

GuitarAMI: um Instrumento Musical Aumentado que Transpõe Restrições Intrínsecas do Violão

Eduardo Aparecido Lopes Meneses^{1*}, José Eduardo Fornari Novo Junior¹

¹NICS - Núcleo Interdisciplinar de Comunicação Sonora
Universidade Estadual de Campinas

Rua da Reitoria, 165 - Cidade Universitária “Zeferino Vaz” – 13083-872 Campinas, SP

eduardo@edumeneses.com, tutifornari@gmail.com

Abstract. *Despite its huge spread and popularity, the acoustic guitar has intrinsic and characteristic restraints of its physical structure. This article discusses the construction and use of an augmented musical instrument (AMI) entitled GuitarAMI. While most of the digital musical instruments are designed to allow control of sound synthesis and transformations during the instrumental performance, this AMI was created in order to modify some of the intrinsic characteristics of the acoustic guitar, commonly considered restrictive in guitaristic practice.*

Resumo. *Apesar da sua enorme disseminação e popularidade, o violão apresenta restrições intrínsecas e características da sua estrutura física. Este artigo discute a construção e utilização de um instrumento musical aumentado (AMI) intitulado GuitarAMI. Enquanto grande parte dos instrumentos musicais digitais foram criados para possibilitar o controle de síntese e transformação sonora durante a performance instrumental, este AMI foi criado com o objetivo de modificar algumas das características intrínsecas do violão, comumente consideradas restritivas na prática violonística.*

1. Introdução

A partir da segunda metade do século XX houve uma verdadeira revolução tecnológica, ocorrida com a chegada de diversos dispositivos e plataformas eletrônicas que, além de possuírem um baixo custo, promoveram formas facilitadas de construção (*hardware*) e programação (*software*), possibilitando a utilização destes recursos em atividades artísticas antes pouco beneficiadas pelos avanços tecnológicos, como a performance musical ou a criação de instalações artísticas. Entre outros, podemos citar como exemplos de implementações utilizando software-livre e interfaces gestuais de baixo custo, as descritas em [Aliel and Fornari, 2013] e [Fornari, 2012]. Esta facilidade de programação é de fundamental importância para diminuir a curva de aprendizagem sobre determinado dispositivo e o tempo investido em sua criação, como afirma Perry Cook:

A performance musical com novas interfaces [instrumentos] de computador é atualmente corriqueira como resultado da disponibilidade de dispositivos eletrônicos de baixo custo, novos sensores para capturar diversos parâmetros como intensidade [do movimento] e localização e novos programas de computador para síntese e manipulação do som em tempo real. [Cook, 2001, p. 1; Tradução e comentários nossos.]¹.

*Bolsista CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior).

¹Musical performance with entirely new types of computer instruments is now commonplace, as a result of the availability of inexpensive computing hardware, of new sensors for measuring physical parameters such as force and position, and of new software for real-time sound synthesis and manipulation [Cook, 2001, p. 1].

