



## Programação de um sensor inercial com 6 graus de liberdade e seu uso na exploração gestual da performance musical

MODALIDADE: COMUNICAÇÃO

SUBÁREA: Composição e Sonologia

*Augusto Armondes*

*Universidade Federal de Minas Gerais  
augusto\_armondes@hotmail.com*

*Caio Campos*

*Universidade Federal de Minas Gerais  
costacaiocampos@gmail.com*

*Eduardo Martins*

*Universidade Federal de Minas Gerais  
eduardo-spinardi@hotmail.com*

*Marcos Alves*

*Universidade Federal de Minas Gerais  
marcosac7@yahoo.com.br*

*Sérgio Freire*

*Universidade Federal de Minas Gerais  
sfreire@musica.ufmg.br*

**Resumo.** O artigo descreve a programação de um sensor inercial sem fio de 6 graus de liberdade em diferentes plataformas computacionais (MacOs e Linux) e discute as possibilidades de sua utilização na prática musical. A escolha deste sensor – MetamotionR – baseou-se em sua pequena dimensão e peso, transmissão sem fio com uma amostragem de 100 Hz, quantidade e qualidade dos dados gerados. Um algoritmo de fusão de dados já embarcado no sensor permite que os dados de aceleração linear sejam independentes da gravidade e que sua atitude espacial seja estimada. O texto também aborda diferentes maneiras de se visualizar e processar os dados de movimento, e traz exemplos de seu potencial como ferramenta para análise de performances e como interface humano-máquina em projetos interativos de criação musical.

**Palavras-chave.** IMUs, Streaming de dados, Multimodalidade, Performance musical

**Title.** Programming an Inertial Sensor with 6 Degrees of Freedom and its Use in Gestural Exploration of Musical Performances.

**Abstract.** The article describes the programming of a 6-degrees of freedom wireless inertial sensor in different computing platforms (macOS and Linux) and discusses the possibilities of its use in musical practice. The choice of this sensor – MetamotionR – was based on its small size and weight, wireless transmission with a sampling rate of 100 Hz, and quantity and quality of the generated data. A fusion algorithm already built into the sensor allows linear acceleration data to be gravity-independent and its spatial attitude to be estimated. The text also addresses different ways of visualizing and processing motion



data and provides examples of its potential as a tool for performance analysis and as a human-machine interface in interactive music creation projects.

**Keywords.** IMUs, Data Streaming, Multimodality, Musical Performance

## Introdução

O recente e contínuo avanço tecnológico na área de sensores tem alavancado as pesquisas que abordam a dualidade gesto/som, tanto no desenvolvimento de instrumentos aumentados (TURCHET, 2018) quanto na análise de performances musicais (CAMURRI;VOLPE, 2011). Com equipamentos de captura de movimento cada vez mais compactos e de menor custo, a abordagem da multimodalidade na prática musical - e em todo o processo de comunicação e expressão humana - ganha novas possibilidades de exploração.

Com uma grande variedade de sensores – os chamados IMUs (Inertial Measurement Units) – no mercado, é necessário um estudo aprofundado sobre a programação e funcionamento desses equipamentos, a fim de entender e interpretar corretamente questões de latência, tipos de medida, formatação e transmissão dos dados, fatores essenciais em análises que exigem sincronia de todas as fontes de informações.

Este trabalho descreve o processo de programação do IMU MetaMotionR, da empresa MBientLab, em dois sistemas operacionais - MacOS e Linux - e o modo de visualização e processamento dos dados, além do potencial de sua aplicação em análises de performance e como interface digital em sistemas interativos.

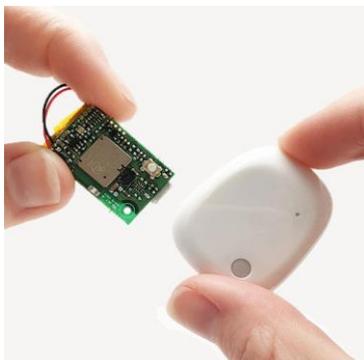
Inicialmente apresentamos o sensor escolhido, seus protocolos de conexão e algoritmos. Em seguida descrevemos a programação do IMU nos dois sistemas operacionais disponíveis e o modo de transmissão das informações por streaming. Na sequência, abordamos a visualização e processamento dos dados de aceleração linear, velocidade angular e coordenadas polares juntamente com descritores de áudio. Discutimos então um pequeno exemplo de análise gestual, seguido de um experimento curto associando os dados do sensor com dados advindos da análise de áudio. Concluimos com as considerações relativas ao presente trabalho e os próximos passos da pesquisa.

