



Instituto Politécnico
de Castelo Branco

Instituto Politécnico de Castelo Branco

Ângelo, Tiago Alexandre da Silva

Hipertrombone : HTB1

<https://minerva.ipcb.pt/handle/123456789/3608>

Metadados

Data de Publicação	2020
Resumo	O Hipertrombone (HTB1 - siglas de Hipertrombone) consiste na aplicação de sensores num trombone de varas, originando um novo instrumento. Este novo instrumento ou hiperinstrumento é capaz de conferir novas possibilidades musicais, mantendo ainda as possibilidades obtidas por um trombone convencional....
Editor	IPCB. ESART
Palavras Chave	Trombone, Hiperinstrumentos, Controladores gestuais, Novos instrumentos musicais, Síntese sonora
Tipo	report
Revisão de Pares	Não
Coleções	ESART - Música - Variante de Música Eletrónica e Produção Musical

Esta página foi gerada automaticamente em 2024-04-27T15:06:37Z com informação proveniente do Repositório

Hipertrombone : HTB1



Tiago Alexandre da Silva Ângelo



Instituto Politécnico de Castelo Branco
Escola Superior de Artes Aplicadas

Setembro de 2010

Hipertrombone : HTB1

Tiago Alexandre da Silva Ângelo

Relatório submetido para as disciplinas de Projecto Individual I e
Projecto Individual II do curso de Música Electrónica e Produção
Musical

Orientado por Rui Dias



Instituto Politécnico de Castelo Branco
Escola Superior de Artes Aplicadas

Setembro de 2010

I. Resumo

O Hipertrombone (HTB1 - siglas de *Hipertrombone*) consiste na aplicação de sensores num trombone de varas, originando um novo instrumento. Este novo instrumento ou hiperinstrumento é capaz de conferir novas possibilidades musicais, mantendo ainda as possibilidades obtidas por um trombone convencional.

Palavras-chave: trombone, hiperinstrumentos, controladores gestuais, novos instrumentos musicais, síntese sonora.

II. Agradecimentos

Aos meus pais Manuel e Maria da Silva Ângelo por todo o apoio prestado.

Ao Rui Dias por ter contribuído para a maior parte do meu conhecimento relacionado com música e tecnologia, por me ter orientado durante a execução deste projecto e por ser um excelente amigo.

Ao Gustavo Costa, pessoa, músico e compositor exemplar que me influenciou de forma bastante positiva.

Ao Carlos Guedes, por todo o seu conhecimento profundo prestado na área da composição e programação.

Ao José Alberto Gomes pela sua contribuição crítica.

As amigos Márcio Filipe, João Menezes e Luís Marques pelo apoio prestado e pela sua contribuição crítica.

III. Índice

I. Resumo

II. Agradecimentos

III. Índice

1. Introdução

2. HTB1-h: Controlador gestual

2.1. O trombone e a aquisição de dados gestuais

2.2. Plataforma: leitura e comunicação dos dados gestuais

2.2.1. Arduino: *hardware*

2.2.2. Arduino: *software*

2.3. Sensores: função e aplicação

2.3.1. Acelerómetro

2.3.2. Sensor de ultra-sons e sensor de infra-vermelhos

2.3.3. Sensor de força

2.3.4. Sensor de sopro

2.3.5. Microfone

2.4. HTB1-h: *hardware*

3. HTB1-s: Processador Sonoro

3.1. Plataforma: ambiente gráfico de programação, Max/MSP

3.2. Síntese sonora

3.2.1. Síntese subtrativa

3.2.2. Síntese *AM* e *RM*

3.2.3. *Flanging*

3.2.4. As Transformadas de Fourier

3.2.4.1. Convolução

3.2.4.2. *Frequency-domain pitch shifting*

3.2.4.3. *Time-Freeze*

3.2.5. Síntese Granular

3.3. HTB1-s: *software*

3.3.1. Módulos

3.3.2. Mapeamento

4. Conclusão

5. Anexos

5.1. Exemplos de hiperinstrumentos

6. Referências

1. Introdução

O uso da tecnologia na área da música não é exclusivo do séc. XX, no entanto os progressos tecnológicos realizados neste século trouxeram uma outra dimensão aos processos de composição e produção musicais, bem como novos instrumentos, afectando de várias formas a música contemporânea. Contando com vários desenvolvimentos no campo da síntese sonora, principalmente desde meados dos anos 50, actualmente contamos com uma larga variedade de sintetizadores sonoros, tanto sob o formato de hardware (electrónicos) como de software (digitais).[6]

Até ao final do séc. XIX o desenvolvimento de instrumentos musicais dependia unicamente das propriedades acústicas de cordas, tubos e membranas. [7] E apesar destes instrumentos acústicos proporcionarem uma paleta de sons bastante diversificada, o advento da electricidade e consequentemente dos sintetizadores electrónicos (também denominados sintetizadores analógicos) causaram um enriquecimento desta paleta sonora, motivando novas experimentações sonoras e musicais. No entanto estes sintetizadores apresentavam algumas limitações relativamente à sua programação, memória e precisão. [6]

“ O desenvolvimento do computador e a subsequente aplicação na síntese, gravação, armazenamento e reprodução musicais é um dos desenvolvimentos mais preponderantes na tecnologia da música do séc. XX” (Ross Kirk). [3] Este desenvolvimento permitiu a Max Mathews, investigador da Bell Telephone Laboratories, realizar as primeiras experiências de síntese sonora num computador (síntese sonora digital). [6] Tendo criado em 1957 o primeiro programa de computador capaz de realizar síntese sonora, intitulado MUSIC I. [3] As repercussões destes desenvolvimentos são evidentes, sendo que presentemente o meio mais comum de síntese sonora é um altifalante controlado por um computador. [6]

O poder de cálculo e armazenamento dos computadores fez com que as limitações dos sintetizadores electrónicos se tornassem nas virtudes dos sintetizadores digitais. Estes proporcionam uma paleta de sons consideravelmente maior que a síntese sonora electrónica uma vez que “qualquer som, desde o mais simples ao mais complexo,

que possa ser produzido através de um altifalante, pode ser sintetizado por este meio, (...)o que proporciona uma atracção óbvia para compositores. Isto porque a síntese sonora digital efectua a ponte entre o que pode ser imaginado e o que pode ser ouvido” (John Chowning). [6]

A execução de sintetizadores (electrónicos e digitais) e a expressão musical por eles conseguida conheceu um aumento exponencial na década de 80, em grande parte devido ao protocolo MIDI (*Musical Instrument Digital Interface*) e ao surgimento de computadores cada vez mais eficientes. [7] E, embora este protocolo tenha sido bastante preponderante no desenvolvimento e concepção de novos instrumentos, não só não conseguiu contornar a tendência que se vinha fazendo notar pela indústria musical ao comercializar o teclado (idêntico ao de um piano) como interface padrão entre músicos e sintetizadores, como contribuiu para a consolidação da mesma. [7]

O aparecimento de computadores pessoais cada vez menos dispendiosos e mais potentes e a vasta disponibilidade de sintetizadores no formato digital possibilitaram músicos e investigadores a implementarem os seus próprios instrumentos musicais digitais, criando uma procura de controladores alternativa ao teclado. Onde ferramentas de programação como o Max/MSP e o PureData (Pd) contribuíram em larga escala. [7]

Segundo Marcelo Wanderley e Eduardo Miranda, um instrumento musical digital ou DMI (*Digital Musical Instrument*) é qualquer instrumento musical que empregue métodos digitais de síntese sonora e possua uma interface de controlo ou um controlador gestual, controladores estes que têm como função afectar parâmetros musicais do sintetizador sonoro em tempo-real. [7] Dentro desta categoria os autores do livro “*New Digital Musical Instruments*” dividem os novos controladores gestuais em quatro grupos, baseados no grau de semelhança com instrumentos existentes:

- Controladores gestuais alternativos: “Controladores alternativos que apresentem poucas ou nenhuma semelhanças com instrumentos existentes”
- Controladores gestuais inspirados em instrumentos: “Controladores gestuais inspirados em instrumentos existentes ou que pretendem ultrapassar algumas das limitações do modelo original, mas que não pretendem reproduzi-los de forma exacta”
- Controladores gestuais tipo-instrumento: “Controladores gestuais que são modelados a partir da superfície de controlo de instrumentos acústicos, com o propósito de reproduzir completamente as suas características iniciais”
- Instrumentos aumentados (também denominados hiperinstrumentos, instrumentos híbridos, ou metainstrumentos : “Instrumentos acústicos aumentados pelo uso de vários sensores” (ver exemplos no cap. 5)

O âmbito deste projecto engloba o desenvolvimento de um hipertrombone, um instrumento aumentado com base num trombone de varas (ver Fig. 1.1 - esquema do hipertrombone). Para tal são necessários os seguintes passos:

- Escolher a informação gestual que se pretende usar de modo a controlar o sistema. (ver cap. 2.1)
- Estabelecer os métodos para captação destes dados (escolha dos sensores). (ver cap. 2.3)
- Definir os processos de síntese sonora constituintes do processador sonoro. (ver cap. 3.2)
- Mapeamento dos dados obtidos pelos sensores para controlo de parâmetros do processador sonoro. (ver cap. 3.3.2) [7]

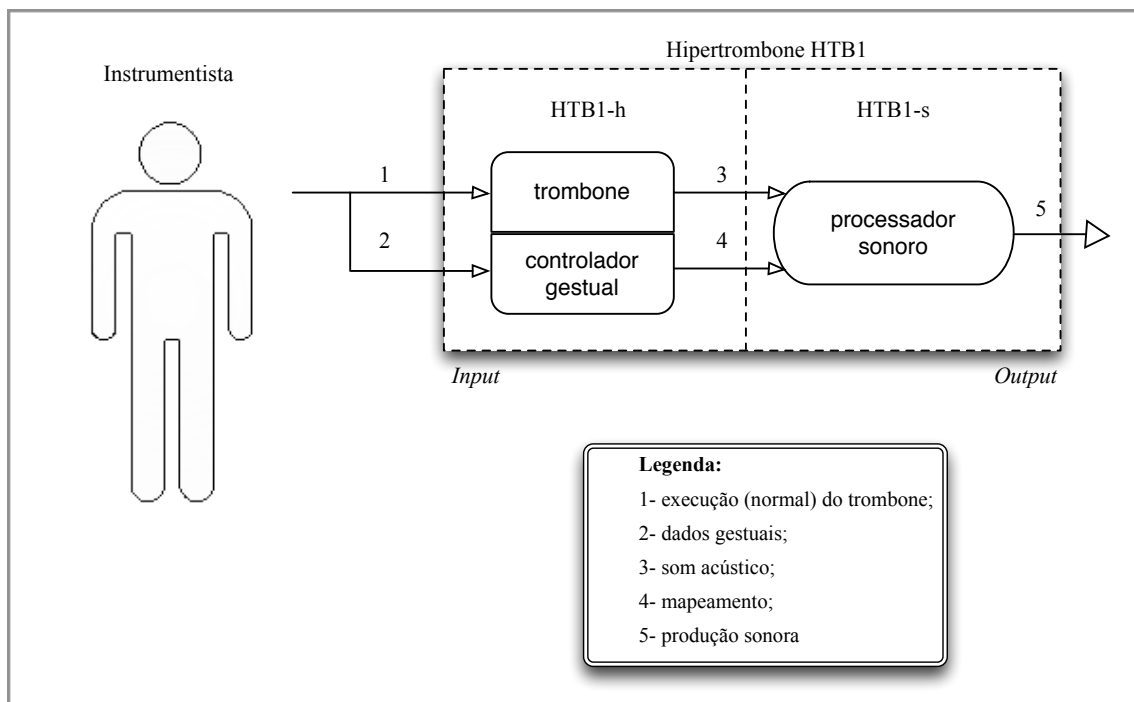


Fig. 1.1 — esquema do hipertrombone.

Ao contrário dos instrumentos acústicos, a separação entre a unidade responsável pela geração sonora e a superfície (ou interface) de controlo (ou também controlador gestual) é possibilitada nos instrumentos digitais. [7] Isto é, se tivermos em conta o trombone, os elementos como o bocal e a vara são simultaneamente responsáveis pela interacção entre músico e instrumentista e pela geração sonora.

Esta divisão em duas unidades independentes pode trazer grandes vantagens, uma vez que possibilita várias formas de mapeamento entre o controlador gestual e o processador sonoro. [6]

A concepção e construção do hipertrombone abrange assim duas etapas de desenvolvimento diferentes:

- Desenvolvimento do controlador gestual (HTB1-h), descrito no capítulo 3.
- Desenvolvimento do processador sonoro (HTB1-s), descrito no capítulo 4. (ver Fig. 1.1)

O objectivo deste projecto consiste portanto no aumento das capacidades tímbricas e musicais do trombone, com vista a alargar o seu repertório bem como criar novas possibilidades composicionais sem no entanto perder as características originais deste. (ver Fig. 1.2)

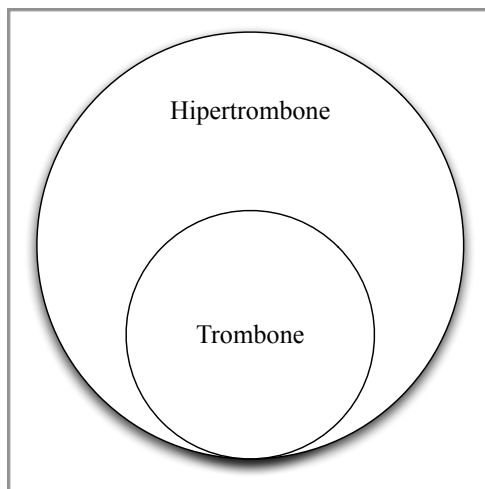


Fig. 1.2 — capacidades (hipotéticas) dos instrumentos

2.HTB1-h: Controlador gestual

No caso dos instrumentos acústicos, os dados constituintes do gesto resultante da interacção entre músico e instrumento podem ser adquiridos de três maneiras distintas:

- Directamente: aquisição de dados de características físicas, relativos às acções do instrumentista. Por norma, cada variável física necessita de um sensor (ver definição no capítulo 3.3) especializado para a sua aquisição.
 - Indirectamente: aquisição de dados relacionados com a produção sonora do instrumento e as suas propriedades. Os sensores usados para este método são normalmente os microfones.
 - Fisiologicamente: aquisição de dados relativos ao funcionamento natural do corpo humano, também denominados biosinais. Este método não é posto em prática neste projecto uma vez que estes dados, regularmente contaminados com ruído, são de difícil obtenção e de certa maneira poderão estar menos próximos da execução do instrumento.
- [7]

Para a elaboração do controlador gestual HTB1-h foram utilizados ambos os métodos, directo e indirecto, para a obtenção de dados gestuais produzidos na interacção entre trombonista e trombone. Contudo o método directo foi o mais preponderante, já que é mais simples que o método indirecto. Este facto está relacionado com a complexidade dos diferentes parâmetros resultantes da produção sonora de um instrumento acústico tradicional. [7]

2.1. O trombone e a aquisição de gestuais

Segundo a classificação de Hornbostel-Sachs o trombone enquadra-se na categoria dos aerofones, pertencendo à sub-categoria dos metais. [2] Este é constituído por três secções principais: a campânula, a vara e o bocal. Consultar a figura 2.1 para uma descrição mais detalhada da constituição do trombone de varas. Esta figura não pretende enumerar todas as partes constituintes de um trombone de varas mas sim fornecer informação necessária para o prosseguimento da leitura deste relatório.