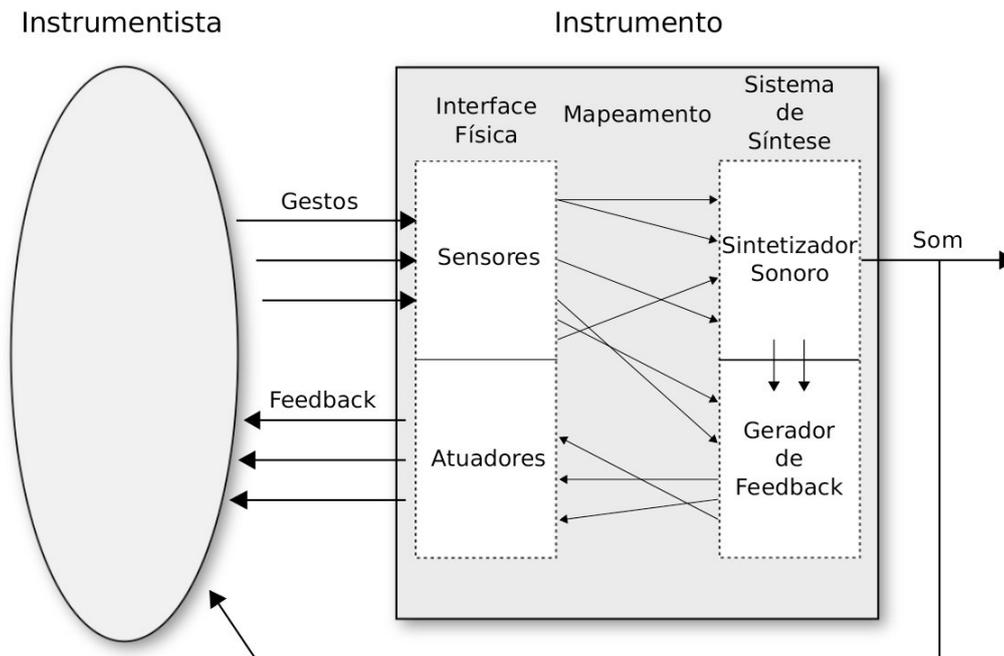


Figura 1: O modelo de DMI de Mark Marshall [Marshall, 2008, p. 26], onde pode-se visualizar a relação entre os diversos componentes. Um mesmo *hardware* pode originar diferentes DMIs (ou AMIs) alterando-se algoritmos de síntese e manipulação sonora executados pelo processador.

Este cenário proporcionou a construção de diversos instrumentos musicais digitais (DMIs, sigla de *Digital Musical Instruments*) e instrumentos musicais aumentados (AMIs, sigla de *Augmented Musical Instruments*), além de uma considerável quantidade de controladores digitais inspirados ou similares à instrumentos acústicos tradicionais, alguns deles descritos por Miranda e Wanderley em [Miranda and Wanderley, 2006, p. 20]. Utilizamos neste trabalho a sigla DMI para designar de forma genérica controladores gestuais utilizados no fazer musical, sejam estes constituídos apenas de componentes eletrônicos, possuírem partes eletrônicas e acústicas ou serem exclusivamente virtuais.

Os AMIs são DMIs específicos onde instrumentos musicais tradicionais foram aumentados com a inclusão de sensores e atuadores que permitem a este novo instrumento híbrido o controle de parâmetros sonoros ou mesmo a criação de novos sons [Miranda and Wanderley, 2006, p. 21-22].

2. GuitarAMI

Para a construção de um AMI deve-se considerar não somente a montagem de sensores e atuadores em um instrumento acústico, mas também o próprio instrumento acústico, o mapeamento entre os dados gestuais capturados pelo dispositivo e sua conversão em comandos para que a programação seja executada pelo computador responsável pelo processamento sonoro. Todos estes componentes são parte integrante de um AMI funcional.

No âmbito geral, os modelos de DMIs propostos por Bert Bongers, David Birnbaum, Marcelo Wanderley e Perry Cook foram analisados em [Marshall, 2008] e Mark Marshall criou um modelo de construção de DMIs que serviu de ponto de partida para a criação de um AMI utilizando o violão.

A construção e experimentação do primeiro protótipo de um AMI utilizando o violão foi construído durante o segundo semestre de 2014 [Meneses et al., 2015a]. Este dispositivo foi intitulado **GuitarAMI** e em sua criação foram utilizados *hardwares* e *soft-*



Figura 2: Performance do trio B.E.A.T.⁷ no evento *Palestras e Performances*, ocorrido no Auditório do Instituto de Artes da Unicamp em 24 de junho de 2015.

ware livres. A escolha dos componentes do AMI também considerou a facilidade de programação e construção, já que as plataformas Arduino² e Pure Data³ (PD), utilizadas no dispositivo, são amplamente documentadas e possuem como premissa a facilidade de uso.

O GuitarAMI foi utilizado tanto em performance quanto como ferramenta pedagógica onde, para cada uma das diferentes situações, foram utilizados *patches* (algoritmos desenvolvidos em PD) diferentes.