## Características do sensor escolhido

O MetaMotionR (Figura 1), desenvolvido pela MBientLab, é um sensor portátil em formato de relógio que transmite, via protocolo *Bluetooth Low Energy* (BLE), informações de aceleração linear, velocidade angular e magnetômetro em três eixos, além de temperatura, luminosidade ambiente e pressão barométrica. O protocolo BLE permite que os dispositivos

em uso entrem em um modo de baixa energia ou inatividade quando a transmissão/recepção de um evento é concluída, operando com mais eficiência em largura de banda limitada.

**Figura 1 – MetaMotionR, pequeno sensor de movimento em formato de relógio desenvolvido pela empresa MBientLab**



Fonte: Site de Abdullah Albaitam – <http://fabacademy.org/2019/labs/bahrain/students/abdullah-albaitam/assignments/week09/>

O sensor permite diversas configurações de uso. Para registro em memória interna, acessada através do aplicativo *Metabase App* (Windows, iOS e Android), ele opera com amostragem de até 800 Hz. No modo *wireless streaming* a taxa é de 100 Hz para os acelerômetros e giroscópio e 25 Hz para o magnetômetro. Possui um algoritmo de fusão, desenvolvido pela empresa Bosch, permitindo estimativa de aceleração linear sem a influência da gravidade e da atitude espacial.

A fusão possui quatro modos de operação<sup>1</sup> que combinam os sensores de formas diferentes: (a) NDoF (nove graus de liberdade), que estima a orientação absoluta a partir de dados dos três sensores; (b) IMUPlus, que estima a orientação relativa no espaço usando dados do acelerômetro e giroscópio; (c) Compass, combinação de acelerômetro e magnetômetro; (d) M4G, que usa o magnetômetro. Utilizamos, tanto na programação em macOS quanto Linux, o modo IMUPlus, que possui maior sample rate e pelo fato de a orientação absoluta normalmente não oferecer problemas em performances musicais ao vivo.

Com o objetivo de identificar e compreender as limitações do IMU, realizamos um estudo comparativo com o sistema óptico de captura de movimento Qualisys (FREIRE *et al.*, 2020), que nos mostrou uma grande viabilidade do uso deste sensor fora de ambientes controlados, com uma precisão compatível com as aplicações musicais pretendidas.

---

<sup>1</sup> <https://mbientlab.com/tutorials/SensorFusion.html>

Descreveremos a seguir o processo de programação do sensor nos dois sistemas operacionais utilizados.

## **Programação em diferentes linguagens**

A empresa MBientLab não oferece, até o momento, software para o uso deste sensor no modo streaming em sistemas operacionais MacOS e Linux. Como alternativa, são disponibilizados códigos e tutoriais para a compilação do programa pelo próprio usuário.<sup>2</sup> Encontramos códigos em Swift para MacOS e em Python para Linux, os quais foram adaptados no desenvolvimento dos drivers. Apresentaremos inicialmente a programação para o sistema MacOS.

Utilizamos a versão MacOS 10.14 (Mojave)<sup>3</sup> em um computador iMac com processador Intel. Em um código em Swift adaptado, programamos o sensor para o envio dos dados de aceleração linear e velocidade angular (ambos nos eixos x, y e z), e de atitude em ângulos de Euler, a uma taxa de 100 Hz. Com o algoritmo de fusão IMUPlus, os dados de aceleração linear não sofrem influência da força de gravidade. Diversos fatores podem afetar o desempenho nesse tipo de conexão, como hardwares, versão de software, distância entre os equipamentos, interferências e nível de bateria.

Os dados recebidos via BLE são distribuídos em rede local por OSC (Open Sound Control), incorporando um framework desenvolvido por Devin Roth (2016). Cada mensagem OSC é composta por 10 dados de 16 *bits*: três para aceleração linear (*g*), três para velocidade angular (*graus/s*) e quatro para os ângulos de Euler (*graus*); como um destes quatro valores é redundante, ele é excluído dos cálculos posteriores. Fizemos a última atualização da programação para MacOS em 2020, quando realizamos também um teste de latência, comparando o intervalo temporal entre o impacto sonoro da queda do IMU e a variação brusca de seus dados de aceleração, utilizando um MacBook Pro e uma placa de áudio Presonus Studio8. O valor estimado foi de 24,4 ms.

