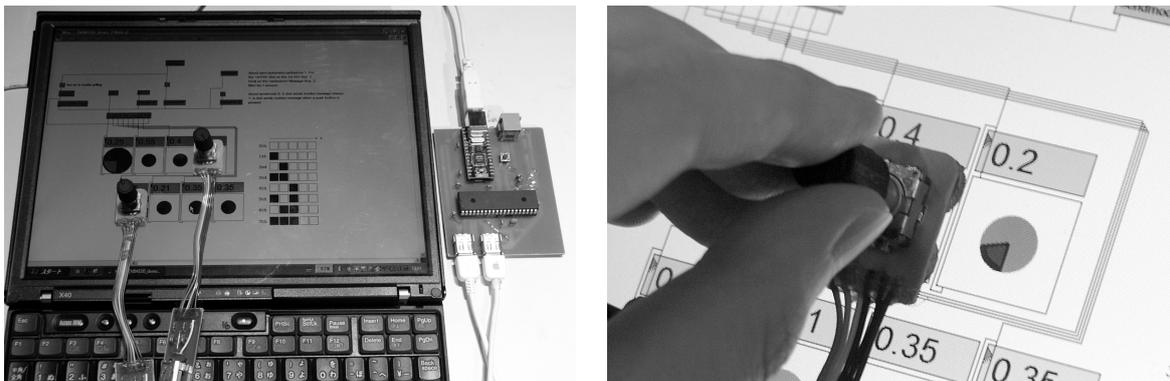


図 2 実際の Spinner の様子

4 実装

4.1 物理コントローラの構成

図 3 は物理コントローラとインタフェース・ボードの構成を示すものである。物理コントローラは、ロータリ・エンコーダ（アルプス電子製）とフォト IC センサ（浜松ホトニクス製）から構成されている。ロータリ・エンコーダには、回転の検出以外に、ノブを押し込むことによって操作できるスイッチが内蔵されている。フォト IC センサは、光センサとアンプを内蔵しており、負荷抵抗の追加により、画面上的の明るさの変化を電圧の変化として簡単に取り出すことができる。ロータリ・エンコーダ、スイッチ、フォト IC センサからのアナログ情報は、インタフェース・ボード上の PIC マイコン（Microchip 社製）で処理する。PIC マイコ

ンは、USB シリアル変換モジュール（マイクロテカ社製）を経由して PC に接続され、データやコマンドの送受信を行う。PC のディスプレイは通常の LCD であるが、物理コントローラを画面に押し付けることによる損傷を防ぐため、約 2mm 厚のアクリル板を画面の大きさに合わせて加工したものを被せた。

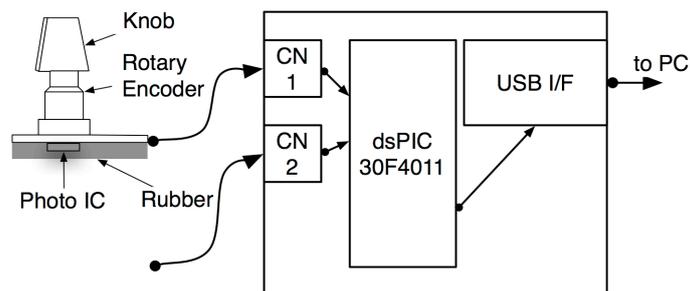


図 3 物理コントローラの構成

4.2 PC 上の GUI

図 4 に PC 上の GUI の例を示す。GUI は、コンピュータ音楽の分野で広く用いられている Max/MSP 環境上で実装されているため、ユーザは必要に応じて簡単にカスタマイズすることができる。既存のものを元にカスタマイズを加えることも可能であるし、新規に作成することも可能である。

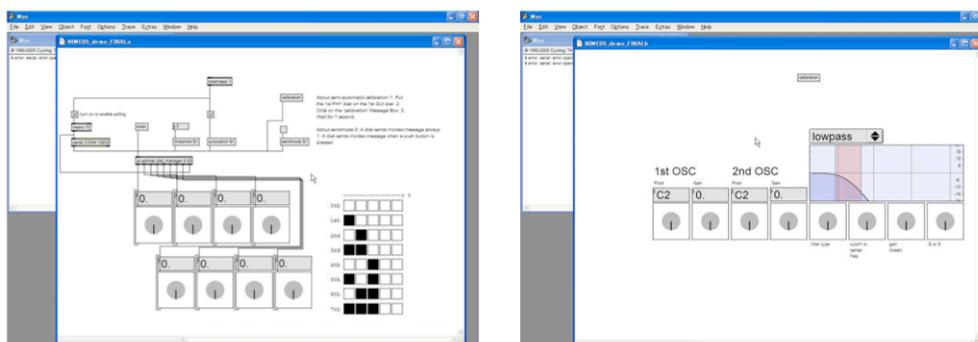


図 4 実際の GUI の例

Spinner に関連した部分（マイコンとのシリアル通信を制御する部分とダイヤル型の GUI を構成する部分）は、Javascript で記述され、Max/MSP からは js オブジェクトとして簡単に利用することができる（図 5）。なお、ダイヤル型の GUI を構成する部分も Javascript で記述されているため、形状や振る舞いなどをカスタマイズすることも容易である。

4.3 認識と関連づけ

ユーザが物理コントローラのノブを押し込むと、それぞれの仮想コントローラは白と黒の点滅により、ID を表示する。点滅と同時に、PC 側からマイコン側にタイミングを通知するためのメッセージが送信される。マイコン側は、フォト IC センサからの明るさ情報と、PC 側からのタイミング情報を組み合わせることにより、BCD エンコードされた点滅パターンをデコードする（図 6）。この図で、もともと仮想コントローラが表示されていた画面上の領域が白く塗りつぶされている場合は 0 を表し、黒く塗りつぶされている場合は 1 を表す。例えば、5 番目のダイヤルであれば、黒 白 黒（すなわち二進数で 101）という順序で点滅する。