Para uma performance do trio B.E.A.T.⁴, mostrado na figura 2 e formado pelos músicos Alê Damasceno (bateria), Walmir Gil (trompete) e Edu Meneses (violão), o GuitarAMI foi utilizado executando um *patch* para enviar comandos no protocolo MIDI para o *software* Integra Live⁵ onde foram utilizados algoritmos de síntese sonora controlados dinamicamente pela posição do violão.

Durante um curso ministrado no Programa Guri⁶, pelo autor principal deste trabalho, os alunos puderam utilizar o GuitarAMI para disparar e controlar eventos sonoros pré-programados, utilizando o sensor ultrassônico disponível no GuitarAMI, que coleta em tempo real dados da distância entre o instrumento e o instrumentista.

O comportamento do GuitarAMI é portanto diretamente controlado pelos algorit-

²Arduino é uma plataforma de *hardware* livre para computação baseado em uma placa de microcontrolador simples, trabalhando em conjunto com um aplicativo utilizado como ambiente de desenvolvimento para escrever algoritmos e enviá-los ao *hardware* [Meneses et al., 2015b]. Maiores informações: <http://arduino.cc/en/Guide/Introduction>.

³Pure Data, ou PD, é uma linguagem visual de programação de código aberto desenvolvido originalmente no IRCAM por Miller Puckette e projetado para criar algoritmos (*patches*) de geração, controle e manipulação de dados de controle, áudio e vídeo, em tempo real [PD, 2015].

⁴Acrônimo de *Brazilian Electronic Aleatorium Trio*.

⁵O Integra Live é um programa de computador para processamento de áudio interativo originalmente financiado pela Creative Europe Desk UK entre 2007 e 2013. Atualmente o Integra Live é mantido pelo Integra Lab no Birmingham Conservatoire, no Reino Unido [IntegraLive, 2015].

⁶O Programa Guri é um programa de educação musical e inclusão social iniciado em 2008 através de uma parceria entre a Secretaria de Cultura do Estado de São Paulo e a Santa Marcelina - Organização Social de Cultura (<http://www.gurisantamarcelina.org.br/>) [Meneses et al., 2015a].

⁷Vide nota de rodapé 4.

mos contidos neste *patch* do PD. Pode-se também concluir que, de acordo com o modelo de DMI de Marshall, foram utilizados dispositivos diferentes em cada uma das atividades relatadas.

3. Expandindo as possibilidades violonísticas

Diferentemente de DMIs, que podem ser modificados não somente pela adição ou subtração de componentes físicos mas também pela alteração dos algoritmos responsáveis pela síntese e manipulação sonora, instrumentos musicais acústicos possuem características estruturais (físicas) correspondentes a características acústicas que são praticamente imutáveis, decorrentes da forma com que estes instrumentos foram construídos. Tais características, sejam elas consideradas restritivas ou enriquecedoras, podem ser exploradas em performances e composições musicais, ou mesmo evitadas ou minimizadas, de acordo com os critérios estéticos ou escolhas criativas do interprete ou compositor.

Como grande parte dos instrumentos de cordas, o violão possui uma grande variedade timbrística, porém outras características do instrumento, como envelope dinâmico de intensidade sonora com pouca sustentação ou impossibilidade de controle dinâmico após o ataque, são usualmente consideradas severas restrições impostas pela estrutura física do violão.

Uma das possibilidades de utilização do GuitarAMI é na execução de algoritmos programados na forma de *patches* em PD para reverter as restrições descritas anteriormente. Para esta tarefa foi criado um *patch* que utiliza o som emitido pelo violão e realiza manipulações sonoras com a finalidade de sustentar um elemento sonoro registrado pelo computador ou controlar o volume deste mesmo elemento.