A visualização dos dados é feita em Max/Msp, uma linguagem de programação visual voltada para música e multimídia com processamento em tempo real, largamente utilizada em sistemas interativos. Criamos um patch com um monitor para acompanhar a taxa de amostragem real da entrada de dados, e o número de mensagens recebidas por segundo é

---

<sup>2</sup> <https://mbientlab.com/tutorials/MetaWearAPI.html>

<sup>3</sup> A programação realizada funciona em sistemas posteriores ao 10.14, com processadores Intel. O uso do emulador Rosetta com os novos processadores M1 não deu resultados satisfatórios até o momento, principalmente com mais de um sensor conectado. E a MBientLab ainda não sinalizou suporte para esses novos processadores da Apple.

contado e apresentado como uma curva, que varia em torno de 100 Hz. Os dados de cada mensagem são gravados em um buffer, com resolução de 1 ms, e marcados com o seu tempo de chegada. Observando o intervalo de tempo entre duas mensagens subsequentes em um trecho, notamos a maior presença dos valores 0, 3, 12 e 15 ms (com média geral de 9,88 ms). Estas irregularidades são intrínsecas ao protocolo BLE, que utiliza intervalos temporais discretos (múltiplos de 1,25 ms) para transmissão dos dados. Salvamos um registro de erro contendo os instantes em que os intervalos de tempos são longos ( $> 30$  ms).

A partir dessa observação percebemos a necessidade de reconstruir o sample rate de 100 Hz utilizando um timer e um filtro de média móvel. A cada 10 ms, um novo valor para cada uma das três grandezas é gerado na saída do filtro por um clock interno. Como a média geral do intervalo de tempo entre as mensagens está abaixo de 10 ms, e os valores ocorrendo em torno a 30 ms são raros, escolhemos utilizar um filtro com três pontos, alimentado com a entrada de um novo dado quando este ocorre antes do próximo tick, ou com a repetição do dado anterior quando não chegam novas mensagens entre dois ticks. Esta ação impede a repetição de dois valores idênticos na saída do filtro.

Já para o sistema Linux, utilizamos a versão Ubuntu 22.04 LTS, instalado por meio de um dual boot com o sistema MacOS. O código em Python foi desenvolvido a partir de adaptações dos exemplos fornecidos no site de tutoriais da MetaWear, durante o primeiro semestre de 2022. Desenvolvemos uma programação semelhante à feita previamente em linguagem Swift, na qual foi utilizado o algoritmo de fusão IMUPlus para captar os dados de aceleração linear, velocidade angular e ângulos de Euler nos três eixos.<sup>4</sup> Após alguns testes de streaming, verificamos que a implementação no sistema Ubuntu apresentou uma taxa de amostragem mais estável em relação à implementação desenvolvida previamente para MacOS.

Para garantir uma uniformidade de 100 Hz na captação de dados, desenvolvemos um patch no software de programação musical PureData - semelhante ao desenvolvido para Max/MSP - capaz de receber as mensagens OSC advindas da transmissão via BLE e aplicar um filtro de média móvel com ticks a cada 10 ms.<sup>5</sup>

Em busca de uma análise multimodal, fizemos a captação de áudio simultaneamente aos dados de movimento durante a performance. Utilizamos uma interface de áudio Focusrite Saffire Pro 40, que foi viabilizada no sistema Ubuntu pelo driver de áudio Jack. A gravação dos

---

<sup>4</sup> Ver tutorial em <https://drive.google.com/file/d/1IVKN3yfwLdUjI5mCt7HjzfrReG0AGbf/view>. Os demais arquivos estarão disponíveis no site de nosso laboratório: <https://musica.ufmg.br/lapis/>

<sup>5</sup> Esta frequência de amostragem pode ser facilmente alterada, tanto para uso em condições de muita interferência, quanto para ajuste com outros fluxos de informação.

dados do sensor e do áudio são feitas em um patch desenvolvido em PureData, com sample rate de 48 KHz, block size 64 bits, delay de 5 ms e 24 bits. Após a captação, fazemos uma correção de latência de 44 ms, medida por um teste empírico de latência de toda a cadeia de programas utilizados, e transferimos os arquivos para o sistema MacOs a fim de visualizar graficamente os dados coletados.

