

**UNIVERSIDADE DE CAXIAS DO SUL
ÁREA DO CONHECIMENTO DE CIÊNCIAS EXATAS E
ENGENHARIAS**

RAFAEL BASSO SBERSE

**AEROCHORD: UM HIPERINSTRUMENTO MUSICAL
CONTROLADO POR GESTOS**

CAXIAS DO SUL

2026

RAFAEL BASSO SBERSE

**AEROCHORD: UM HIPERINSTRUMENTO MUSICAL
CONTROLADO POR GESTOS**

Trabalho de Conclusão de Curso
apresentado como requisito parcial
à obtenção do título de Bacharel em
Ciência da Computação na Área do
Conhecimento de Ciências Exatas e
Engenharias da Universidade de Caxias
do Sul.

Orientador: Profa. Dra. Elisa Boff

CAXIAS DO SUL

2026

RAFAEL BASSO SBERSE

**AEROCHORD: UM HIPERINSTRUMENTO MUSICAL
CONTROLADO POR GESTOS**

Trabalho de Conclusão de Curso
apresentado como requisito parcial
à obtenção do título de Bacharel em
Ciência da Computação na Área do
Conhecimento de Ciências Exatas e
Engenharias da Universidade de Caxias
do Sul.

BANCA EXAMINADORA

Profa. Dra. Elisa Boff
Universidade de Caxias do Sul - UCS

Prof. Dr. Marcelo Luís Fardo
Universidade de Caxias do Sul - UCS

Prof. Dr. Marcos Eduardo Casa
Universidade de Caxias do Sul - UCS

RESUMO

Este trabalho apresenta o desenvolvimento e a avaliação do Aerochord, um instrumento musical controlado por gestos das mãos, que utiliza uma câmera comum para transformar movimentos em sons digitais. O objetivo é tornar a expressão musical mais acessível, tanto para pessoas sem experiência com instrumentos quanto para pessoas com mobilidade reduzida. Inicialmente, foi realizada uma revisão da literatura sobre interação entre pessoas e computadores e tecnologias aplicadas à música, seguida de uma análise de soluções já existentes. Essa investigação mostrou que ainda há dificuldades em reunir acessibilidade, rapidez de resposta e qualidade de execução musical em um mesmo sistema. Com base nesses resultados, foram definidos os requisitos do Aerochord, incluindo reconhecimento dos movimentos das mãos em tempo real, associação entre gestos e sons e comunicação com softwares musicais por meio do padrão MIDI 2.0, mantendo compatibilidade com versões anteriores quando necessário. O sistema foi desenvolvido em C++ utilizando o *framework* JUCE, com uma estrutura projetada para reduzir atrasos e garantir que os sons acompanhem os movimentos do usuário de forma natural. Os testes realizados demonstraram tempo de resposta inferior ao limite perceptível pela maioria das pessoas, baixo consumo de recursos computacionais e facilidade de uso, mesmo para participantes sem formação musical. Os resultados confirmam a viabilidade técnica e prática da proposta e apontam possibilidades de evolução, como suporte a execuções mais complexas, adaptação automática aos gestos dos usuários e integração com ambientes imersivos. Espera-se que o Aerochord contribua para as áreas de interação humano-computador e tecnologia musical ao oferecer um instrumento acessível, responsivo e expressivo.

Palavras-chave: interfaces gestuais, computação musical, MIDI 2.0, democratização musical, visão computacional, hiperinstrumentos, acessibilidade, baixa latência.

ABSTRACT

This work presents the development and evaluation of Aerochord, a hand-gesture-controlled musical instrument that uses a standard camera to transform movements into digital sounds. The goal is to make musical expression more accessible, both for people without experience with instruments and for people with reduced mobility. Initially, a literature review on human-computer interaction and technologies applied to music was conducted, followed by an analysis of existing solutions. This investigation showed that there are still difficulties in combining accessibility, response speed, and musical performance quality in a single system. Based on these results, the requirements for Aerochord were defined, including real-time hand movement recognition, gesture-to-sound mapping, and communication with musical software using the MIDI 2.0 standard, maintaining compatibility with legacy versions when necessary. The system was developed in C++ using the JUCE framework, with a structure designed to reduce delays and ensure that sounds naturally follow the user's movements. The tests performed demonstrated a response time below the perceptible limit for most people, low consumption of computational resources, and ease of use, even for participants without musical training. The results confirm the technical and practical feasibility of the proposal and point out possibilities for evolution, such as support for more complex performances, automatic adaptation to user gestures, and integration with immersive environments. It is expected that Aerochord will contribute to the fields of human-computer interaction and music technology by offering an accessible, responsive, and expressive instrument.