Para que seja possível a sustentação da intensidade de um som emitido pelo violão por tempo indeterminado, a solução adotada foi a utilização da Transformada Rápida de Fourier, ou FFT⁸. Com a FFT é possível analisarmos um sinal de áudio em determinado momento no tempo para obtermos uma representação contendo a amplitude de todas as frequências do espectro deste mesmo áudio. O resultado desta análise é utilizado para gerar um som de duração indefinida que pode ser sintetizado com a utilização da operação inversa da FFT [Martin, 1986]. Este algoritmo é comumente denominado *Spectral Freeze*, ou simplesmente *Freeze* conforme mostra a figura 4.

Para a programação deste algoritmo utilizamos referencialmente o *patch freeze3.pd* disponibilizado por um membro ativo⁹ do fórum do Pure Data [PD-Forum, 2015]. A comunidade de desenvolvedores do PD é extremamente ativa e muitos dos algoritmos criados por usuários são disponibilizados, alterados e estudados pelos membros da comunidade. Para se adequar à proposta de utilização pelo GuitarAMI, a programação do algoritmo *freeze3.pd* foi modificada com a inclusão de opções de sensibilidade para a captura do áudio a ser sintetizado, parâmetros mapeados de forma direta com os sensores do AMI e um comando para liberação (*release*) do áudio sintetizado. Houve também acréscimo da programação referente ao controle de intensidade sonora do áudio sintetizado.

Os parâmetros disponíveis para controle desta manipulação sonora são o comando instantâneo de captura do áudio e início da sustentação, o comando instantâneo de liberação (*release*) deste mesmo áudio e o controle de volume do evento sonoro gerado pelo computador (o som sustentado indefinidamente). Para que o instrumentista controle estes parâmetros, foram utilizados dois interruptores momentâneos modelo SPST (botões ou *foot switches* de uma via de acionamento simples) disponíveis na base do GuitarAMI

⁸Sigla de *Fast Fourier Transform*.

⁹<http://forum.pdpatchrepo.info/user/ralf>.

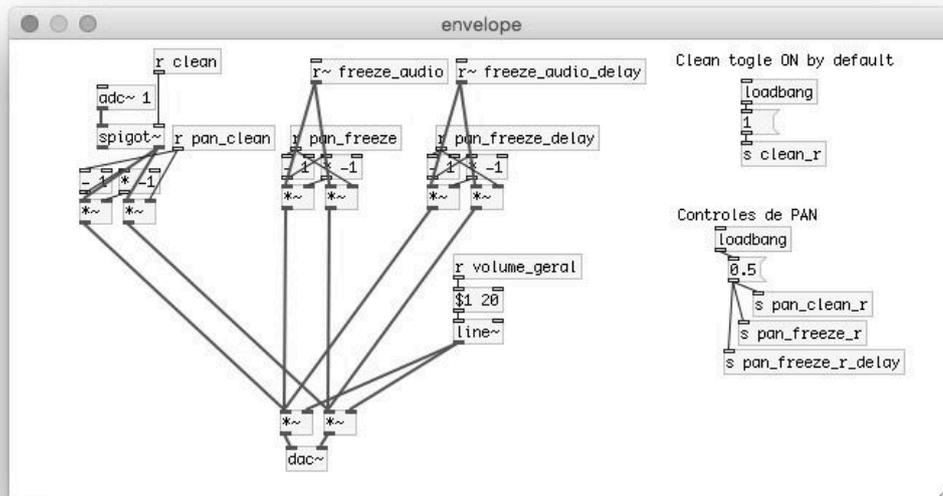


Figura 3: *Subpatch* contendo a programação do algoritmo *Env-shaper*.

(conectada ao computador) e que foram configurados para captura e liberação (*release*) do som do violão pelo algoritmo, além de um dos eixos do acelerômetro disponível no módulo do GuitarAMI (afixado na faixa do violão) para controle de volume geral.