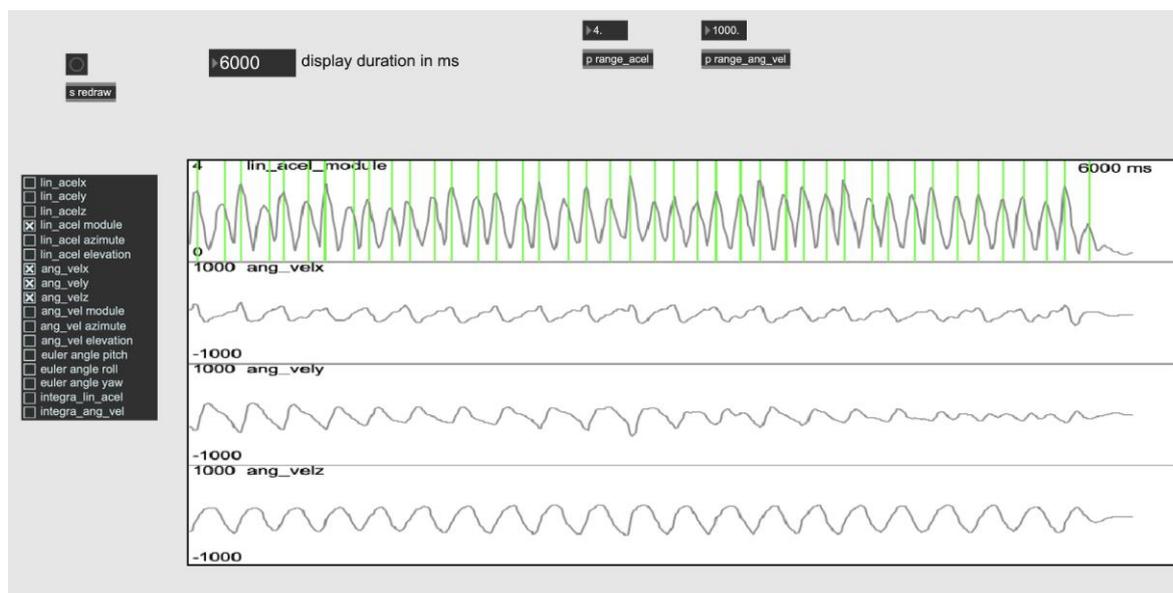
## **Visualização e processamento dos dados**

Até o presente momento, as rotinas desenvolvidas para visualização em tempo real rodam apenas na linguagem Max. Para isto, utilizamos o objeto [jit.lcd], que oferece bastante flexibilidade em seu uso. Este objeto também permite associar visualmente os dados de movimento com diferentes descritores de áudio.

Depois de capturados, os dados de aceleração linear e de velocidade angular são também expressos em coordenadas polares, contando com um valor do módulo do vetor e dois ângulos de orientação: azimute (entre -180 e 180 graus) e elevação (entre -90 e 90 graus). Os dados de aceleração linear podem ser rotacionados segundo os ângulos de Euler calculados pelo próprio IMU. Desse modo, as direções de translação (os eixos de aceleração linear) não são alteradas quando o sensor é rotacionado. Todos os fluxos de dados podem ser filtrados segundo necessidades específicas.

A Figura 2 mostra a interface de escolha e exibição de diferentes curvas dos dados gerados pelo IMU, produzidos por um movimento de vai-e-vem no ar simulando um trêmolo em um reco-reco. Os dados a serem exibidos são escolhidos em um menu. Pode-se também ajustar a escala horizontal (temporal) e vertical dos gráficos, além da possibilidade de redimensionar todo o display.

**Figura 2 – Interface de exibição dos dados gerados pelo IMU. É possível criar diversos arranjos de curvas selecionadas em um menu. Os marcadores em verde indicam os picos estimados da curva. Este gesto advém de uma simulação no ar de um trêmolo mensurado em um reco-reco.**



Fonte: Autoria própria

Desenvolvemos uma ferramenta de segmentação dos dados de aceleração linear e velocidade angular, utilizando um Schmitt trigger, que requer dois limiares para funcionar: um superior, que deve ser ultrapassado para indicação de atividade ou início do gesto; um inferior, que deve ser ultrapassado no sentido contrário para indicar repouso ou final do gesto. Uma vez iniciado o gesto, é possível determinar os pontos máximos das curvas geradas para as diferentes grandezas. Estes momentos podem também ser plotados nos gráficos acima descritos. Para gestos curtos simples ou repetidos, consideramos que o interesse está em sua amplitude máxima e em sua orientação espacial neste momento. Para gestos irregulares de duração mais longa, usamos também um procedimento de integração dos valores que vai sendo ponderado por sua duração, obtendo assim curvas mais suaves relativas aos movimentos de translação e rotação. Para gestos muito lentos, é melhor observar diretamente os ângulos de Euler, já que os valores de aceleração linear e velocidade angular ficam muito baixos, podendo se confundir com ruído de fundo.