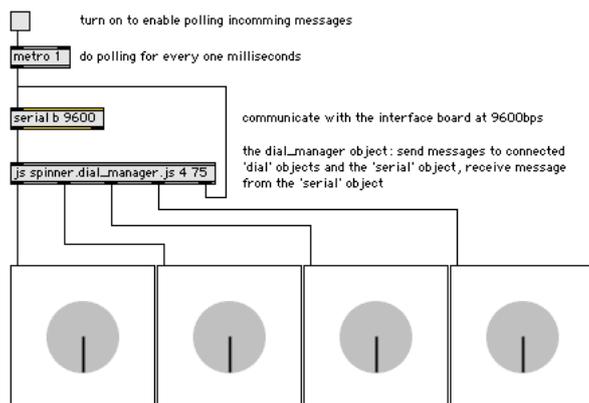


図 5 spinner.dial_manager オブジェクトと spinner.dial オブジェクト

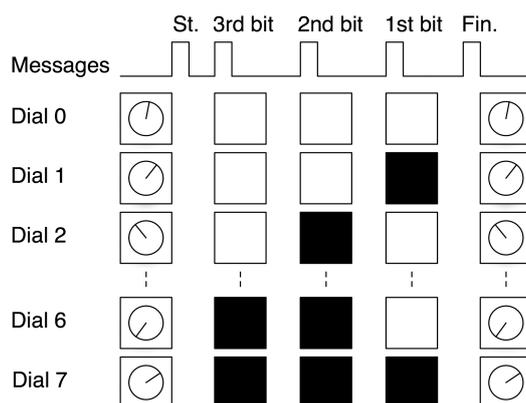


図 6 仮想コントローラで提示される認識パターンとそれぞれの ID の関係および PC からインタフェース・ボードに送信されるタイミング情報

5 結果と考察

5.1 認識時間に関して

認識時間に関しては、理想的には PC の一般的なフレーム・レートである 60fps (すなわち 1 回あたり約 16.6ms) 単位で点滅パターンを提示したかったが、実際にはこのようなレートでの提示は実現しなかった。Max/MSP では、画面表示とその他のコントロールのタイミングは非同期であるため、画面のリフレッシュ・タイミングに合わせて正確に制御することがそもそも難しい。

このため、PC 側の負荷によるタイミングのずれも吸収できるようなマージンも含め、約 75ms を点滅パターン提示の単位とした。画面上に 8 個の仮想コントローラがある場合、1 回の関連付けに必要とする時間は約 220ms である。これは、GUI 操作における典型的な反応時間とほぼ同じであるため [6]、この時間であれば操作に関してはほとんどストレスがないものと思われる。実際に、8 名のユーザによるテストを行ったが、関連付けに必要とする時間に関して、遅いというフィードバックはなかった。

しかし、仮想コントローラの数が増えすぎると、その分だけ関連付けに必要とする時間も長くなるため、今後はできる限りこの時間を短縮していきたいと考えている。

5.2 他のディスプレイ装置への応用に関して

タッチパネルなどの付加装置や、特殊な表示装置を必要としない本研究でのアプローチは、今回の実装例で示したラップトップ PC のディスプレイに限らず、自己発光型のディスプレイ（例：大型のプラズマディスプレイなど）であれば利用することができる。この特長により、様々な用途に応用できる可能性があると考えている。表示するディスプレイのサイズが変われば、ユーザの体験も全く別のものになることが期待できると思われる。

5.3 ワイヤレス化の必要性に関して

本研究での実装例では、物理コントローラからインタフェース・ボードまでは有線で接続されている。このケーブルは、できるだけ細く柔軟なものを選択したが、それでもコントローラの操作時に気になるという意見が多く聞かれた。このサイズのコントローラをワイヤレス化するには、バッテリーの問題や通信方法、レイテンシの問題など、解決すべき課題が数多くあるが、今後の課題として積極的に取り組んでいきたい。

6 おわりに

今後の研究は以下のような点を中心に進めて行く予定である。また、本研究でのアプローチの有効性に関しては、従来のマウス等の入力装置による操作方法との定量的な比較などを行い、検証を進めていきたいと考えている。

- 無線化（Bluetooth など）
- プラズマ・ディスプレイやリヤプロジェクション方式のプロジェクタでの検証
- ダイアル型以外のコントローラ（例：スライダ）への応用
- Processing など [7]、ユーザ自身が簡単にプログラミングできる他の環境への実装

参考文献

- [1] 藤本・大坪：「MASTER OF REASON 3.0」, ピーエヌエヌ新社, 2005
- [2] W.Buxton, R.Hill & P.Rowley : Issues and techniques in touch-sensitive tablet input. Computer Graphics, 19(3), Proceedings of SIGGRAPH'85, pp. 215–223, 1985
- [3] S.Greenberg and C.Fitchett : Phidgets: easy development of physical interfaces through physical widgets. Proceedings of UIST '01, pp. 209–218, 2001
- [4] N.Villar, A.Lindsay, H.Gellersen : Pin & Play & Perform: A rearrangeable tangible interface for musical composition and performance, Proceedings of NIME05, pp. 188–191, 2005
- [5] 赤松・佐近田：トランス Max エクスプレス, リットーミュージック, 2001
- [6] J.Raskin : 「ヒューメイン・インタフェース」, ピアソンエデュケーション, 2001
- [7] C.Reas and B.Fry : Processing: a learning environment for creating interactive Web graphics, Proceedings of the SIGGRAPH 2003, 2003