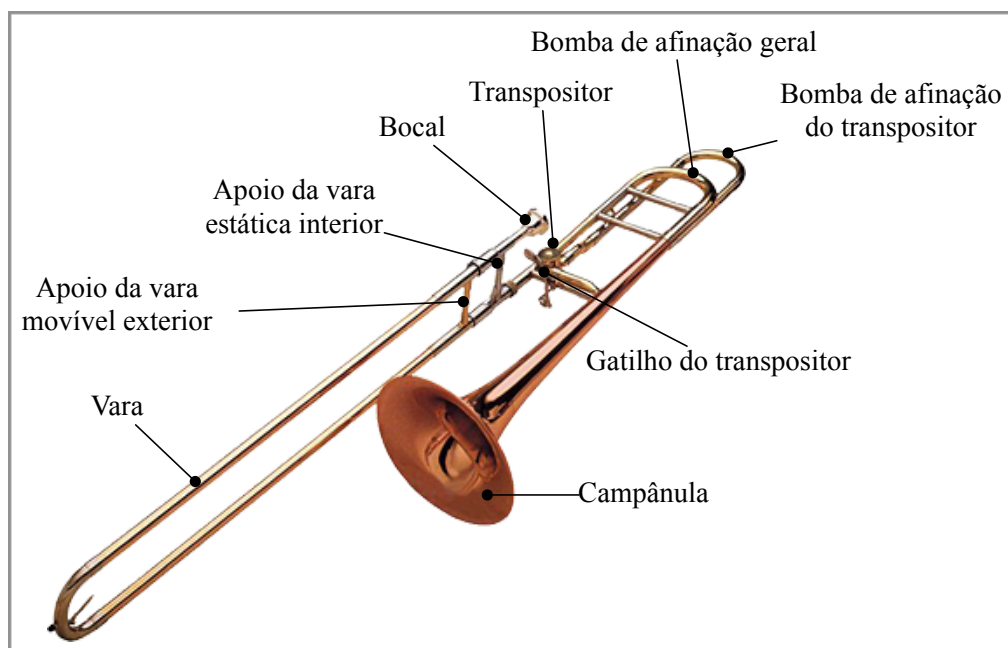


Fig. 2.1 — partes constituintes de um trombone de varas.

Aquisição directa

Os dados gestuais que se pretendem obter, segundo o método directo, no desenvolvimento deste projecto englobam o movimento do próprio corpo do instrumento, a movimentação e posicionamento da vara, pressão do fluxo de ar e a pressão exercida sobre o gatilho do transpositor. A escolha destes será discutida com maior rigor no capítulo 4.3 conjuntamente com os sensores capazes de perceber estes movimentos. Enquanto que o mapeamento destes se encontra descrito no cap. 5.

Aquisição indirecta

Wanderley e Depale (2004) identificam 4 parâmetros sonoros possíveis da aquisição indirecta:

- Energia do sinal, correspondendo ao nível dinâmico ou de amplitude do som ou possivelmente à distância do instrumento ao microfone.
- Frequência fundamental, representa a altura do som e está relacionada com a sua característica melódica.
- Envelope espectral, parâmetro sonoro que envolve a distribuição dos parciais sonoros de amplitude e está de certa forma também relacionado com a energia conferida ao sinal.
- A aquisição dos parciais de frequência, amplitude e fase de um determinado som podem por si só apresentar a informação obtida pelos parâmetros interiores. [7]

Na elaboração deste projecto foram apenas utilizados o primeiro e o último parâmetro, uma vez que demonstraram ser suficientes para as necessidades de controlo impostas pelo processador sonoro. O capítulo 2.3.5 trata da sua obtenção enquanto que o capítulo 3 trata da sua utilização.

A aquisição de todos os dados possíveis da interacção entre músico e instrumento seria um trabalho fustigante no que diz respeito à concepção de um instrumento aumentado. O que faz com que a escolha destes seja um passo bastante importante.

2.2. Plataforma: leitura e comunicação dos dados gestuais

A forma mais prática de ler os dados gestuais obtidos pelos sensores é através da utilização de uma plataforma (ou interface) de prototipagem que nos permitirá variados tipos de configurações de componentes electrónicos. Recorrendo a um determinado protocolo de comunicação que nos permitirá aceder a estes dados num computador. “Na comunicação serial são enviados impulsos digitais consecutivamente entre computadores. (...) Um protocolo é o conjunto de parâmetros sobre o qual os dois dispositivos entram em conformidade de modo a enviarem informação.” [4]

Normalmente dotadas de várias entradas e/ou saídas analógicas e digitais estas plataformas de prototipagem electrónica actuam como interface entre sensores e computador. Adaptando-se notavelmente às necessidades exigidas pela concepção de novos controladores gestuais, dada a sua enorme flexibilidade.

Os fundamentos intrínsecos à escolha desta interface residiram sobretudo em quatro aspectos (ver Tabela 1.1):

- Acessibilidade, ou seja, facilidade de utilização e programação, para além da compatibilidade com o sistema operativo Mac OS X e com o *software* de programação Max/MSP).
- Tipo de comunicação, sendo que para este projecto o mais indicado seria um protocolo de comunicação sem fios no sentido de evitar o obstrucionismo causado pelos cabos na performance do hiperinstrumento.
- Um mínimo de 6 entradas analógicas e 5 digitais;
- Preço da interface.

Tabela 1.1 — Comparação de plataformas de prototipagem. (Apenas são apresentadas interfaces compatíveis com os requerimentos descritos anteriormente)

Interface	Acessibilidade	Comunicação	Preço (aprox.)
PhidgetInterfaceKit 8/8/8	muito boa	USB (cabo)	€60
Aduino Duemilanove	boa	USB (cabo)	€20
Miditron	boa	MIDI (cabo)	€115
Miditron Wireless	boa	MIDI (sem-fios)	€380
Electrotap Teabox	razoável	S/PDIF (cabo)	€370
WiSe Box	muito boa	WiFi/OSC (sem-fios)	€800
Arduino Bluetooth	boa	Bluetooth (sem-fios)	€90

Após alguma pesquisa conclui-se que a plataforma Arduino Bluetooth, apesar de não ser a mais indicada, quando comparada por exemplo com a Wise Box que nos permite estabelecer conexões com 16 dispositivos compatíveis com WiFi e OSC simultaneamente, é no entanto a opção que melhor serve os propósitos deste projecto, mantendo um orçamento minimamente aceitável. Para além do mais representa uma vantagem significativa na definição dos dados obtidos quando comparada com dispositivos MIDI. Uma vez que as mensagens MIDI têm uma definição de 7 *bits* (0~127) enquanto que as processadas pela Arduino têm 10 *bits* (0~1023). [9]

“Arduino é uma plataforma de computação física *open-source* baseada numa simples placa dotada de entradas e saídas (I/O - *Input/Output*) e num ambiente de programação(...)” (IDE - *Integrated Development Environment*) “(...)que implementa a linguagem de programação *Processing*.” [1]

A placa/interface Arduino Bluetooth, também denominada ArduinoBT, providencia um total de 14 entradas/saídas digitais e 6 entradas analógicas e como o próprio nome indica recorre ao protocolo *Bluetooth* (tecnologia sem-fios) para estabelecer uma comunicação serial com o computador.

2.2.1. Aplicação: hardware

Esta secção refere-se à componente física da plataforma Arduino: a placa ou interface de prototipagem na qual são montados os sensores e restantes componentes electrónicos constituintes do sistema a desenvolver, neste caso o controlador gestual.

Componentes da ArduinoBT

Os componentes constituintes desta interface podem ser visualizados na figura 2.2, que embora não apresente uma legenda completa, inclui as partes necessárias para a compreensão do relatório.

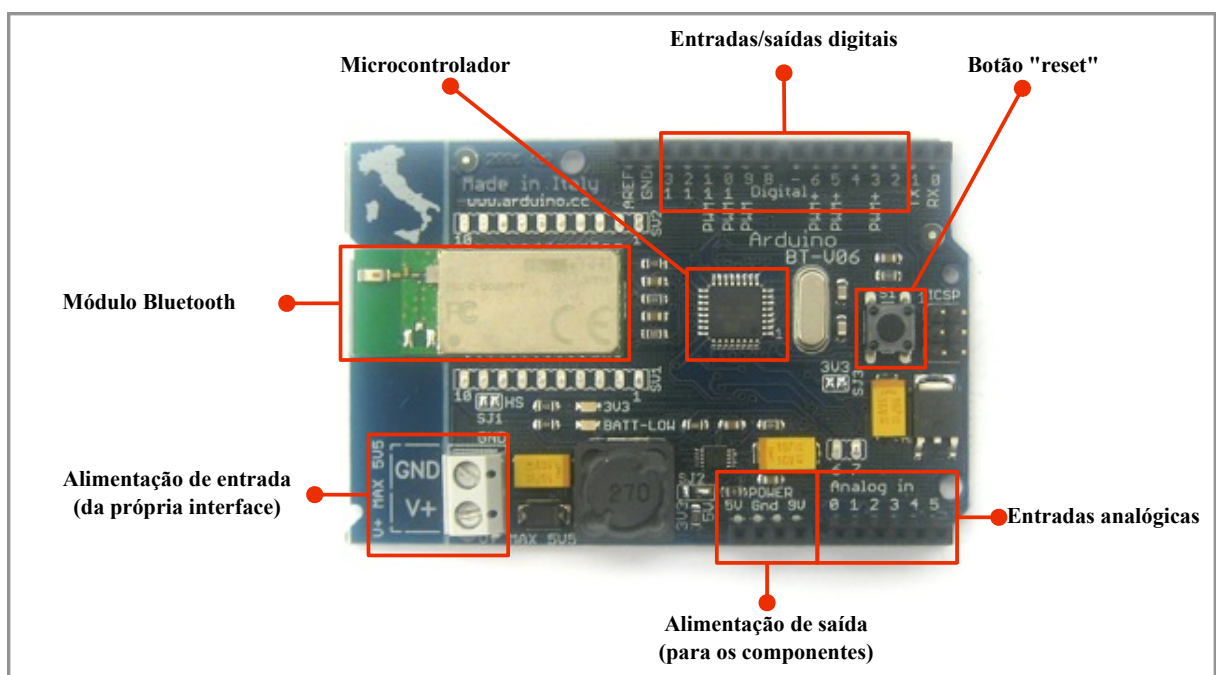


Fig.2.2 — componentes da interface ArduinoBT.

Desta legenda constam as seguintes partes/componentes:

- Alimentação de entrada: esta interface necessita de uma alimentação de no máximo 5.5V (caso contrário poderão ocorrer danos no microprocessador).
- Alimentação de saída: alimentação usada para os componentes que se pretendam conectar à interface. É daqui que provém a alimentação para os sensores.
- 6 entradas (ou pinos) analógicas: local responsável pela leitura de sinais contínuos no tempo, como é o caso dos sinais eléctricos. É nestes pinos que se conectam a maioria dos sensores.
- 14 entradas/saídas digitais: estes pinos podem ser configurados pelo código introduzido através do IDE (ver próximo subcapítulo) no microcontrolador servindo os propósitos de leitura e envio de sinais discretos, como é o caso dos sinais digitais. Este pinos são bastante versáteis, podendo adquirir e transferir sinais binários(0/1) tal como uma aproximação dos sinais contínuos através da modulação da largura de pulso (técnica apenas referenciada neste relatório uma vez que esta não é crucial para o desenvolvimento do projecto, apesar de ser utilizada num dos sensores)
- Microcontrolador: o núcleo da interface, este chip está a cargo da execução do código introduzido no IDE, fazendo a ponte entre a recepção, modulação, e envio dos dados.
- Módulo *Bluetooth*: responsável pela comunicação serial entre a interface e o dispositivo *bluetooth* do computador.
- Botão “reset”, serve simultaneamente para reiniciar o código introduzido no microchip e para estabelecer/re-estabelecer uma ligação entre dispositivo *bluetooth* da Arduino e do computador.

Alimentação da placa

Quanto a este assunto identificaram-se dois problemas, descritos hierarquicamente juntamente com a sua resolução:

1. A escolha de uma única bateria, de maneira a simplificar o controlador gestual, principalmente em relação ao peso e ergonomia e *design* do mesmo. E garantindo simultaneamente tensão eléctrica (ou voltagem) suficiente. Na resolução deste problema foi escolhida uma pilha de 9V (rectangular), porque é a pilha mais vulgarmente comercializada que por si só é capaz de uma potência de pelo menos 5V.
2. A resolução do primeiro problema dá origem no segundo, a tensão debitada pela pilha de 9 volts tem que ser reduzida para 5 para compatuar com a alimentação possível da placa. No entanto a solução é de fácil obtenção, sendo necessário apenas um componente conectado entre a pilha e a entrada de alimentação da ArduinoBT: um divisor de voltagem. (ver Fig. 2.21)

Colocação e acoplagem da Arduino ao trombone

O *design* de um instrumento aumentado implica, na maior parte dos casos, uma acoplagem física entre o controlador gestual e o instrumento. Como todos os sensores envolvidos neste projecto envolvem de algum modo o contacto directo com o trombone, seria

conveniente colocar a unidade responsável pela transdução dos sensores (ArduinoBT) o mais perto possível destes. A solução mais óbvia pareceu a colocação e acoplamento da Arduino Bluetooth entre o tubo anterior à bomba de afinação e o tubo posterior (anterior à câmpanula, ver Fig.2.3.). Isto porque na execução do instrumento esta é superfície que mais regularmente se encontra paralela à linha do chão, o que permite firmar a interface

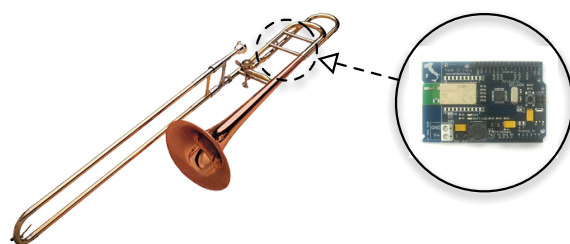


Fig. 2.3 — colocação da interface no trombone.

com maior facilidade e segurança. Para além de não impossibilitar os movimentos do músico na execução do instrumento e permitir espaço suficiente para ArduinoBT.

Para melhorar a superfície de encaixe e expandir a protecção da placa Arduino Bluetooth, visto que esta constitui um aspecto central na descodificação dos dados gestuais que se visam captar, esta foi montada numa caixa plástica de prototipagem. Assim sendo, não é a interface Arduino que está directamente acoplada ao trombone, mas sim a caixa, recorrendo ao uso de fita de velcro para o efeito.