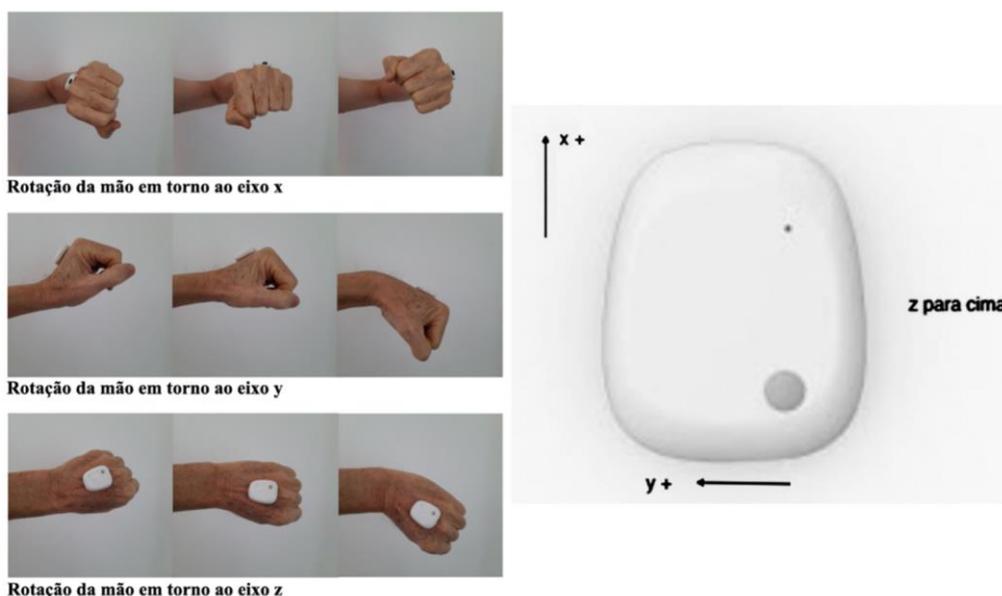
Os descritores de áudio que utilizamos são expressos, em sua maioria, por dados que se atualizam a cada 10,67 ms (janelas de análise de 2048 pontos, com *hopsize* de 512 amostras, com uma frequência de amostragem de 48 kHz). Nesses casos, modificamos levemente a amostragem dos dados do sensor para sincronização, além de aplicar um atraso de 24,4 ms (no

caso de um sistema com um MacBook Pro) e de 44 ms (no sistema Linux com um iMac) aos descritores de áudio.