Assim, o instrumentista pode capturar e liberar a síntese sonora ao acionar os interruptores com os pés, além de controlar a intensidade sonora da síntese, afastando ou aproximando o violão em relação ao seu corpo. Para esta movimentação do instrumento efetuamos o mapeamento gestual de forma que o parâmetro de controle de intensidade do *Freeze* (mostrado na figura 4) seja controlado simultaneamente pelos 3 eixos de sensibilidade do acelerômetro. Este tipo de mapeamento é chamado de *many-to-one* [Hunt et al., 2000] e, ao acrescentar maior complexidade no mapeamento, proporciona também uma experiência de controle mais próxima àquela oferecida pelos instrumentos acústicos tradicionais [Wanderley and Depalle, 2004].

Para que o instrumentista possa controlar a intensidade de um determinado som após sua emissão, de forma independente e sem a necessidade de utilizar o algoritmo *Freeze*, foi programado um *subpatch* dentro do *patch Freeze*. Este é responsável exclusivamente pela modelagem de envelope dinâmico em tempo real (*real-time envelope shaper*) da intensidade sonora; ou *Env-shaper*.

As informações geradas pela interação entre instrumentista e o sensor ultrassônico foram coletadas pelo módulo do GuitarAMI e convertidas em comandos que podem ser mapeados de forma direta (*one-to-one*) a um algoritmo de controle de intensidade sonora atuando diretamente sob o som captado do violão.

Este sensor ultrassônico é responsável pela aquisição de gestos instrumentais efetivos [Cadoz and Wanderley, 2000] realizados com uma das mãos do instrumentista que modula a intensidade sonora captada no violão. É, desta forma, possível ao instrumentista gerar desta forma envelopes dinâmicos antes impossíveis no violão tradicional.

4. Discussão e Conclusão

Este artigo apresentou o projeto e desenvolvimento de um violão acústico expandido, intitulado GuitarAMI. Observou-se que a utilização dos *patches Freeze* e *Env-shaper* de

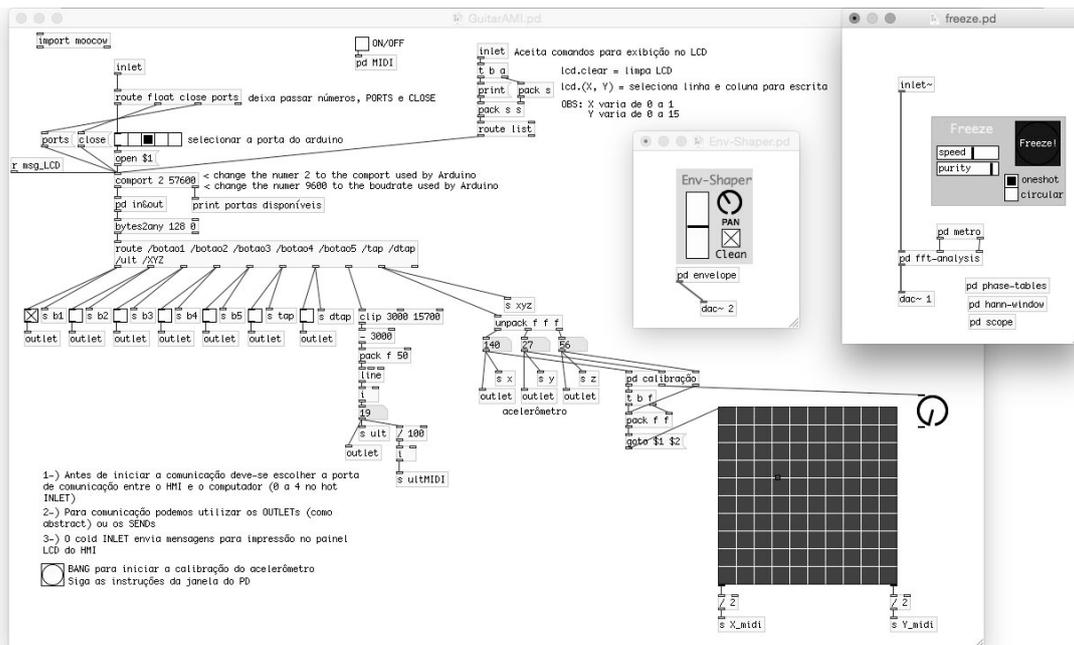


Figura 4: Patch para Pure Data contendo os algoritmos *Freeze* e *Env-shaper*, além da versão preliminar do algoritmo de controle e mapeamento gestual do GuitarAMI.

fato expandiram a sonoridade do violão, conforme apresentado na seção 3.