2.2.2. Aplicação: software

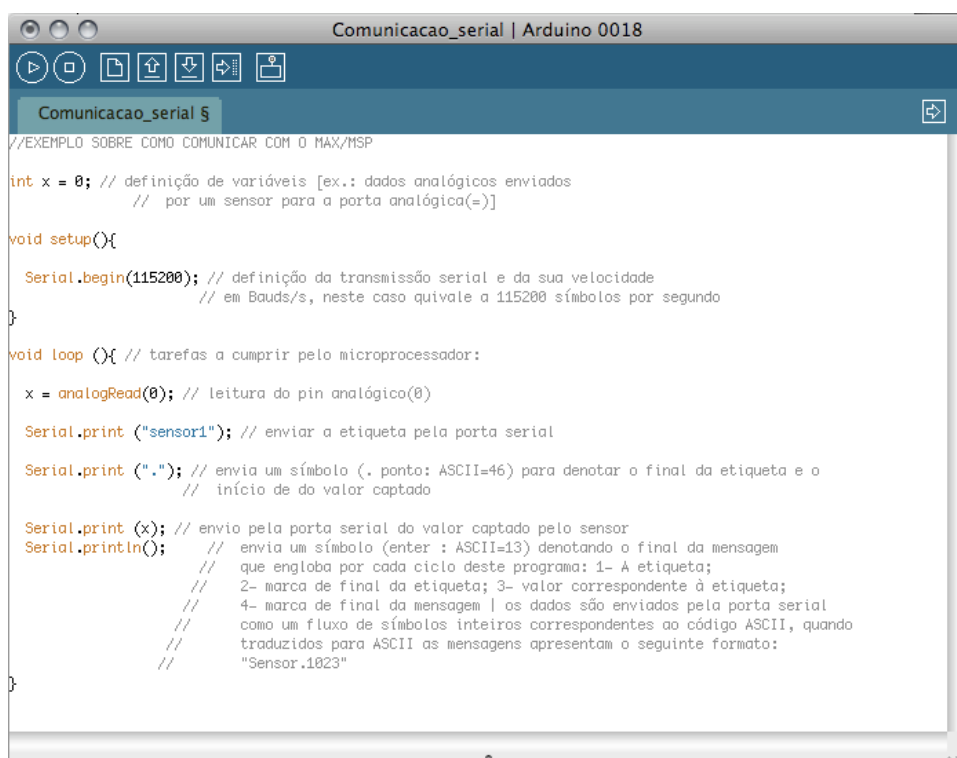
Esta secção refere-se à componente virtual da plataforma Arduino: o IDE (*Integrated Development Environment* ou Ambiente Integrado de Desenvolvimento). Este é o que nos permite carregar um código (denominado *sketch* no programa Arduino) para o microprocessador incorporado na interface física, fornecendo-lhe informação sobre as tarefas exactas a realizar. Este Ambiente Integrado de Desenvolvimento recorre à linguagem Arduino, baseada na linguagem de programação *open-source*: Processing.

Comunicação serial entre Max/MSP e Arduino

Nesta secção são descritos os meios necessários, bem como um método conveniente para receber os dados provenientes da placa Arduino no *software* de programação Max/MSP (consultar cap.3.1)para posterior mapeamento (ver cap.3.3.2).

Através do ambiente de programação Arduino é carregado no microprocessador um programa responsável pela aquisição dos dados obtidos pelos sensores (sinais eléctricos)

e posterior envio destes pela porta serial no formato de sinais digitais. (ver exemplo da Figura 2.4)



```
//EXEMPLO SOBRE COMO COMUNICAR COM O MAX/MSP

int x = 0; // definição de variáveis [ex.: dados analógicos enviados
          // por um sensor para a porta analógica(=)]

void setup(){

  Serial.begin(115200); // definição da transmissão serial e da sua velocidade
                       // em Bauds/s, neste caso quivale a 115200 símbolos por segundo
}

void loop () { // tarefas a cumprir pelo microprocessador:

  x = analogRead(0); // leitura do pin analógico(0)

  Serial.print ("sensor1"); // enviar a etiqueta pela porta serial

  Serial.print ("."); // envia um símbolo (. ponto: ASCII=46) para denotar o final da etiqueta e o
                     // início de do valor captado

  Serial.print (x); // envio pela porta serial do valor captado pelo sensor
  Serial.println(); // envia um símbolo (enter : ASCII=13) denotando o final da mensagem
                  // que engloba por cada ciclo deste programa: 1- A etiqueta;
                  // 2- marca de final da etiqueta; 3- valor correspondente à etiqueta;
                  // 4- marca de final da mensagem | os dados são enviados pela porta serial
                  // como um fluxo de símbolos inteiros correspondentes ao código ASCII, quando
                  // traduzidos para ASCII as mensagens apresentam o seguinte formato:
                  // "Sensor.1023"
}
```

Fig. 2.4 — *sketch* Arduino exemplificando o método utilizado para envio dos dados adquiridos pelos sensores.

No Max/MSP, o <objecto> serial, que tem como função a recepção e envio de dados através de portas seriais, recebe os sinais digitais sob a forma de um fluxo de números inteiros, em código ASCII (ver definição em Anexos). Como o objectivo passava pela aquisição individual dos dados de cada sensor, foi adoptado um método (semelhante a um protocolo) para a identificação dos dados físicos obtidos na placa Arduino, que sendo enviados como um fluxo de números inteiros etiquetados, eram recebidos e convertidos no Max/MSP de maneira a obter dados concretos e manuseáveis acerca do estado físico dos sensores. Sendo que o formato final destas mensagens consiste na identificação (etiqueta), que contém informação acerca da origem da variável obtida (por ex: sensor1), e na variável (estado ou valor obtido pelo componente etiquetado). Consultando as figuras 2.4 e 2.5 pode-se perceber o funcionamento deste método.

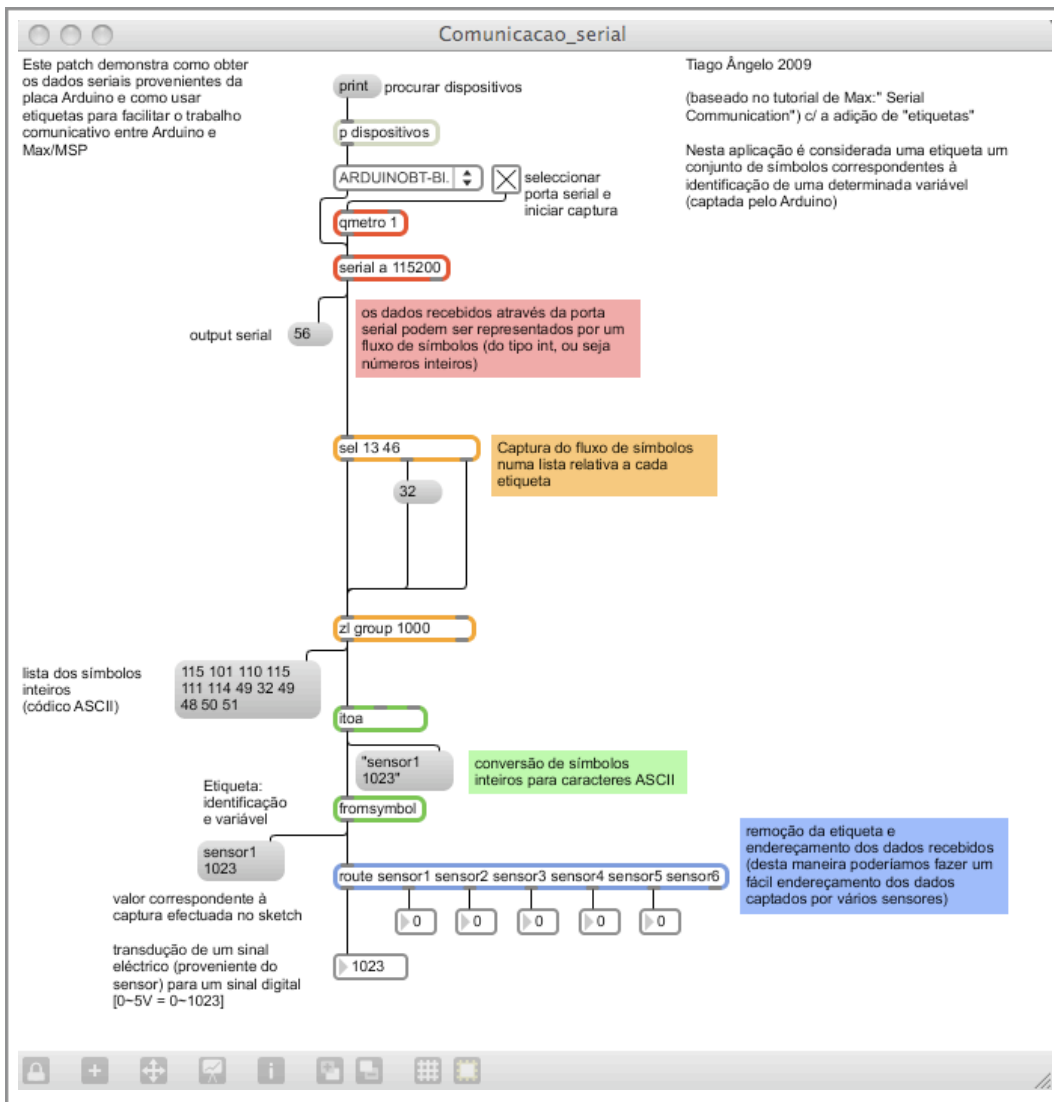


Fig. 2.5— *patch* Max/MSP exemplificando a comunicação, leitura e conversão dos dados provenientes da comunicação serial estabelecida com a placa Arduino.

2.3. Sensores: função e aplicação

“Um dos princípios fundamentais por detrás da computação física é a transdução, ou conversão de uma forma de energia para outra.” (Tom Igoe e Dan O’Sullivan)[4]

Os transdutores, componentes responsáveis pela transdução, podem ser categorizados em sensores, actuadores ou uma mistura dos dois. Os actuadores são componentes capazes de converter sinais eléctricos noutros tipos de energia, sendo o altifalante ou coluna de som um dos actuadores mais vulgares. Estes poderiam ter sido, por exemplo, usados no *feedback* táctil do controlador gestual, no entanto a realização deste projecto enfatiza a aquisição de dados gestuais em detrimento do *feedback* obtido na execução do hiperinstrumento.[4]

Por sua vez, os sensores são componentes capazes de transformar várias formas de energia, como por exemplo calor, movimento, luz e pressão acústica, em sinais eléctricos susceptíveis de serem “compreendidos” por um computador. O microfone é um exemplo de um sensor sobejamente conhecido. [4]

As formas de energia que se pretendiam captar com os sensores para a concepção do controlador gestual HTB1-h, abrangiam os seguintes parâmetros:

- Movimento do trombone causado pelo instrumentista;
- Posição e distância da vara do trombone em relação ao seu estado de repouso(1ª posição, ver Tabela 1.4);
- Fluxo de ar ou sopro do instrumentista;
- Toque do polegar esquerdo na gatilho do transpositor;
- Ondas acústicas produzidas pela *performance* do trombonista.

Visando a captação destes parâmetros, os sensores explorados neste projecto e a sua função, ou gesto capazes de transduzir, bem como o método de aquisição que apresentam (directo ou indirecto) são referenciados na seguinte tabela:

Tabela 1.2 — sensores usados para a obtenção de dados do controlador HTB1-h.

Sensores	Gesto	Aquisição
acelerómetro	movimento/aceleração	directa
sonar (ou sensor de ultra-sons)	distância de um objecto	directa
sensor de infra-vermelhos	distância de um objecto	directa
sensor de pressão/força	pressão/força aplicada a um objecto	directa
sensor de sopro	sopro	directa
microfone	pressão acústica	indirecta

Uma abordagem comparativa relativamente aos sensores disponíveis no mercado não é realizada visto não ter tanta importância para este relatório como a escolha da plataforma, abordada no capítulo 2.2. No entanto é conveniente salientar que a escolha destes se debruçou basicamente na capacidade de adquirir os dados pretendidos e na acessibilidade: facilidade de utilização, montagem e programação. Explicações extensivas acerca do funcionamento dos sensores não são também abordadas neste relatório dada a complexidade da sua natureza.

2.3.1. Acelerómetro

Um acelerómetro, tal como o próprio nome indica, é um sensor capaz de perceber a aceleração ou velocidade do movimento. Esta aceleração pode-se manifestar de duas formas: de forma dinâmica ou estática:

- A forma dinâmica engloba a aceleração do sensor quando este é movido de um ponto para outro.
- A forma estática refere-se à força gravitacional exercida quando o sensor roda sobre o seu próprio eixo, sendo possível calcular o grau de inclinação do sensor.

MMA7260Q

Este acelerómetro de três eixos (ver Fig. 2.6) tem como limite de alimentação o valor de 3.3V. Como a saída de alimentação da placa Arduino para os sensores é de 5V usou-se uma técnica de divisão de voltagem recorrendo à montagem de duas resistências em paralelo. (ver Fig. 2.8) A diferença de potencial entre a entrada(V_{in}) e saída(V_{out}) de voltagem é dada pela proporção: $(R1+R2):R2$.

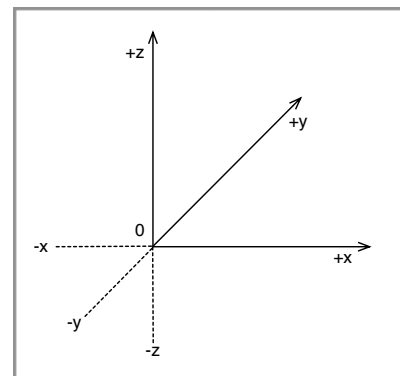


Fig.2.6 — representação gráfica de 3 eixos x, y e z.

O mesmo efeito pode ser alcançado com uma resistência variável (ou potenciômetro, ver Fig.2.7) uma vez que este é basicamente constituído por duas resistências. Tendo apenas que ajustar o nível da resistência manualmente. Portanto, usando uma resistência variável de 4.7 KOhm e ajustando o valor de R1 para cerca de 1,7K e o valor de R2 para 3 KOhm temos: $(R1 + R2) : R2 \Leftrightarrow (1,7 + 3) : 3 \Leftrightarrow 4,7 : 3$. Ou seja, conseguimos uma proporção de 4.7 volts de entrada para 3 de saída, o que soluciona o problema de alimentar o acelerómetro de forma segura.

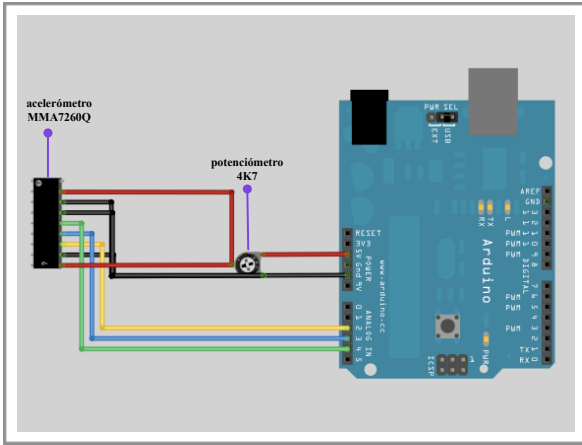


Fig.2.7 — esquema das conexões entre o sensor e a placa Arduino

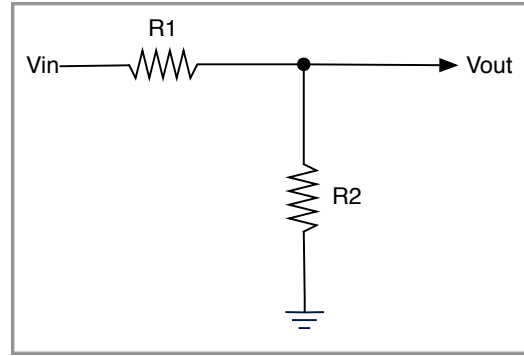


Fig.2.8 — diagrama de um divisor de voltagem empregando resistências montadas em paralelo.