## Pequeno exemplo de análise gestual

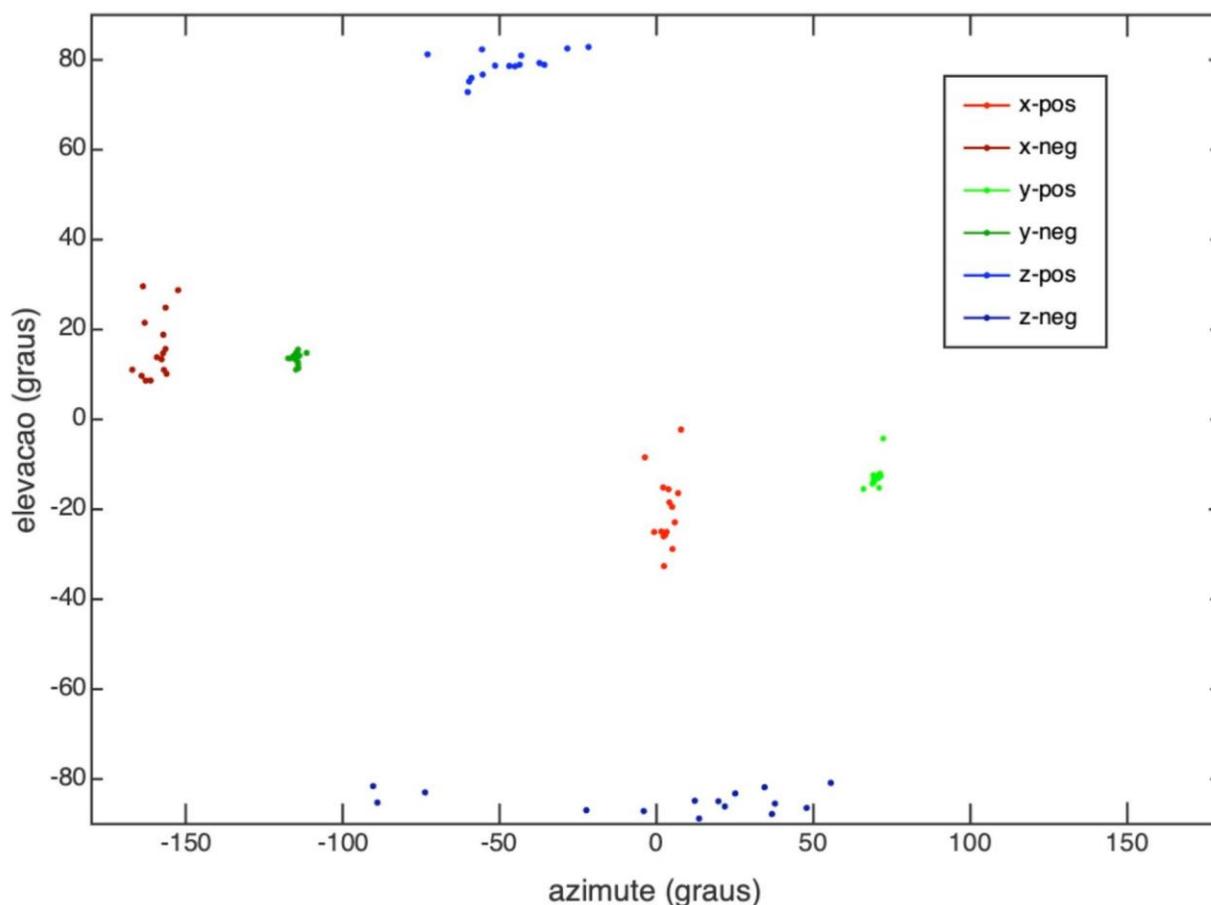
Realizamos um pequeno experimento com gestos simples de rotação nos três eixos principais. Foram executadas 15 rotações para cada eixo, nos sentidos positivo e negativo (ver Figura 3). No momento de valor máximo do módulo da velocidade angular, extraímos os valores de azimute e elevação. Os resultados obtidos podem ser vistos na Figura 4. A combinação desses dois valores permite uma clara separação entre cada uma das seis rotações simples. O eixo principal de cada rotação pode ser identificado pela concentração dos valores em uma única região: azimute em torno  $3^\circ$  para rotações positivas e em torno a  $-159^\circ$  para rotações negativas no eixo x; azimute em torno  $70^\circ$  para rotações positivas e em torno a  $-115^\circ$  para rotações negativas no eixo y; elevação em torno a  $79^\circ$  para rotações positivas e em torno a  $-85^\circ$  para rotações negativas no eixo z. Há um certo espalhamento de valores, principalmente nas rotações em torno aos eixos x e z; isto se deve à maior dificuldade de se realizar esses gestos com a mão sem envolver os outros eixos. A distribuição dessas variáveis em um plano nos permite vislumbrar que gestos curtos mais complexos, utilizados em uma performance musical ao vivo, também possam ser identificados e classificados segundo os diferentes eixos de rotação colocados em jogo.

**Figura 3 – Orientação no sentido crescente dos eixos no sensor e a rotação em cada eixo.**



Fonte: Autoria própria

**Figura 4 – Valores de azimute e elevação para gestos simples de rotação, efetuados em torno de cada eixo nos dois sentidos. Foram realizadas 15 amostras de cada um deles. Pode-se notar que é possível classificá-los sem dificuldade.**