O GuitarAMI se comportou como esperado e com o patch apresentado na figura 4 foi possível alterar o envelope dinâmico ou sustentar indefinidamente sons emitidos pelo instrumento, desde que estes estejam dentro do espectro de sons que possam ser captados pelo equipamento utilizado, microfone ou captador específico do instrumento.

É interessante relatar algumas particularidades percebidas durante os testes de performance com o GuitarAMI. Como esperado, na utilização do algoritmo *Freeze* observou-se que o momento de acionamento do interruptor é crucial para definir a qualidade do som sustentado. Ao efetuar a captura durante o momento de ataque do envelope sonoro original do violão foi enviado ao algoritmo uma janela de captura contendo diversos ruídos transientes característicos desta fase, como o som das unhas do instrumentista friccionando as cordas. O momento ideal para registrar a janela de captura parece ser imediatamente após o ataque, quando grande parte dos ruídos transientes (estocásticos) já perderam energia e temos a presença mais definida da parte tonal (determinística) do sinal sonoro do violão [Serra, 1989].

A síntese sonora realizada pelo *Freeze* se torna mais distante do som original quanto maior for a sua complexidade. Acordes e sons muito ricos em harmônicos podem originar sons sintetizados com características sonoras bem diferentes das esperadas pelo instrumentista. Esta característica pode ser utilizada expressivamente durante uma performance, porém se distancia do objetivo do algoritmo, que foi programado para produzir um som que seja auditivamente equivalente ao som original, porém com duração indefinida.

O controle da intensidade sonora do som sintetizado no algoritmo *Freeze* requer posicionamento e movimentação precisas do violão. Parâmetros de sensibilidade da captura gestual realizada pelo acelerômetro devem ser calibrados de acordo com o instrumentista para fornecer uma resposta expressiva condizente com a intenção estética deste.

O algoritmo *Env-shaper*, apesar de sua simplicidade, requer alguma prática do

instrumentista para poder apresentar resultados satisfatórios durante a performance. É possível recriar em tempo real envelopes sonoros recorrentes na música acusmática e que antes eram realizados com a retrogradação do envelope sonoro, conforme exemplificados na figura 5.

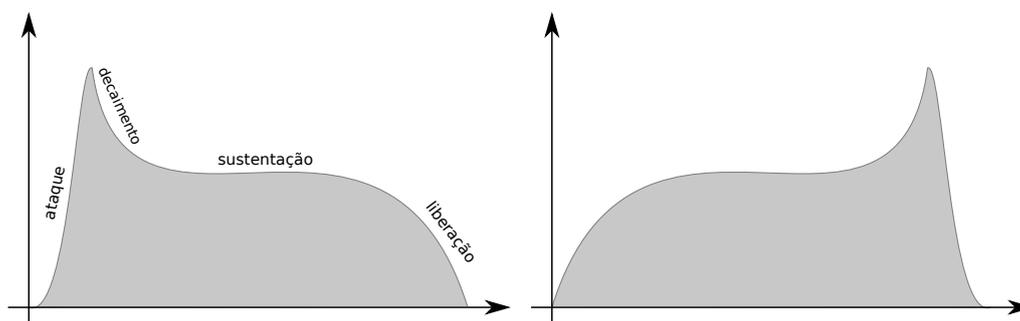


Figura 5: À esquerda exemplo de um envelope sonoro com seus componentes; À direita a retrogradação do mesmo envelope (uma das possibilidades de utilização do algoritmo *Env-shaper*). Utilizando o *Env-shaper* o instrumentista é capaz de criar diferentes formatos de envelope sonoro em tempo real.