Como para o desenvolvimento do instrumento aumentado HTB1 a única forma de controlo requerida pelo processador sonoro foi a inclinação no eixo y, de maneira a obter informação acerca da postura do instrumentista. Isto é, se tem o instrumento mais ou menos erguido na vertical. Este facto simplificou a aquisição de dados uma vez que não se revelou necessário o cálculo de grandezas reais, e que foram portanto obtidos através de ensaios experimentais.

Acoplagem ao instrumento

Este sensor foi montado numa pequena caixa feita de contraplacado para permitir a sua protecção e estabilidade no acoplamento.

Esta caixa foi então acoplada à campânula do trombone pois é o local mais sujeito a movimentações (tirando a ponta da vara) e onde podemos colocar o sensor sem obstruir a *performance* do instrumentista (ao contrário da ponta da vara). (ver Fig. 2.9)

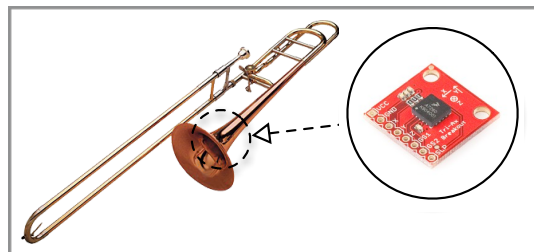


Fig. 2.9 — localização do sensor no trombone.

2.3.2. Sensores de ultra-sons e infravermelhos

Normalmente dotados dum emissor e dum receptor, estes sensores permitem saber a distância entre estes e uma superfície reflectora. Calculando a diferença temporal entre o envio de um sinal e a recepção do mesmo é possível determinar a distância entre o sensor e a superfície. A intenção do uso destes dois sensores recai sobre uma leitura concisa da posição da vara do trombone.

Tabela 1.3 — sensores e características.

Sensor	Tipo	Distância mínima (cm)	Distância máxima (cm)
Sharp GP2D120XJ00F	infra-vermelhos	4	30
LV-MaxSonar - EZ1	ultra-sons	15.24	645

Como a distância da vara de um trombone (tenor em Sib, como é o caso) vai desde os 0 aos 61,2 centímetros (ver Tabela 1.4), e as distâncias calculadas pelos sensores (infravermelhos e ultra-sons) vão desde os 4 aos 30 cm e dos 15,24 aos 645cm (respectivamente), o uso dos dois sensores é obrigatório quando se pretende uma leitura da posição da vara em toda a sua extensão. (ver Tabela 1.3)

Tabela 1.4 — dados relativos à posição da vara de um trombone tenor em Sib.[10]

Posição	Nota Fundamental	Distância (cm)
1	Sib / Lá#	0
2	Lá	8.8
3	Láb / Sol#	18.1
4	Sol	28
5	Solb / Fá#	38.4
6	Fá	49.5
7	Mi	61.2

Sharp GP2D120XJ00F

Este sensor apesar de ser bastante fácil de conectar (ver esquema na Fig.2.11), dificultou a aquisição de dados uma vez que têm um funcionamento não-linear, apresentado uma curva cuja fórmula não é referenciada (ver curva na Fig.2.10). A aquisição destes dados passou assim por uma série de ensaios até chegar a valores concisos.

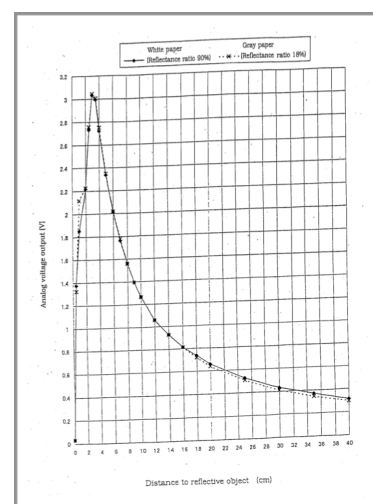


Fig.2.10 — curva de funcionamento do sensor [eixo x: distância(cm); eixo y: tensão eléctrica (y)]

Acoplagem ao instrumento

Dado que ambos os sensores (infravermelhos e ultra-sons) tinham a função de captar a distância da vara do trombone em relação à sua posição inicial (1ª posição), a sua montagem numa única caixa tornou-se óbvia.

Ambos necessitavam de uma superfície reflectora em ordem efectuarem o cálculo da distância e após alguns testes usando o apoio da vara exterior (ver Fig. 2.12) para este efeito, denotou-se que este reflectia com sucesso os sinais enviados pelos sensores, evitando assim a adição de um componente reflector ao instrumento.

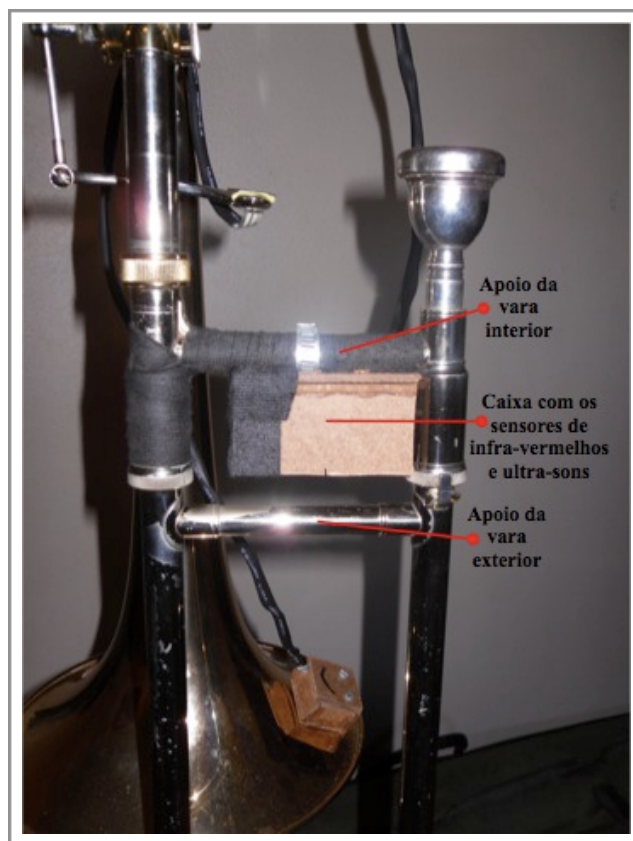


Fig. 2.12 — localização da caixa dos sensores de distância.

2.3.3. Sensor de força

O sensor usado é denominado FSR (*Force Sensing Resistor*) e é basicamente uma resistência de variação susceptível à força a ele aplicada. Quanto maior for a força aplicada a este menor será o seu valor de resistência (ver gráfico da figura 2.13). Logo, quanto mais força aplicarmos maior será a voltagem recebida na placa Arduino.

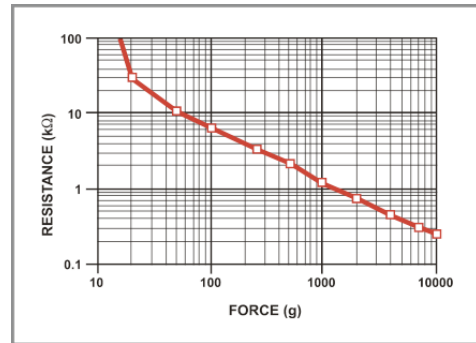


Fig. 2.13 — relação força-resistência do sensor FSR

Existem duas condições para o tipo de informação que se pretende obter com este sensor:

1. O instrumentista tem o dedo pousado no gatilho? Verdadeiro ou falso?
2. Mudança do estado da condição anterior de falso para verdadeiro.

A aplicação do sensor (ver Fig. 2.14 e 2.15) e obtenção destes dados é bastante simples. Sendo no entanto necessário um algoritmo que capaz de transformar o sinal enviado pelo sensor num sinal binário que cumpra as condições descritas anteriormente.

Acoplagem ao instrumento

Para esta função foi escolhido um sensor apropriado ao toque humano e cujo tamanho se enquadrasse na aplicação ao gatilho transpositor do trombone (ver figura em baixo).

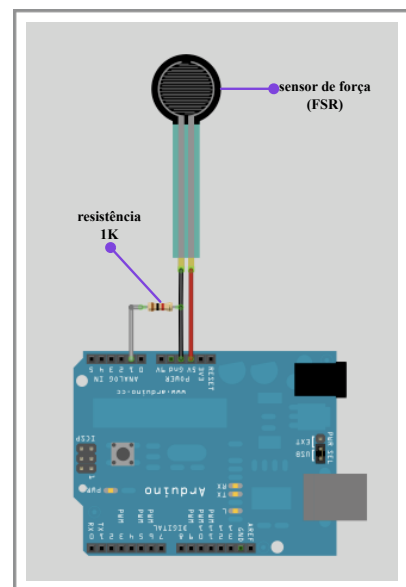


Fig. 2.14 — esquema das conexões entre a placa Arduino e o FSR.

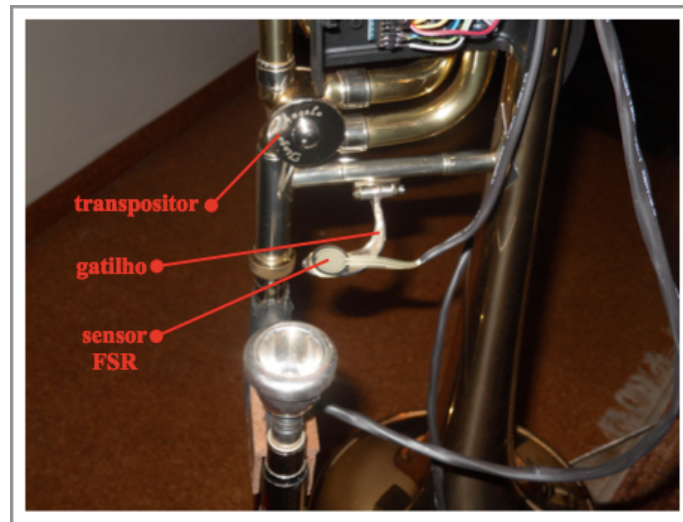


Fig. 2.15 — localização e aplicação do sensor de força no gatilho do transpositor.

2.3.4. Sensor de sopro

Este foi o sensor que proporcionou maiores dificuldades uma vez que se teria que recorrer à importação de um, já que, desconhecendo a razão, não se encontraram sensores de pressão de ar (apropriados à respiração humana) comercializados em Portugal.

A solução passou portanto na construção de um sensor de pressão de ar com base num sensor piezoelétrico (também denominado *piezo*). Estes sensores são capazes de transformar sinais piezoelétricos em sinais eléctricos, quando excitados de forma mecânica, como por exemplo dobrando, percutindo, ou neste caso soprando.[4]

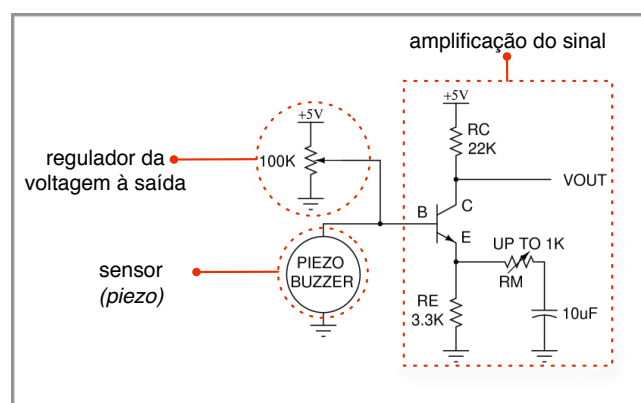


Fig. 2.16 — diagrama do sensor de sopro consultado no tutorial “Piezo Electric Sound Meter”. Acedido em http://www.nerdkits.com/videos/sound_meter/ a 14 de Dezembro de 2009.

O alcance da voltagem produzida por um *piezo* abrange tanto mudanças bruscas de algumas centenas de *volts* como mudanças mais pequenas de apenas alguns *microvolts*. Como a forma mecânica de excitação empregue (sopro) produz diferenças de voltagem bastante pequenas é necessária a amplificação do sinal para que este possa ser percebido pela interface Arduino, ou seja sinais com um âmbito de tensão eléctrica entre os 0 e os 5 *volts*.

Recorrendo ao diagrama da figura 2.16 construiu-se o sensor de sopro esquematizado na figura 2.17. Este possui uma secção encarregada de amplificar o sinal introduzido pelo *piezo*, o próprio *piezo* e uma resistência variável (ou potenciómetro) para regular a amplificação do sinal.

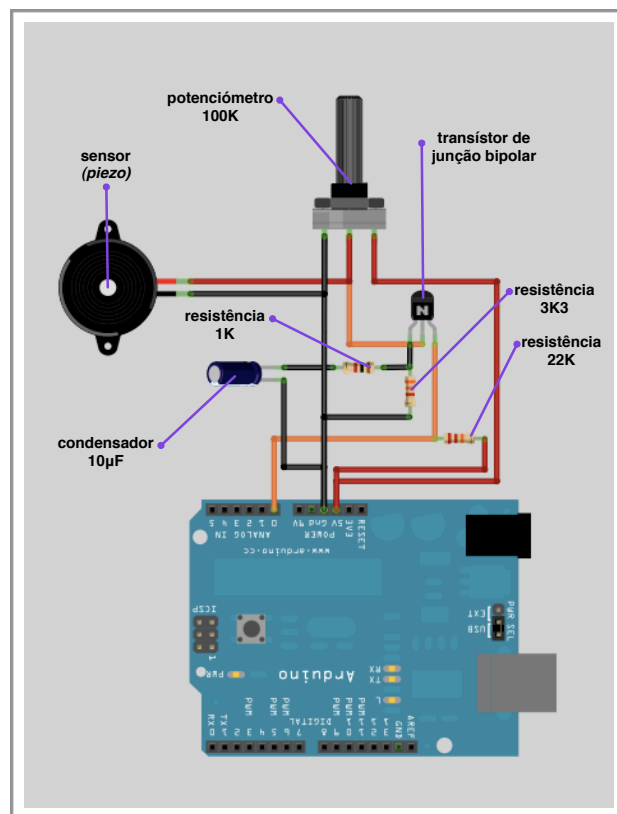


Fig.2.17 — esquema das conexões do sensor de pressão de ar.

O sinal proveniente do sensor de sopro é recebido na porta analógica da ArduinoBT e posteriormente enviados para o Max/MSP. Aqui um algoritmo responsável pelo tratamento deste sinal calcula o valor médio (do sensor em repouso), mínimo e máximo do sinal, efectuando ainda uma interpolação entre valores e nova amplificação.

Acoplagem ao instrumento

A captação da pressão do fluxo de ar é obtida removendo a bomba de afinação do transpositor, colocando, com a ajuda de um suporte, o sensor piezoeléctrico à saída deste tubo (ver Fig. 2.18). O resto do circuito do sensor de pressão de ar (amplificador e regulador) estão montados numa placa própria para o efeito e que colocada por cima da ArduinotBT.