Keywords: gesture interfaces, musical computing, MIDI 2.0, musical democratization, computer vision, hyperinstruments, accessibility, low latency.

LISTA DE ABREVIATURAS E SIGLAS

IHC	Interação Humano-Computador
IMD	Instrumento Musical Digital
MIDI	<i>Musical Instrument Digital Interface</i>
UMP	<i>Universal MIDI Packet</i>
MPE	<i>MIDI Polyphonic Expression</i>

SUMÁRIO

1	INTRODUÇÃO	8
2	INTERAÇÃO GESTUAL E HIPERINSTRUMENTOS	11
2.1	Interação Multimodal e Gestual	11
2.1.1	O Paradigma da Interação Multimodal	11
2.1.2	A Expressividade da Interação Gestual na Música	12
2.2	Dispositivos e Ferramentas para Detecção de Gestos	13
2.3	Hiperinstrumentos Musicais	15
2.3.1	Definição e Características	15
2.3.2	Evolução e Exemplos Notáveis	16
2.3.3	Hiperinstrumentos e Interação Gestual	19
2.4	Considerações sobre o Capítulo	19
3	COMPUTAÇÃO E MÚSICA	21
3.1	Fundamentos da Computação Musical	21
3.2	O Protocolo MIDI (Musical Instrument Digital Interface)	22
3.2.1	Fundamentos e Limitações do MIDI 1.0	23
3.2.2	Arquitetura Técnica do MIDI 2.0	23
3.2.3	Novas Capacidades Expressivas do MIDI 2.0	25
3.2.4	Persistência de Performances e Arquivos	25
3.2.5	Justificativa Final para a Arquitetura do Aerochord	26
3.3	Teoria Musical Aplicada	26
3.3.1	Escala Cromática e Oitavas	26
3.3.2	Articulações	27
3.3.3	Efeitos de expressividade	27
3.4	Considerações sobre o Capítulo	28
4	ESTADO DA ARTE EM INTERFACES MUSICAIS GESTUAIS	29
4.1	Paradigmas de Interação em Interfaces Musicais Gestuais	29
4.1.1	Interação de Alta Fidelidade com Contato Físico	30
4.1.2	Gestos Livres em Volume 3D	30
4.1.3	Detecção de Pose com Visão Monocular	31
4.2	Mapeamento Gesto-Som	32
4.2.1	Estratégias de Mapeamento	32
4.2.2	Filosofias de Design de Mapeamento: <i>Movement-First</i> vs. <i>Sound-First</i>	33
4.3	Avaliação, Usuários e Contextos de Uso	33

4.3.1	Métricas e Metodologias de Avaliação	34
4.3.2	Perfis de Usuário e Cenários de Aplicação	34
4.4	Síntese Crítica e Posicionamento do Aerochord	34
4.4.1	Mapeamento Comparativo das Abordagens	35
4.4.2	A Lacuna Identificada e a Proposta de Valor do Aerochord	35
4.5	Tendências e Perspectivas Futuras	36
5	DESENVOLVIMENTO DA PROPOSTA DE SOLUÇÃO	37
5.1	Requisitos do Sistema	37
5.2	Modelagem de Casos de Uso	38
5.3	Arquitetura do Sistema	39
5.4	Captura e Detecção de Pose	39
5.5	Máquina de Estados e Análise Gestual	40
5.6	Estratégias de Mapeamento Gesto-Som	41
5.7	Geração de Protocolo e Suporte Nativo a MIDI 2.0	43
6	TESTES E ANÁLISE DE RESULTADOS	45
6.1	Avaliação de Desempenho Técnico e Latência	45
6.1.1	Resultados de Latência	46
6.1.2	Consumo de Recursos do Sistema	46
6.1.3	Distribuição e Expressividade dos Eventos	46
6.2	Estabilidade e Avaliação Qualitativa	47
6.3	Teste Piloto de Usabilidade: Estudo de Caso	47
6.3.1	Metodologia do Teste e Curva de Aprendizagem	48
6.3.2	Feedback Qualitativo e Limitações Identificadas	49
7	CONSIDERAÇÕES FINAIS	50
7.1	Síntese dos Resultados	50
7.2	Limitações do Estudo	51
7.3	Trabalhos Futuros	51
	REFERÊNCIAS	53