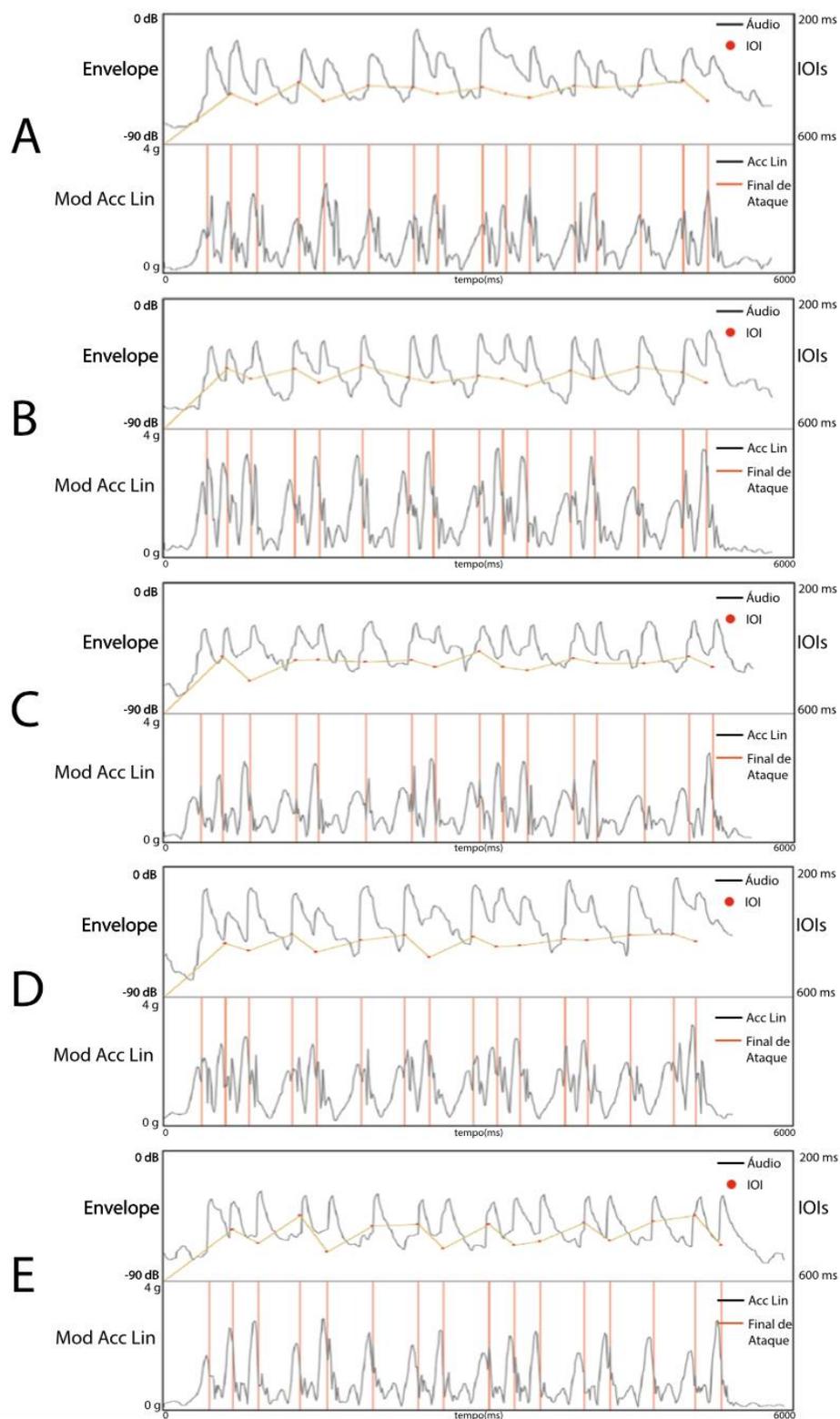
Fonte: Autoria própria

## Exemplo de associação com descritores de áudio

Realizamos também um outro experimento curto, desta vez associando dados gerados pelo IMU com dados advindos da análise de áudio (ARMONDES *et al.*, 2020). Escolhemos o motivo inicial da *Clapping Music* (Figura 5), de Steve Reich (1980), para servir de comparação entre diferentes performances realizadas por integrantes do grupo de pesquisa. Um sensor foi posicionado no dorso da mão dominante de cada pessoa. Foram feitas duas repetições do motivo, com os dados extraídos visualizados em tempo real na tela do Max, realizadas nas mesmas condições de captação, sem utilização de metrônomo. O início e o fim da visualização foram determinados com um pedal. Utilizamos a detecção do final de ataque, o envelope dinâmico (estimado em janelas de 2048 pontos, com quatro superposições parciais), os IOIs (inter-onset intervals) e o módulo das acelerações lineares. A detecção de onsets e de finais de ataque é feita a partir de um envelope dinâmico com uma resolução oito vezes maior do que a



**Figura 6 – Curvas de envelope dinâmico, IOIs normalizados, módulo de aceleração linear e pontos de final de ataque de cinco performances da Clapping Music.**



Fonte: Autoria própria

Analisaremos agora as curvas do módulo da aceleração linear. A, C, D e E possuem amplitude da aceleração linear próxima, enquanto B apresenta um valor maior que os demais. Esse fato pode estar relacionado à técnica utilizada: o músico B movimentou as duas mãos, uma de encontro à outra, enquanto os outros mantiveram a mão sem o sensor parada e a com sensor em movimento. Importante ressaltar que maior módulo de aceleração linear não se traduz em maior amplitude de envelope dinâmico, uma vez que B não possui o maior pico de dinâmica, e também não indica alguma facilidade ou dificuldade na reprodução do ritmo proposto, pois observamos semelhanças nos IOIs entre esse músico e os outros mais regulares. Na maioria das vezes, há um aumento do módulo da aceleração na sucessão de duas ou três colcheias.

Considerando os momentos de ataque nas performances, podemos perceber que esses pontos ocorrem após o pico de aceleração, ou seja, o maior valor de aceleração ocorre antes do encontro entre as duas mãos, o que contradiz a intuição inicial compartilhada por nós, de que a mão acelera ao máximo até o contato. Ainda não está claro se este pico é resultado da preparação do ataque, ou está ligado a algum mecanismo de absorção de impacto.