Uma vez que foi removida a bomba do tubo transpositor, quando o gatilho é pressionado, a saída de ar deixa de ser a campânula para passar a ser o tubo transpositor. Deixando portanto a execução normal do uso do transpositor (também denominado de rotor ou disjuntor) impossibilitada.

A execução deste passa agora a ser exclusiva à detecção do fluxo de ar. E, embora seja executável, não é necessária a vibração dos lábios (chamado *buzzing*), devendo o instrumentista soprar pelo bocal.

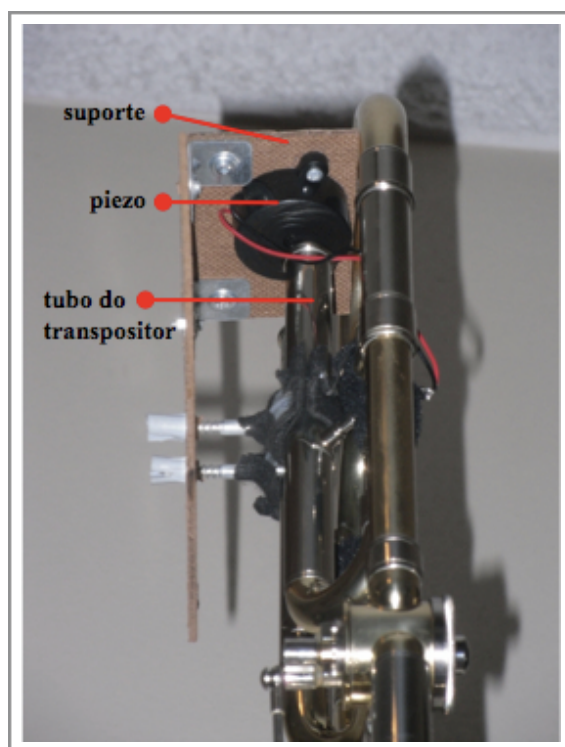


Fig. 2.18 — localização e aplicação do sensor piezoeléctrico.

2.3.5. Microfone

O microfone é o único sensor cujos dados não são recebidos pela interface Arduino. Uma vez que os dados que aqui se pretendem captar para além de estarem directamente relacionados com os métodos indirectos (ver capítulo 2.1) têm também a função de sinal (acústico) de entrada no processador sonoro. Sendo portanto uma placa de som o meio mais conveniente para este efeito.

Os dados captados passam pela utilização da distribuição dos parciais sonoros de frequência, amplitude e fase (estes serão abordados no cap. 3.2.4) e também da energia do sinal. Quanto a este último foi desenvolvido um algoritmo capaz de detectar quando este sinal ultrapassa um determinado limiar (*threshold*) de amplitude.

Dentro de uma variada gama de microfones comercialmente disponíveis foi escolhido um Audio-Technica Pro 35XCW por ser um microfone sem-fios com uma boa relação qualidade-preço. Este é um microfone de condensador anexado a uma pinça para colocação na campânula. (ver Fig. 2.19, na página seguinte)

De forma geral, foi escolhido equipamento sem-fios para evitar ao máximo o obstrucionismo causado por cabos, auferindo assim liberdade de movimento ao *performer*.

2.4. HTB1-h: revisão

Resumindo o capítulo 2, a secção do instrumento HTB1 relativa à aquisição de dados (controlador gestual) intitula-se HTB1-h ('h' - *hardware*) é constituída pelos blocos identificados na figura em baixo:

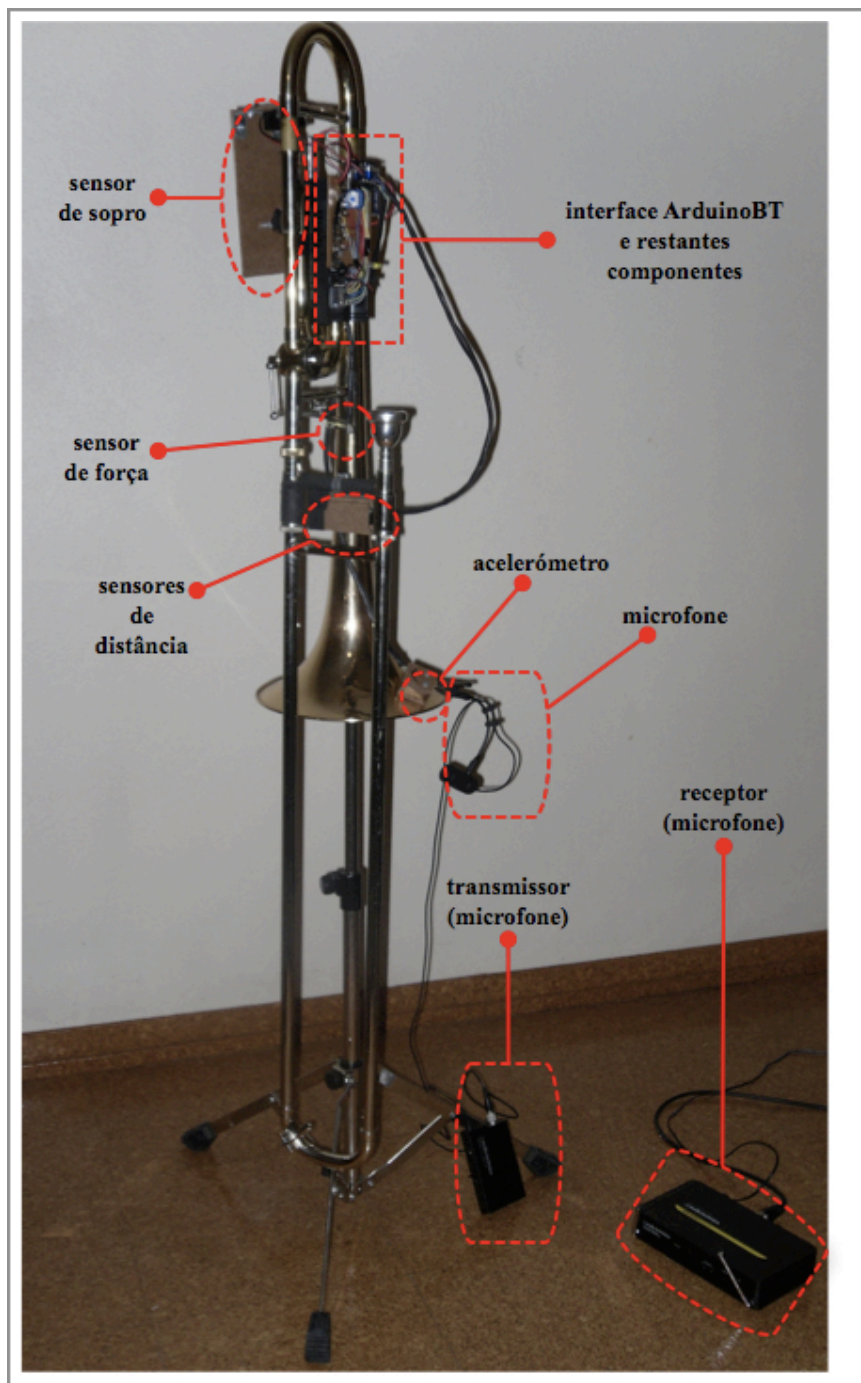


Fig. 2.19 — blocos constituintes do controlador gestual HTB1-h.

Para além dos sensores mencionados no capítulo 2.3 foram também adicionados 4 interruptores (ver Figura 2.20 e 2.21) para activar/desactivar os quatro módulos do processador sonoro (ver cap. 3.3.1).

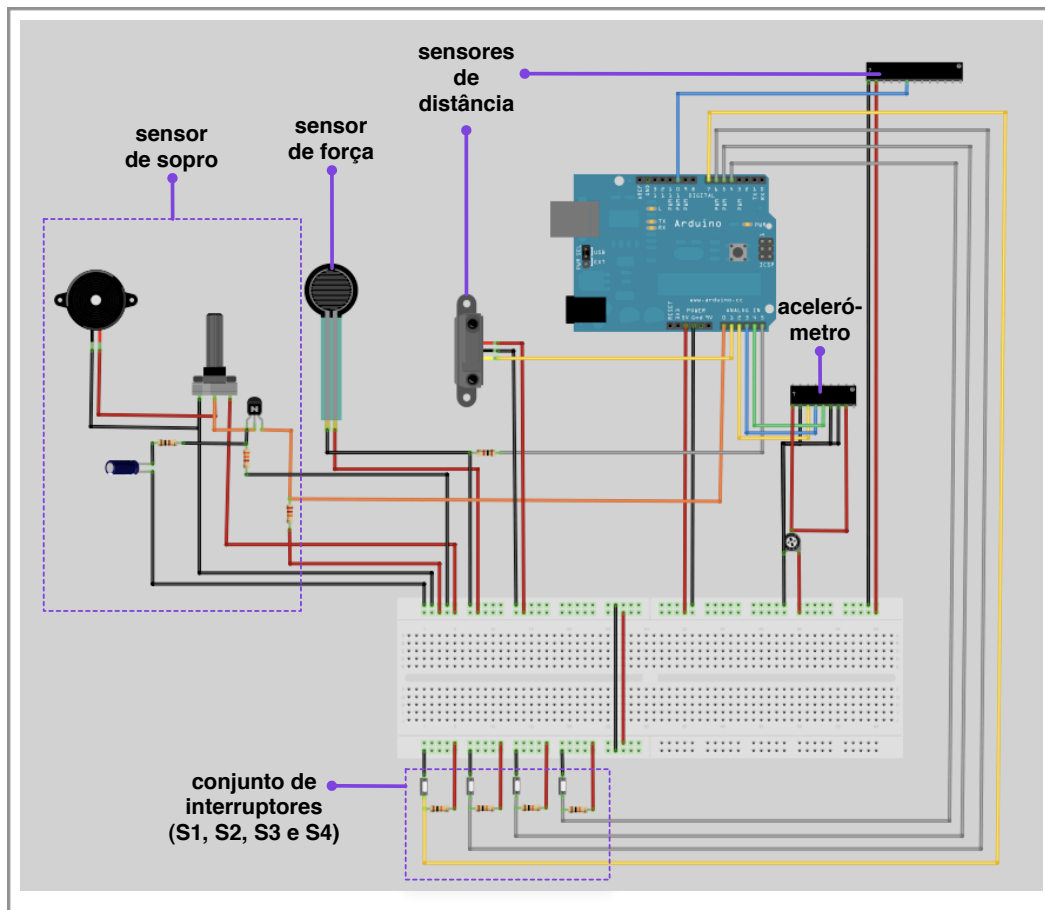


Fig. 2.20 — esquema de ligação à placa Arduino Bluetooth de todos os sensores e componentes (excepto o microfone).

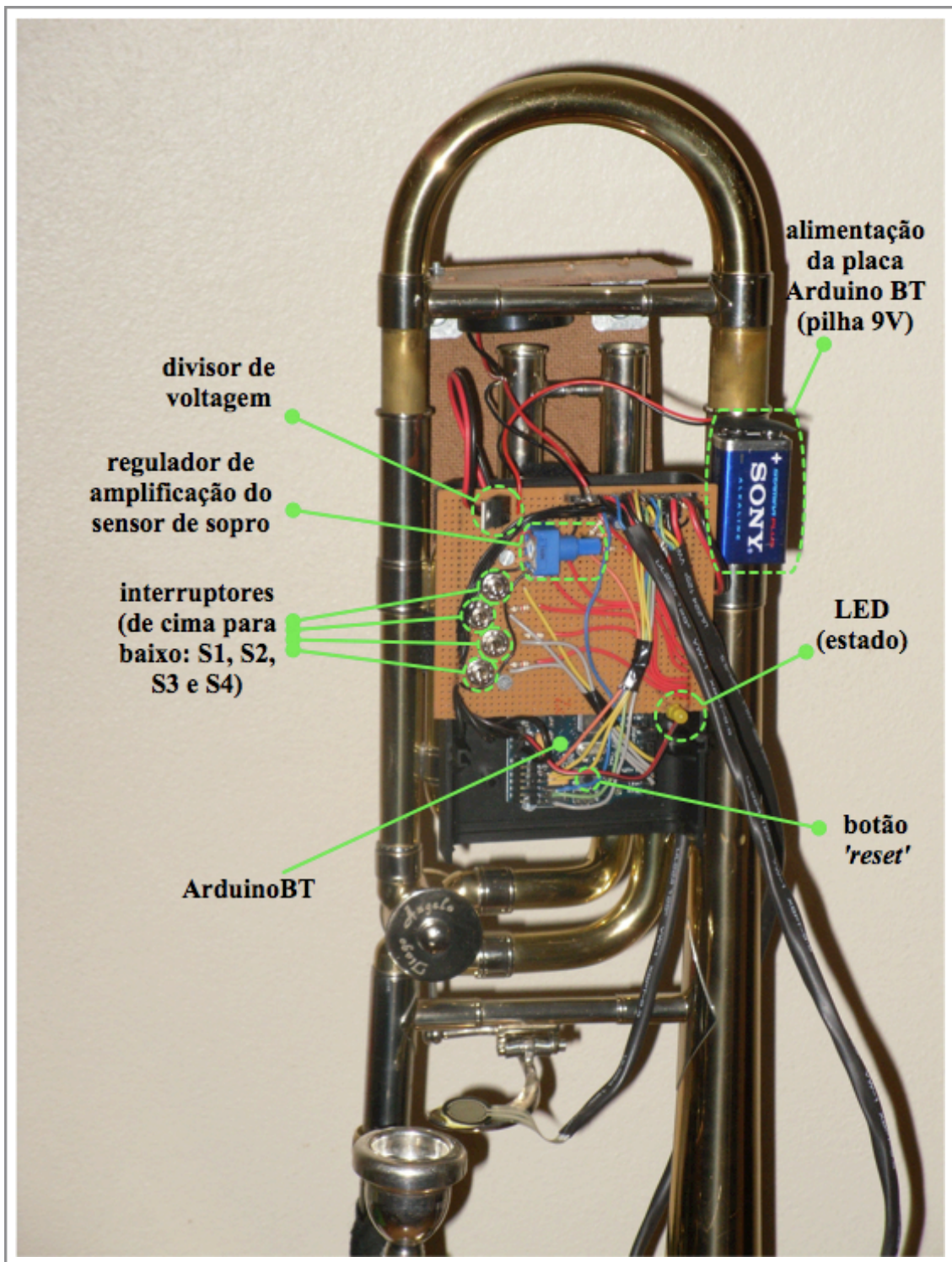


Fig. 2.21 — interruptores e outros componentes.

3. HTB1-s: Processador Sonoro

O processamento de sinais digitais (*DSP — Digital Signal Processing*) é uma técnica bastante vasta e que não se aplica exclusivamente à síntese sonora digital. Por norma este método está relacionado com o manuseamento de sinais discretos. Estes provêm de sinais contínuos, transformados por um conversor AD (analógico-digital) que depois de manuseados ou processados são novamente convertidos em sinais contínuos. (ver Fig. 3.1)

Este acontecimento está na base do processador sonoro HTB1-s. No contexto deste relatório um processador sonoro será qualquer sistema capaz de alterar, em tempo-real, as propriedades tímbricas de um determinado som. Para o desenvolvimento deste será necessário um ambiente de programação de síntese digital, que nos permita compor o algoritmo que, neste caso, irá produzir som a partir do processamento de um sinal sonoro de entrada.