5. Trabalhos Futuros

O GuitarAMI apresenta uma série de possibilidade de futuros desenvolvimentos e expansões. Entre outras, com relação à diferentes parametrizações dos algoritmos utilizados. O algoritmo *Freeze* possui diversos parâmetros de programação passíveis de alteração e que podem alterar o resultado sonoro apresentado. Ainda são necessários ajustes para obter uma síntese sonora psicoacusticamente mais próxima do som emitido pelo violão.

Pode também vir a ser desejável incorporar permanentemente estes algoritmos à programação do GuitarAMI e fornecer ferramentas para que o instrumentista possa mapear, de forma intuitiva, os movimentos recebidos e processados pelo dispositivo. Um *patch* especificamente programado para interconectar o GuitarAMI a quaisquer outros algoritmos desenvolvidos pelo instrumentista ou por um compositor poderá vir a contribuir no sentido a expandir as possibilidades de uso desta plataforma em futuros estudos e desenvolvimentos.

Referências

- Aliel, L. and Fornari, J. (2013). Creating an Ecologically Modeled Performance Through the Remote Manipulation of Multiple Soundscapes. In *Arquivos do XIV Simpósio Brasileiro de Computação Musical (SBCM)*, Brasília - DF - Brasil. Grupo de Computação Musical - IME/USP.
- Cadoz, C. and Wanderley, M. M. (2000). Gesture-music. In Wanderley, M. M. and Battier, M., editors, *Trends in Gestural Control of Music*, pages 71–94. Editions IRCAM – Centre Pompidou, Paris.
- Cook, P. (2001). Principles for designing computer music controllers. *Conference on New Interfaces for Musical Expression (NIME-01)*, pages 1–4.
- Fornari, J. (2012). Designing Bodiless Musical Instruments. In *Anais do 10^o Congresso de Engenharia de áudio*, number 10, pages 41–46, São Paulo. 16^a Convenção Nacional da AES Brasil.

- Hunt, A., Wanderley, M. M., Kirk, R., Pompidou, I. C., and France, P. (2000). Towards a Model for Instrumental Mapping in Expert Musical Interaction. Technical report.
- IntegraLive (2015). Integra live - web site. In <http://www.integralive.org/>. Último acesso em 13 de março de 2015.
- Marshall, M. T. (2008). *Physical Interface Design for Digital Musical Instruments*. PhD thesis, McGill University, Montreal.
- Martin, G. V. (1986). Final project report: Real-time Sound Freezing and Spectral Shifting. Technical report.
- Meneses, E. A. L., Fornari, J., and Wanderley, M. M. (2015a). Projeto e construção de uma interface gestual para improvisação livre. In *Anais do 13º Congresso de Engenharia de Áudio da AES Brasil*, pages 75–81. AES – Audio Engineering Society – Brazil Section.
- Meneses, E. A. L., Fornari, J., and Wanderley, M. M. (2015b). A study with hyperinstruments in free musical improvisation. In *Anais do XI Simpósio Internacional de Cognição e Artes Musicais*, Pirenópolis/GO. Associação Brasileira de Cognição e Artes Musicais (ABCM).
- Miranda, E. R. and Wanderley, M. M. (2006). *New Digital Musical Instruments: Control and Interaction Beyond the Keyboard*. Computer Music and Digital Audio Series. A-R Editions, Inc.
- PD (2015). Pure data - web site. In <http://puredata.info/>. Último acesso em 20 de abril de 2015.
- PD-Forum (2015). Pure data oficial forum. In <http://forum.pdpatchrepo.info/>. Último acesso em 14 de setembro de 2015.
- Serra, X. (1989). *A System for Sound Analysis/Transformation/Synthesis based on a Deterministic plus Stochastic Decomposition*. PhD thesis, Stanford University.
- Wanderley, M. M. and Depalle, P. (2004). Gestural control of sound synthesis. *Proceedings Of The Ieee*, 92(4):632–644.