Outro ponto de destaque ocorre no momento imediatamente após o final do ataque, que chamamos de rebote: C e E apresentam módulos menores de aceleração linear no rebote, enquanto A, B e D possuem valores maiores nesses momentos. Podemos interpretar esse fato com a técnica utilizada, onde uma mão mais solta causa mais rebote, enquanto uma mão mais firme se movimenta menos após o impacto e, portanto, menos rebote. Apesar de em E ocorrer menor regularidade nos IOIs, não podemos afirmar que um modo específico colabore para a precisão dos ataques do trecho, já que C, utilizando a mesma técnica, possui grande regularidade, assim como A ou B, que utilizam outra técnica. A performance C apresenta um comportamento distinto nas pausas mais longas, com um pequeno pico de aceleração bem anterior ao próximo ataque, fato que não ocorre nas outras performances.

Com essa primeira abordagem sobre gesto e som, percebemos que cada performer realiza de modo próprio os gestos envolvidos na realização do trecho e seria possível a identificação de cada um a partir da análise desses dados.

## **Considerações finais**

Este trabalho abordou as etapas necessárias para se colocar em funcionamento um sensor portátil, sem fio, de baixo custo e com grande potencial para aplicações musicais, em dois sistemas operacionais distintos. A implementação em Linux mostrou-se bastante promissora, e acreditamos que seja facilmente adaptada para micro-computadores como os Raspberry Pi. Consideramos que a estabilidade na recepção de dados via BLE foi maior com o

Linux do que com o MacOS rodando no mesmo computador. No entanto, a latência de resposta do IMU foi acima da expectativa. Como não temos muita experiência com este sistema operacional, pode ser que ainda seja possível otimizar esse atraso, criando algoritmos para identificação de possíveis fatores: seria o código Python utilizado? Sua biblioteca de OSC? A transmissão de mensagem OSC do ambiente Python para o PureData?

Mostramos também, ainda que de maneira incipiente, as possibilidades de se visualizar diferentes aspectos gestuais de modo interativo, que podem ter grande valor pedagógico na prática de instrumentos que utilizam intensamente movimentos das mãos. Além disso, a grande capacidade de se diferenciar aspectos espaciais de gestos curtos pode ter uma aplicação efetiva em situações baseadas na prática de *air drum*.

Por fim, a associação de dados de áudio com dados de movimento nos abre novas possibilidades de análise de performances, como tentamos demonstrar com um exemplo bem simples. Além disso, nos coloca uma série de questões que necessitam aprofundamento: como a aceleração (uma grandeza física de interpretação muitas vezes difícil) das mãos se relaciona com a produção sonora? Como se dá esta relação em diferentes instrumentos de percussão, ou nas diversas arcadas das cordas? Como os gestos de rotação e translação se combinam em diferentes situações? A utilização de gravação em vídeo com uma taxa alta de captura de quadros pode auxiliar a entender algumas dessas questões. E, certamente, questões similares podem ser exploradas por meio de situações criativas nas quais os gestos comandam processos de transformação e síntese sonora. Em resumo, acreditamos ter apresentado aqui algumas possibilidades abertas pela incorporação de um simples sensor para a prática e a análise de performances e para a criação musical interativa.

## Referências

ARMONDES, Augusto; FREIRE, Sérgio; CAMPOS, Caio Costa; SILVA, Rubens. *Uma primeira aproximação multimodal da técnica de strumming*. In: XXX Congresso da Anppom. 2020.

CAMURRI, Antonio; VOLPE, Gualtiero. Multimodal Analysis of Expressive Gesture in Music Performance. In: SOLIS, Jorge; NG, Kia (Orgs.), *Musical Robots and Interactive Multimodal Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, p. 47–66.

FREIRE, Sérgio; SANTOS, Geise; ARMONDES, Augusto; MENESES, Eduardo A. L.; WANDERLEY, Marcelo M. *Evaluation of Inertial Sensor Data by a Comparison with Optical Motion Capture Data of Guitar Strumming Gestures*. *Sensors*, v. 20, n. 19, 2020.

REICH, Steve. *Writings on music, 1965-2000*. Oxford: Oxford University Press, 2002.



ROTH, D. SwiftOSC. Framework OSC para linguagem Swift, 2006.

TURCHET, Luca. *Some Reflections on the Relation between Augmented and Smart Musical Instruments*. In: PROCEEDINGS OF THE AUDIO MOSTLY 2018 ON SOUND IN IMMERSION AND EMOTION, 2018, New York, NY, USA: Association for Computing Machinery, 2018.