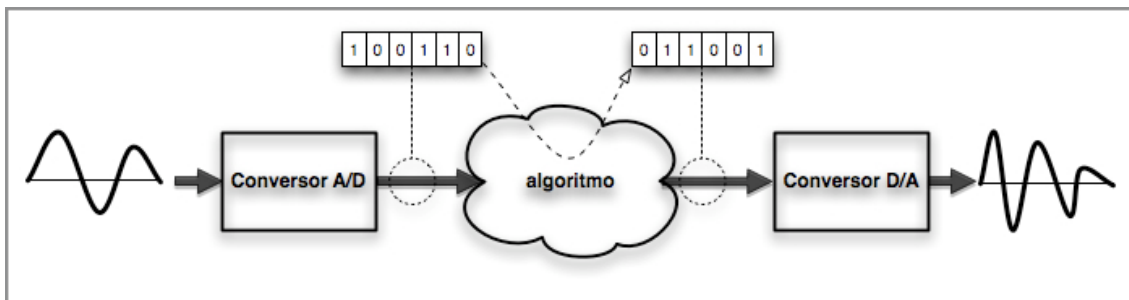


Fig. 3.1 — esquema base do processamento de sinais digitais (*DSP*)

3.1. Plataforma: ambiente gráfico de programação, Max/MSP

Ambientes gráficos de programação, como o próprio nome indica, são ferramentas que possibilitam a implementação de um programa através de uma interface visual. Isto é, o compilador não executa um código sob o formato de texto, mas sim um conjunto de blocos (ou objectos) interligados por cabos (denominados *patch cords*, em Max/MSP), visualizados graficamente. Estes têm um papel importantíssimo na sociedade ao aproximar as potencialidades de computação a curiosos, artistas, músicos e compositores de uma forma bastante intuitiva, quando comparados com as restantes linguagens de programação baseadas em texto (como por exemplo o CSound). [6]

“ O primeiro programa de compilação gráfica, Max, foi desenvolvido por Miller Puckette em 1988. (...) Assim que computadores capazes de processar sinais áudio em tempo-real se tornaram disponíveis, Puckette e David Zicarelli adicionaram o MSP(...)” (que significa *Max Signal Processing*) ao Max (agora Max/MSP). (PUCKETTE, Miller) [5]

A plataforma continuada por Miller, intitulada Pure Data, representa um óptimo concorrente à versão comercializada por Zicarelli (Max/MSP). No entanto a qualidade da documentação do Max/MSP foi preponderante para a escolha deste na realização do projecto.

Convém referir ainda que as ferramentas de programação descritas neste capítulo se referem apenas às que possuam a capacidade de síntese digital sonora.

3.2. Síntese sonora

A síntese sonora compreende um largo número de técnicas, estando fora do alcance deste projecto a utilização ou experimentação extensiva de todas estas. No entanto será pertinente salientar que a escolha final destes recaiu sobre motivos de gosto pessoal. E foram escolhidas após uma série de ensaios experimentais utilizando *software* como o Ableton Live e o Logic Pro, para além de um *patch* Max/MSP contendo várias formas de síntese e no qual se poderiam testar as ligações estabelecidas entre si.

As técnicas de síntese sonora escolhidas serão seguidamente descritas de uma forma não exaustiva de maneira a possibilitar ao leitor uma compreensão mínima dos métodos implementados no processador sonoro HTB1-s.

3.2.1. Síntese subtractiva

Este tipo de síntese foi o menos explorado na implementação do processador sonoro, no entanto convém referenciar que este método remete ao uso de filtros, algoritmos capazes de aumentar ou atenuar uma ou várias frequências da componente espectral de um sinal.

3.2.2. Síntese AM e RM

A síntese por modulação de amplitude (ou síntese AM — *Amplitude modulation*) é efectuada através de uma multiplicação de dois sinais (portador e modulador) no domínio do tempo, o que corresponde a uma convolução no domínio da frequência.

Se olharmos para o diagrama da figura 3.2 podemos notar que o oscilador em cima (modulador) é usado para modular a amplitude do sinal de entrada (portador). Aqui o sinal

modulador é unipolar, ou seja, em vez de conter valores entre -1 e +1 (como a maioria dos sinais áudio) este é restringido entre 0 e +1, como se lhe tivesse sido adicionada uma constante. [6]

Este tipo de modulação gera um par de bandas laterais para cada componente sinusoidal da moduladora, que se vão afastando da componente sinusoidal da portadora à medida que a frequência da moduladora aumenta, conforme podemos ver na figura 3.3. [6]

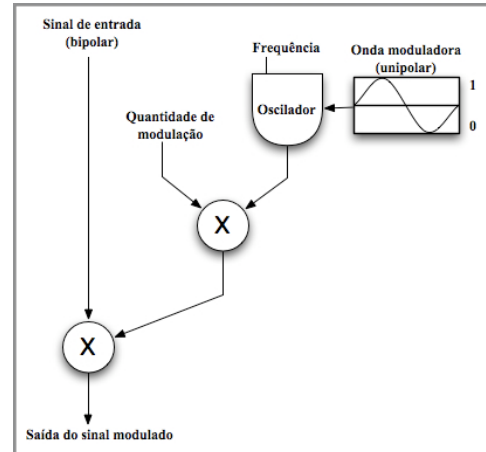


Fig. 3.2 — figura baseada num esquema de síntese AM básico, do livro “Computer Music Tutorial”.

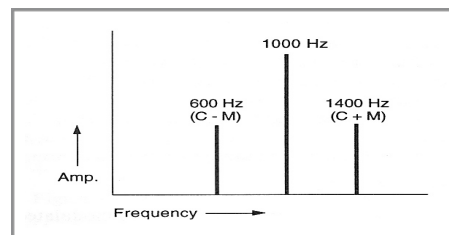


Fig. 3.3 — Espectro resultante da síntese AM de uma portadora (C) de 1KHz e uma moduladora (M) de 400Hz. [6]

A síntese RM (*Ring Modulation*) ou modulação em anel obteve o seu nome devido à sua aplicação analógica que consiste na utilização de díodos dispostos numa configuração em anel. [6]

Este tipo de modulação faz também parte da modulação de amplitude, sendo que a diferença entre o tipo de modulação abordado no anteriormente reside no facto de

o sinal modulador ser bipolar. Esta diferença reflecte-se no espectro, sendo que a frequência da portadora desaparece, restando apenas as bandas laterais, formadas do mesmo modo que em síntese *AM*. (ver fig 3.4 e 3.5)

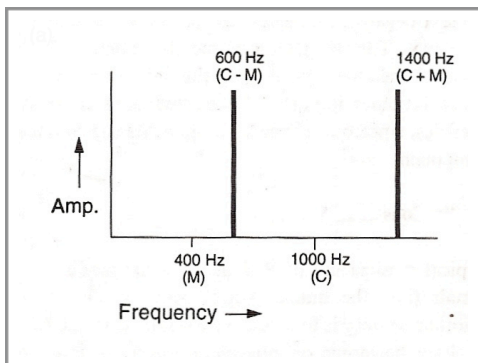


Fig. 3.4 — Espectro resultante da síntese RM de uma portadora (C) de 1KHz e uma moduladora (M) de 400Hz, verificando-se o aparecimento das bandas laterais, tal como na síntese *AM*, mas com o desaparecimento da componente sinusoidal da portadora. [6]

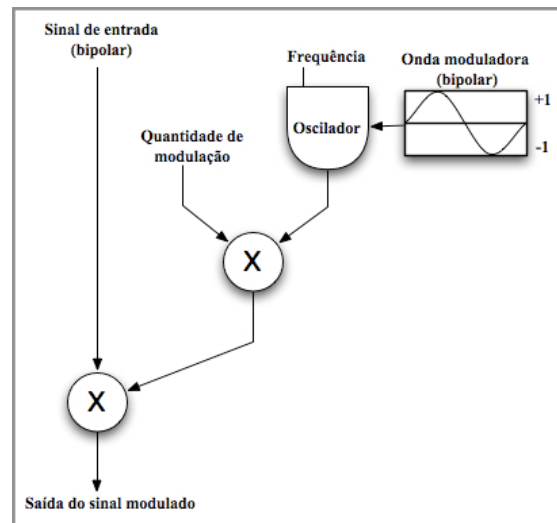


Fig. 3.5 — Esquema simples de síntese em anel. [6]

3.2.3. Flanging

Flanging é um efeito de *delay* de tempo variável, isto é, se tivermos um sinal de entrada e o modular-mos fazendo-o passar por um bloco de atraso (ou *delay line*) cuja duração é variável (ver Fig 3.6). [6]

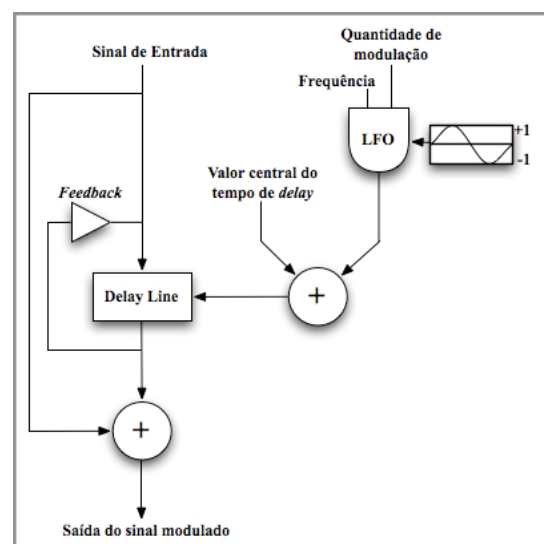


Fig. 3.6 — Esquema base de um *flanger*. O *LFO* (*Low Frequency Oscillator* — oscilador de baixa frequência) é responsável por introduzir a variação do tempo de atraso (*delay*). [6]

3.2.4. As Transformadas de Fourier

Em 1822 o Barão de Fourier, Jean Baptiste Joseph, publicou uma teoria na qual “uma forma de onda periódica (onda que se repete num intervalo de tempo regular) pode ser decomposta em combinações em simples combinações de ondas sinusoidais de diferente amplitude, frequência e fase”(ROADS, Curtis). E é portanto uma método bastante útil, não só na análise sonora como em processos de ressíntese. [6]

Esta teoria, a análise de Fourier, pode tomar várias formas consoante o tipo de sinais que se pretendem analisar. De todas as séries de Fourier interessa-nos referir para este relatório apenas a variante *STFT (Short-Time Fourier Transform)*.

Esta transformada de Fourier permite-nos, através de janelas de análise (*analysis windows*) dividir um sinal contínuo, representado-o como uma série de blocos contendo informação espectral (frequência e amplitude) e cuja sequência pretende representar o sinal original. Exercido sobre sinais discretos, este algoritmo tem como resultado um “espectro discreto de frequências, isto é, a medida de energia num conjunto de frequências igualmente espaçadas” (ROADS, Curtis).

3.2.4.1 Convolução

Resumidamente a Convolução é um processo de multiplicação de dois sinais no domínio da frequência. O efeito resultante é um cruzamento espectral (amplitude e frequência) dos dois. O diagrama da figura 3.7 representa a implementação deste algoritmo. Do qual de dois sinais amostrados X e Y:

- $X(A) = a * ib$
- $Y(A) = c * id$

Se obtém o terceiro sinal:

$$\bullet Z(A) = X(A) * Y(A) = (a * ib) * (c * id) \\ = ac -bd + i(ad + bc)$$

O método de coordenadas aqui representado é o cartesiano, uma vez que foi o algoritmo usado, dado ser menos exigente do ponto de vista computacional (Dobrian, 2006).

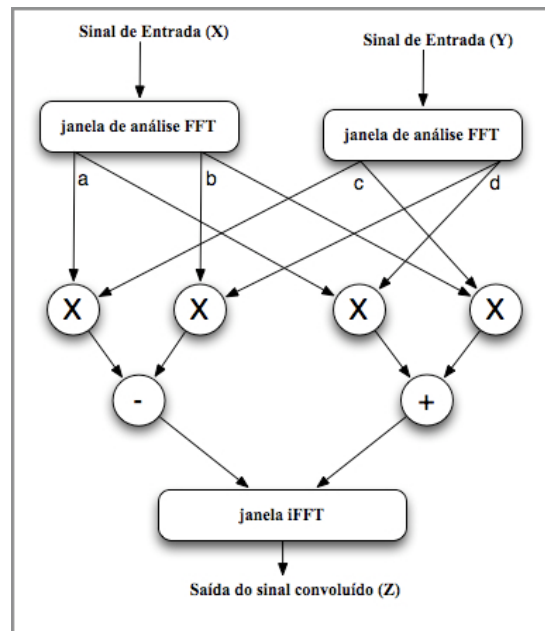


Fig.3.7 — diagrama do processo de convolução cartesiano

3.2.4.2. *Frequency-domain pitch shifting*

Utilizando as transformadas de Fourier é possível obter a alteração da altura (*pitch shifting*) de um som em tempo-real (com um atraso temporal igual à janela de análise, normalmente curto), mantendo a componente temporal intacta, ao contrário dos métodos de *pitch shifting* no domínio temporal, que afectam a frequência e a duração do sinal. [6]

3.2.4.3. Time-Freeze

Este processo tem a característica de “congelar” um sinal temporalmente, o que “envolve a extensão do espectro de magnitude de um bloco *STFT* indefinidamente, propagando as fases baseando-se numa estimativa instantânea de frequência” (VICKERS, Earl et al.)

Este processo de repetição e conseguido através do uso de *feedback delays* (blocos de atraso realimentados), sendo que se obtém uma congelação infinita quando o valor de *feedback* está à unidade (ver Fig. 3.8). Aqui convém salientar que o método de coordenadas polar é usado para se puder trabalhar com as variáveis de fase e magnitude. [8]

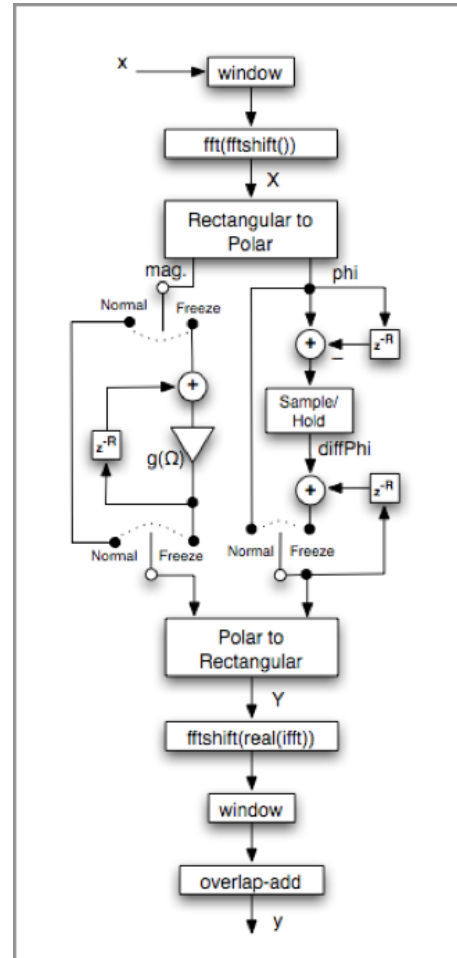


Fig. 3.8 — algoritmo básico de Congelamento -Temporal. [8]

3.2.5. Síntese Granular

A síntese granular aborda o som de um ponto de vista atômico, no sentido em que este é tratado como uma partícula (ou grão — *grain*) normalmente com uma duração compreendida entre 1ms e 100ms (duração mínima para a percepção de duração, frequência e amplitude). Este tipo de síntese consiste depois na constituição de várias camadas destas partículas, podendo todas elas possuírem parâmetros, como duração, frequência e amplitude diferentes, possuindo portanto um potencial tímbrico enorme. [6]

Por norma é realizada através de uso de vários osciladores que têm um *buffer* (memória que neste caso armazena um ficheiro de áudio) e são modulados por uma envolvente. (ver Fig.3.9) [6]

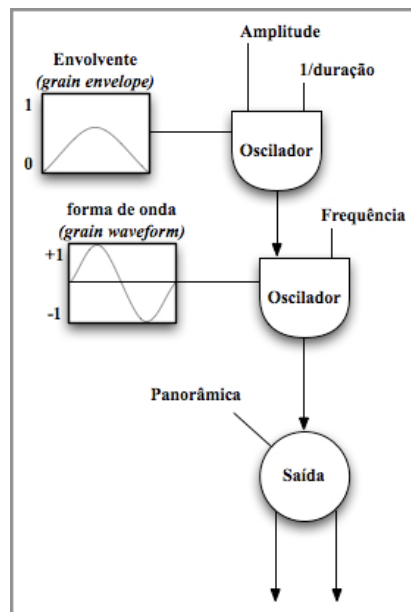


Fig. 3.9 — figura baseada num esquema de síntese granular simples do livro “Computer Music Tutorial”.

3.3. HTB1-s: *software*

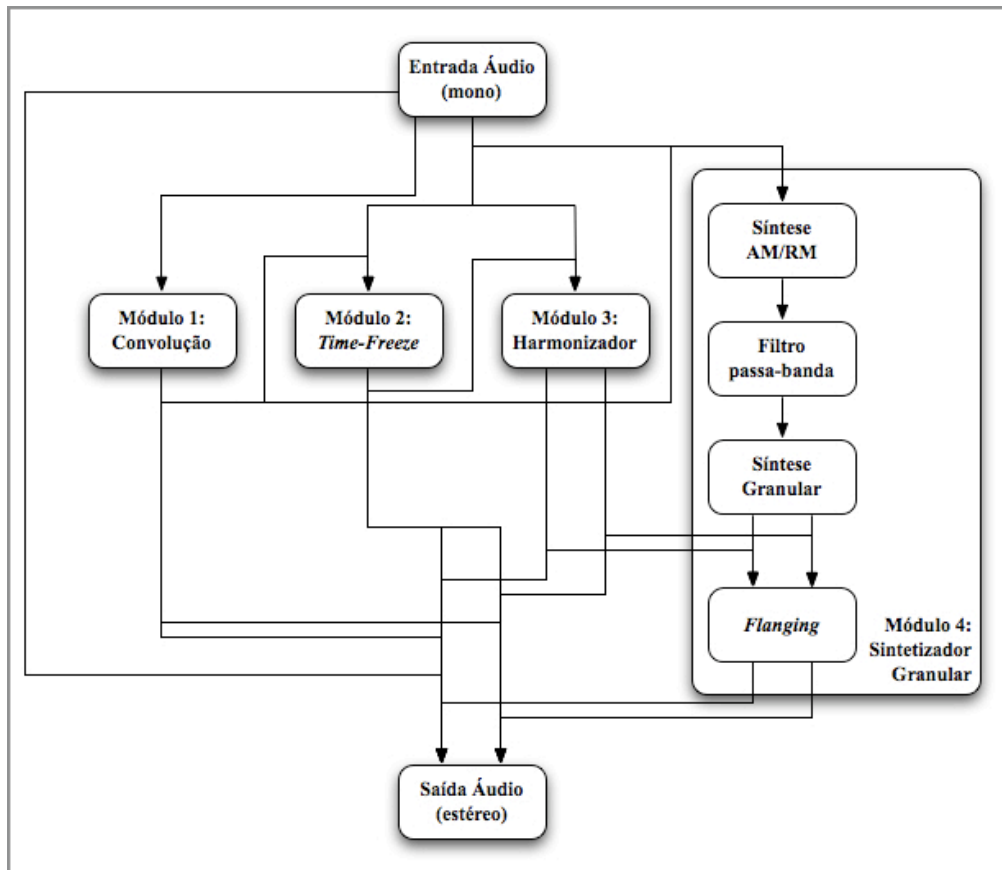


Figura 3.10 — HTB1-s: diagrama das conexões entre módulos

Estes módulos (ver figura em cima) no seu todo formam o processador sonoro constituinte do instrumento aumentado HTB1.

Estes possuem parâmetros comuns como uma unidade de ganho, uma unidade de mistura de sinal (*mix: dry/wet*) e duração de *fade in* e *fade out* de activação dos módulos (*slew time*), bem como a possibilidade de gravar vários parâmetros (utilizando o objecto do Max/MSP *ptrstorage*) para seja possível mudar a configuração do processador para, por exemplo, a execução de diferentes peças musicais.

3.3.1. Módulos

Módulo 1: Convolução

Características:

- O sinal contendo a informação relativa à frequência é dado por um *sampler*;

- O sinal contendo a informação relativa à amplitude é o sinal de entrada (proveniente do microfone colocado no trombone);

- algoritmo para detecção de amplitude do sinal de entrada e correspondência proporcional da posição de leitura do *sampler* (responsável pelo conteúdo espectral de frequência).

Parâmetros: (ver Fig. 3.11)

- *GAIN* — ganho do *sampler*;
- *INPUT THRESHOLD* — limiar de amplitude, a partir do qual o algoritmo descrito é activado;
- *FREQUENCY* — velocidade de leitura do *sampler*;
- *MULTIPLYING FACTOR* — factor de multiplicação do valor de amplitude do sinal de entrada direccionado para o algoritmo de detecção.

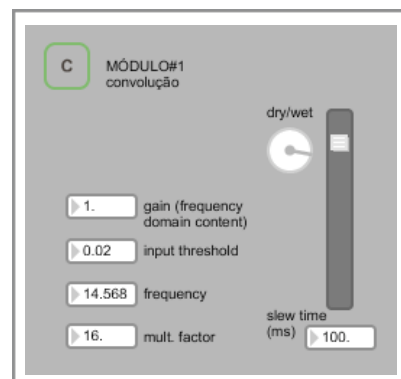


Fig. 3.11 — Módulo de convolução.

Módulo 2: *Time-Freeze*

Características:

É aplicada uma envolvente ao sinal de entrada que é activada quando este atinge um certo limiar. No sentido de tentar solucionar o problema causado pela introdução de ruídos de fundo que são depois “congelados” no tempo formando ruído tonal. Não tendo no entanto surtido os resultados desejados. [6]

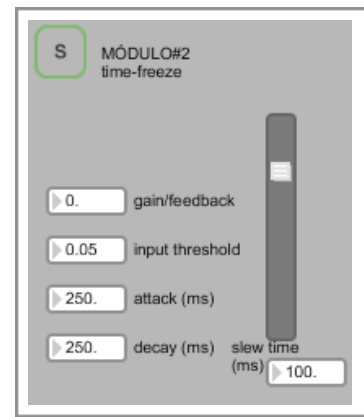


Fig. 3.11 — Módulo de congelação temporal.

Parâmetros: (ver Fig. 3.12)

- *GAIN/FEEDBACK* — ganho da realimentação dos blocos de atraso (*delay lines*);
- *INPUT THRESHOLD* — limiar de amplitude do funcionamento da envolvente;
- *ATTACK* — tempo de ataque, em milisegundos, da envolvente;
- *DECAY* — tempo de decaimento (ms) da envolvente.

Módulo 3: Harmonizador

Características:

- 4 vozes (5 se considerar-mos o sinal de entrada como sendo uma voz do acorde);

- Algoritmo que recorre a uma base de dados contendo cerca de 60 acordes. Cada acorde foi submetido a uma análise prévia, calculando um valor arbitrário de tensão.

Este valor, compreendido entre 0 e 5, é uma média do valor de tensão de cada nota constituinte do acorde em relação à tónica.

(ver Fig.). Os acordes são escolhidos aleatoriamente, dentro das possibilidades que enquadram um determinado valor de tensão.

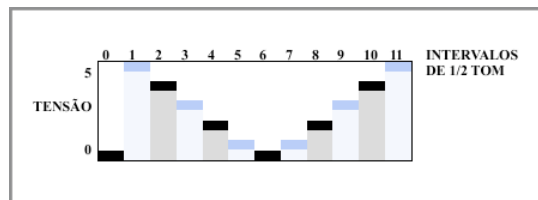


Fig. 3.13 — Valor de tensão correspondente a cada intervalo (0=uníssono; 11= 7ª maior).

- A cada voz do harmonizador é aplicada uma oscilação de amplitude, compreendida entre os 0 e os 16Hz (*tremolo*), na tentativa de simular as diferenças de amplitude que ocorrem na realidade quando juntamos por exemplo um *ensemble* de trombones. O que adiciona um diferença significativa na vivacidade do som resultante.

Parâmetros: (ver Fig. 3.12)

- *CHORD TENSION* — valor de tensão do acorde a ser gerado;
- *GLIDE* — tempo, em milisegundos, em que é feita a mudança de intervalos quando é gerado um novo acorde;
- *TRANSPOSITION INCREMENT* — incrementação aplicada à altura (intervalo) aplicado a todas as quatro vozes.

Módulo 4: Sintetizador Granular

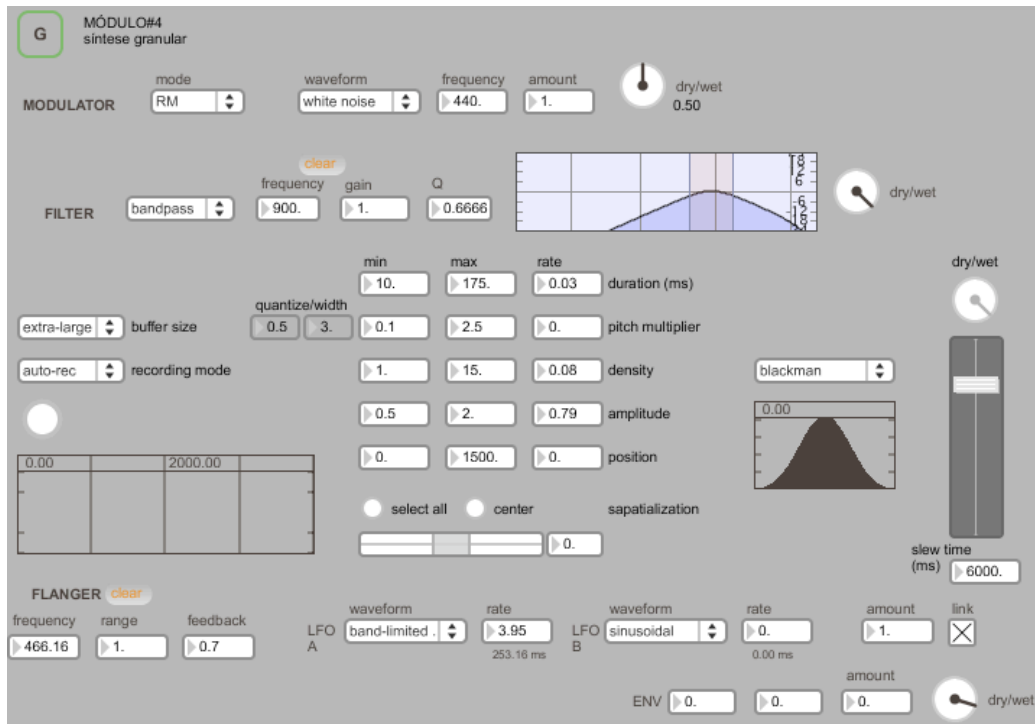


Fig. 3.14 — Módulo do sintetizador granular.

Características: (ver Fig. 3.14)

- Modelo de síntese granular baseado no *patch* de Nobuyasu Sakodna;
- Foram adicionados osciladores aos parâmetros do *patch* mencionado em cima;
- Foi abordada uma técnica semelhante à síntese assíncrona adicionando o parâmetro densidade, ao contrário da síntese síncrona realizada no *patch* de Sakodna;
- Foi implementado um algoritmo que possibilitou a leitura e gravação do mesmo *buffer* em tempo-real. Este tinha como função evitar que o posicionamento das partículas/grãos no *buffer* não coincidisse com a posição de gravação neste, para evitar
- À entrada do sinal estão um submódulo de síntese *AM/RM* e de síntese subtrativa (filtro *biquad*-). E à saída encontra-se um *flanger*. Estas adições foram realizadas uma vez que o efeito de síntese granular aplicado ao som limpo de um trombone não proporciona resultados muito interessantes.

Modulator

Parâmetros: (ver Fig. 3.14)

- *MODE* — tipo de síntese (*AM* ou *RM*);
- *WAVEFORM* — forma de onda usada no oscilador modulador;
- *FREQUENCY* — frequência dos oscilador modulador;
- *AMOUNT* — quantidade de modulação.

Filter

Parâmetros: (ver Fig. 3.14)

- *MODE* — tipo de filtro;
- *FREQUENCY* — frequência de corte;
- *GAIN* — ganho;
- *Q* - largura de banda;

Sintetizador granular

Parâmetros: (ver Fig. 3.14)

- *BUFFER SIZE* — tamanho do *buffer* acedido pelo sintetizador e para o qual é gravado o som proveniente do microfone colocado no trombone;

- *RECORDING MODE* — modo de gravação (*loop*- gravação contínua, *rec 1x* - grava apenas uma vez, *auto-rec* - grava quando o sinal de entrada ultrapassa um determinado *threshold*);
- *DURATION* — duração mínima, máxima e frequência (*rate*) de variação em *Hertz* do tamanho (em milisegundos) das partículas/grãos;
- *PITCH MULTIPLIER*— altura mínima, máxima e frequência (*rate*) de variação em *Hertz* da altura das partículas, para além de quantização (*quantization*) e largura (*width*) da quantização efectuada à velocidade de leitura das partículas;
- *DENSITY* — densidade mínima, máxima e frequência (*rate*) de variação em *Hertz* da densidade de partículas;
- *AMPLITUDE* — amplitude mínima, máxima e frequência (*rate*) de variação em *Hertz* do amplitude (em milisegundos) das partículas;
- *POSITION* — posição mínima, máxima e frequência (*rate*) de variação em *Hertz* da posição de leitura (em milisegundos) no *buffer*;
- *SPATIALIZATION* — localização (esquerda-direita) da propagação das partículas, bem como a sua frequência de variação (*rate* em *Hz*);
- *ENVELOPE* — forma de onda usada para a envolvente das partículas.

Flanger

Parâmetros: (ver Fig. 3.14)

- *FREQUENCY* — frequência ou inverso da duração do tempo dos blocos de atraso;
- *RANGE* — alcance da variação da frequência;
- *FEEDBACK* — ganho da realimentação introduzida nos blocos de atraso;
- *WAVEFORM* — forma de onda do oscilador de baixa frequência;

- *RATE* — frequência do oscilador;
- *AMOUNT* — quantidade de modulação provocada pelos osciladores de baixa frequência;
- *LINK* — sincronização dos osciladores A e B usados para o canal esquerdo e direito respectivamente;
- *ENVELOPE ATTACK* — tempo de ataque da envolvente aplicada à frequência do *flanger*;
- *ENVELOPE DECAY* — tempo de decaimento da envolvente aplicada à frequência do *flanger*;
- *ENVELOPE AMOUNT* — quantidade de modulação aplicada pela envolvente.

3.3.1. Mapeamento

Neste subcapítulo são apresentados, sob o formato de tabela (ver em baixo), os mapeamentos (ver Tabela 1.5) dos dados provenientes do controlador gestual (HTB1-h) para o processador sonoro(HTB1-s). No âmbito do instrumento aumentado (HTB1) o mapeamento representa portanto o elo de ligação entre interface e instrumento.

Tabela 1.5 - Mapeamentos do controlador gestual para os módulos do processador sonoro.

Módulo	Parâmetro	Sensor	Mapeamento
Módulo 4: Sintetizador Granular	frequência da variação de duração das partículas	sensor de sopro	a intensidade do sopro está directamente relacionada com um aumento desta frequência
“	panorâmica	“	um aumento de intensidade provocado no sensor equivale a uma abertura panorâmica (esquerda-direita) proporcional
“	frequência da variação de posição de leitura do <i>sample</i> (captado ao vivo)	sensor de distância da vara	quanto mais perto a vara estiver da 1ª posição menor será a variação da posição de leitura
“	frequência da variação da densidade de grãos	“	um aumento de intensidade corresponde a um aumento desta frequência
“	captação do <i>input</i> áudio para o <i>buffer</i>	microfone	quando o valor de amplitude enviado ultrapassa um determinado threshold o <i>buffer</i> é actualizado

Módulo	Parâmetro	Sensor	Mapeamento
Módulo 4: <i>Flanging</i>	mistura entre o sinal processado e não-processado (proveniente do sintetizador granular)	“	maior intensidade provocada no sensor proporciona um aumento do sinal não-processado na mistura
“	âmbito de frequência	“	quanto maior for a intensidade do sopro maior é o âmbito
“	frequência	sensor de distância da vara	a distância da vara captada por estes sensores é mapeada para que na 1ª posição corresponda à freq. de um Sib3 e na 7ª posição a um Mi3
Módulo 3: Harmonizador	geração de acordes	FSR	um toque no sensor corresponde à geração de um acorde, este módulo apenas é activado quando o polegar estiver repousado no sensor
“	valor de tensão do acorde a ser gerado	sensor de distância da vara	os valores de tensão (0..5) correspondem à posição em que se encontra a vara (1ª..6ª)
Módulo 2: <i>Time-Freeze</i>	feedback	acelerómetro	uma aceleração positiva no eixo do y corresponde a um aumento de feedback e vice-versa
Módulo 1: Convolução	frequência do <i>sampler</i>	sensor de ultrasons/infravermelhos	as posições captadas da vara (1..7) são mapeadas para que a frequência do <i>sampler</i> vá de Sib a Mi, sem interpolação
“	posição de leitura do <i>sampler</i>	microfone	a posição de leitura inversamente proporcional à intensidade sonora captada pelo sensor

4. Conclusão

HTB1-h

De um modo geral os objectivos propostos para a construção de um controlador gestual apropriado para um trombone de varas foram alcançados. A aquisição de dados gestuais introduzidos pelo instrumentista é bastante vasta quando comparada com trombones aumentados existentes (ver Anexos) e a ligação sem-fios entre controlador e computador veio trazer grande liberdade de movimentação quando comparado com alguns instrumentos aumentados. No entanto a ergonomia e facilidade de montagem/desmontagem do controlador podem sempre ser melhorados.

HTB1-s

Os resultados sonoros obtidos pelo processador são interessantes e proporcionam variadas possibilidades do ponto de vista composicional, no entanto nota-se que este poderia ter alguma interactividade para além da introduzidos pelos gestos físicos do instrumentista. A aquisição de dados de forma indirecta poderia portanto ter sido mais explorada, bem como a introdução de algoritmos inteligentes capazes de compreender os dados gestuais percebidos.

HTB1

De um ponto de vista geral ficou a noção de que o controlador HTB1-h será uma plataforma estável para o desenvolvimento de várias aplicações. Enquanto que a aplicação HTB1-s poderá vir a tomar várias formas, no sentido em se poderão programar vários processadores distintos para o uso do controlador gestual.

A intenção de alargar as características sonoras típicas de um trombone de varas convencional foi conseguido. Trazendo um leque de novas possibilidades para este instrumento, que poderá servir de impulso para a composição de peças musicais destinadas a este. A inserção deste novo instrumento em novos *ensembles* (como por exemplo um *ensemble de laptops*) parece agora uma opção a ter em conta.

Em soma, o repertório e formações passíveis de integrar o hipertrombone são agora maiores do que as do trombone de varas convencional.

5. Anexos

5.1. Exemplos de hiperinstrumentos

Flautas transversais aumentadas

•Descrição: “ A flauta transversal é talvez o instrumento acústico mais vulgarmente aumentado através do uso de sensores. Diferentes tecnologias de detecção (ou sensores) foram usados, dependendo dos objectivos de cada projecto, permitindo uma grande variedade de estratégias de controlo.

As adições mais comuns em flautas acústicas são sensores nas chaves de maneira a detectar as dedilhações, tal como na flauta MIDI desenvolvida no IRCAM em meados de 1980. Contudo, a dedilhação é apenas uma das possíveis variáveis de controlo que podem ser obtidas aumentando uma flauta com sensores. Outras variáveis podem também ser detectadas, tal como a inclinação do instrumento, a rotação em torno do seu eixo, o seu peso nos dedos de suporte, a sua distância de um ponto de referência, bem como a informação acerca da coluna de ar na embocadura.

Podemos ainda reparar que, sendo a flauta um instrumento monofónico, a identificação de altura pode ser conseguida através de uma aquisição indirecta, isto é, através da análise do som produzido. Contudo, uma vantagem da aquisição directa através de sensores nas chaves é que a dedilhação pode ser detectada mesmo quando a flauta não está a ser tocada. outros controlos de informação não podem ser obtidos directamente através da análise do som, tal como a inclinação, que pode ser obtido com sensores. Os problemas de usar sensores, neste caso, incluem a manutenção e a obstrução. Os sensores e os cabos podem não funcionar da melhor maneira e os instrumentistas podem ter que aprender como executar os sensores ou adaptarem técnicas performativas previamente estudadas de maneira a controlarem as variáveis desejadas.” [8]

Exemplos de flautas aumentadas:

- IRCAM
- LMA
- McGill University
- hyper-flute

“Augmented trumpet”

- Autores: Perry Cook e Dexter Morrill
- Data e Local de desenvolvimento: 1989, Colgate University
- Descrição: “Perry Cook e Dexter Morrill construíram dois instrumentos aumentados baseados numa trompete e num trombone de pistões. Sensores de pressão (microfones) foram colocados no bocal da trompete e na surdina de trombone para detecção de altura (‘pitch’) e nas campânulas para acompanhar a envolvente da amplitude. Interruptores e potenciômetros lineares eram operados pelos dedos, estando montados perto dos pistões. Três interruptores colocados por baixo dos pistões permitem detectar as suas posições (abertos ou fechados). Estes eram accionados pressionando os pistões ou os próprios interruptores. Uma pedaleira era também usada como controlador extra.” [9]

“Mutant Trumpet”

- Autor: Ben Neil
- Data e Local de desenvolvimento: 1992, STEIM
- Descrição: “A ‘*Mutant Trumpet*’ é um instrumento híbrido electroacústico com três campânulas, dois conjuntos de pistões/válvulas e uma vara de trombone. O som é convertido por um *pickup* no bocal para informação MIDI através de um conversor

MIDI. Existem três botões localizados próximo dos pistões, bem como quatro controladores MIDI contínuos sob a forma de sensores de pressão e potenciômetros. O volume acústico da *'Mutant Trumpet'* é também usado como outro controlador MIDI. A interface do computador da *'Mutant Trumpet'* permite activar e manipular sequências, manipular amostras de áudio em tempo-real e controlar luz, *slides* e projecção de vídeo.” [10]

“Hyper-strings”

- Descrição: “ Exemplos de instrumentos de cordas incluem o violino IR de Peter Beyls desenvolvido em 1990; a família de hiperinstrumentos desenvolvidos no MIT por Tod Machover e colaboradores (hiper-violoncelo, hiper-violão, hiper-violino); e o violino aumentado, recentemente desenvolvido no IRCAM por Emmanuel Fléty e colaboradores.” [11]

“Low Brass” e “Meta-trombone”

- Autores: Nicholas Collins e Mark Bromwich (respectivamente)
- Data de desenvolvimento: 1991 e 1997 (respectivamente)
- Descrição: Ambos os controladores eram baseados em trombones modificados pela inserção de vários sensores mas não foram desenhados para proporcionar aos instrumentistas um trombone aumentado. [3]

“Meta-Saxophone”

- Autor: Mathew Burtner
- Data e Local de desenvolvimento: 1997, Stanford university
- Descrição: “ saxofone tenor (...) usando sensores de força, *triggers* e um acelerómetro de 2 eixos. Os sensores de força/flexão estavam localizados nas chaves (Si, Lá, Sol, Fá, Mi e Ré) e nos dois apoios para os polegares. Os *triggers* estavam localizados na campânula e por baixo dos apoios para os polegares. O acelerómetro de 2 eixos estava localizado no corpo do instrumento de maneira a detectar os movimentos da campânula da esquerda para a direita e de cima para baixo. Os sinais dos sensores eram enviados para um micro-controlador Basic Stamp que enviava mensagens de controlo MIDI.” [12] [13]

“SBass”

- Autor: Curtis Bahn
- Data de desenvolvimento: 2001
- Descrição: “ ‘SBass’ (*sensor bass*) consistia num baixo eléctrico de 5 cordas preparado com vários sensores, incluindo um *touchpad* montado no braço, botões, sensores de tacto (posição), sensores de força (*FSR*), potenciómetros rotativos e um acelerómetro dois eixos (...) vários *pickups* foram também adicionados, tal como um microfone de contacto colocado debaixo das cerdas na ponta do arco.” [14] [15]

“Augmented trombone”

- Autores: Serge Lemouton, Marco Stroppa, Benny Sluchin
- Data e Local de desenvolvimento: 1992, IRCAM (Paris, França)
- Descrição: Consiste num trombone aumentado através de sensores de infra-vermelhos que detectam o posicionamento da vara. Através de *hardware* (WiSe Box, também desenvolvido no IRCAM) capaz de digitalizar os dados provenientes de sensores, estes são transformados em mensagens OSC (*Open Sound Control*) que são enviadas através de uma rede sem-fios. [16]

6. Referências

Livros

Banzi, Massimo. “Getting Started with Arduino”. Sebastopol, U.S.A.: Make:Books, 2008. Páginas: V-VII, 1-45.[1]

Cook, Perry. “Real Sound Synthesis for Interactive Applications”. Wellesley, Massachusetts: A K Peters, 2002.

Guion, David M. “The Trombone - History and Music, 1697-1811”. Amsterdão, Holanda: Gordon and Breach Publishers, 1998.

Herbert, Trevor. “The Trombone”. Londres: Yale university Press, 2006.

Henrique, Luís L. “Acústica Musical”. Lisboa, Portugal: Fundação Calouste Gulbenkian, 2006.

Henrique, Luís L. “Instrumentos Musicais”. Lisboa, Portugal: Fundação Calouste Gulbenkian, 2006. Páginas: 12-19, 231-242, 321-376. [2]

Kirk, Ross. “Digital Sound Processing for Music and Multimedia”. Focal Press, 1999. [3]

O’Sullivan, Dan e Igoe, Tom. “Physical Computing”. Boston: Thomson Course Technology, 2004. Páginas: 127, 217-248 [4].

Padilla, António J. G. “Electrónica Analógica”. Lisboa, Portugal: McGraw-Hill, 1993.

Puckette, Miller. “The Theory and Technique of Electronic Music”. World Scientific Publishing, 2007. Página: IX. [5]

Roads, Curtis. “The Computer Music Tutorial”. Cambridge, Massachusetts: MIT Press, 1996. Páginas: IX-XII, 87, 204, 216-223, 436-438, 1073-1106. [6]

Wanderley, Marcelo e Miranda, Eduardo. “New Digital Musical Instruments”.
Middletown, Wisconsin: A-R Editions, Inc. , 2006. Páginas: XV-XXII, 1-59, 103-153.
[7]

Artigos

Cook, Perry R. - TBone: An Interactive WaveGuide Brass Instrument - Synthesis Workbench for the NeXT Machine. Stanford Center for Computer Music Research in Music and Acoustics.

Collins, Nicolas - Low Brass: The Evolution of TRombone-Propelled Electronics. Leonardo Music Journal, Vol.1, No. 1, pp. 41-44 (1991). Grã-Bretanha.

Farwell, Neal - Adapting the trombone: a suite of electro-acoustic interventions for the pipece Rouse. Procedimentos da Conferência Internacionatl NIME (New Interfaces for Musical Expression). França: Paris (2006).

Fléty, Emmanuel - The WiSe Box : a Multi-performer Wireless Sensor Interface using WiFi and OSC. Performance Arts Technology Gropu, Ircam. França:Paris.

Henriques, Tomás - Double slide controller. Buffalo State College. U.S.A.: Buffalo (2009).

Henriques, Tomás - META-EVI - Innovative Performance Paths with a Wind Controller. Faculdade de Ciências Sociais e Humanas, Universidade Nova de Lisboa. Portugal: Lisboa.

Lemouton, Serge [et al.] - “Using the augmented trombone in “I will not kiss your f.ing flag” Procedimentos da Conferência Internacionatl NIME (New Interfaces for Musical Expression). França: Paris (2006).

Quitin-Palacio, Cléo - The Hyper-Flute. Procedimentos da Conferência Internacionatl NIME (New Interfaces for Musical Expression). Canadá: Montreal (2003).

Scavone, Gary e Silva, Andery da - Frequency Content of Breath Pressure and Implications for Use in Control. Procedimentos da Conferência Internacionatl NIME (New Interfaces for Musical Expression). Canadá: Vancouver (2005).

Schiesser, Sébastien [et al.] - On making and playing an electronically-augmented saxophone. Procedimentos da Conferência Internacionatl NIME (New Interfaces for Musical Expression). França: Paris (2006).

Vickers, Earl [et al.] - Frequency Dmain Artificial Reverberation using Spectral Magnitude Decay.[8]

Web

<http://www.arduino.cc/playground/CourseWare/AnalogInput> (Consultado a 31 de Janeiro de 2010) [9]

http://spreadsheets.google.com/pub?key=peZeSrni_qmjx6qzVTuHzNA (Consultado a 18 de Fevereiro de 2010) [10]

<http://arduino.cc/en/Reference/PulseIn> (Consultado a 9 de Fevereiro de 2010) [11]