1 INTRODUÇÃO

Nas últimas décadas, a interação humano-computador tem-se orientado cada vez mais para interfaces que ofereçam modos de uso naturais e intuitivos, movendo-se além de paradigmas baseados em teclados e *mouse* para explorar a entrada gestual e multimodal (ROGERS; SHARP; PREECE, 2023). Paralelamente, a música digital consolidou-se como campo de investigação e aplicação prática, com o surgimento de Instrumentos Musicais Digitais que possibilitam novas formas de criação e execução musical (*performance*) (ROADS, 1996; MIRANDA, 2002). Nesse contexto, as interfaces gestuais musicais emergem como convergência promissora entre IHC (Interface Humano-Computador) e computação musical, propiciando experiências em que movimentos corporais ou de mãos se traduzem diretamente em eventos sonoros (WANDERLEY *et al.*, 2000).

Com o avanço de *frameworks* acessíveis voltados à visão computacional, como o MediaPipe, e com a disponibilidade de protocolos de alta resolução para controle musical, em especial o padrão MIDI 2.0 (*Musical Instrument Digital Interface*), abre-se a possibilidade de desenvolver sistemas que utilizem *hardware* comum. O uso de câmeras de vídeo baseadas no modelo de cores RGB (*Red, Green, Blue*) e computadores de uso geral permite, assim, oferecer interações musicais ricas e dinâmicas (LUGARESI *et al.*, 2019; MIDI Manufacturers Association, 2025).

Ainda que existam diversas propostas acadêmicas e comerciais de interfaces gestuais musicais, persistem barreiras que limitam tanto o acesso quanto a expressividade desses sistemas. Muitas soluções demandam *hardware* especializado e custoso, apresentam latência (tempo de atraso na resposta) que compromete a sensação de fluidez ou dispõem de resolução de controle insuficiente para atender ao desejo de uma expressão musical rica e detalhada. Além disso, a complexidade técnica inerente a essas interfaces pode afastar usuários sem formação musical ou sem experiência em tecnologias avançadas. Indivíduos com mobilidade reduzida frequentemente enfrentam dificuldades adicionais ao operar instrumentos convencionais ou mesmo soluções digitais que requerem motricidade fina. Diante desse panorama, revela-se a lacuna principal que orientou este trabalho: a necessidade de conceber um hiperinstrumento musical gestual que, apoiado em visão computacional por *webcams* e adotando o protocolo MIDI 2.0, fosse simultaneamente acessível, de baixa latência e capaz de oferecer expressividade profissional.

O objetivo geral deste trabalho consistiu em desenvolver e avaliar um hiperinstrumento denominado Aerochord, que utiliza controle gestual via câmera RGB para geração de eventos MIDI 2.0 com resolução estendida, de modo a democratizar a expressão musical e atender a usuários sem treinamento formal. Para tanto, o projeto cumpriu os seguintes objetivos específicos:

- Revisar o estado da arte em interfaces musicais gestuais, identificando paradigmas de interação, tecnologias de captura e lacunas existentes;
- Especificar requisitos funcionais e não funcionais para o Aerochord, enfatizando acessibilidade, latência e expressividade;
- Definir e implementar uma arquitetura modular de *software*, baseada em visão computacional e comunicação via MIDI 2.0, orientada ao desempenho em tempo real;
- Elaborar mapeamentos entre os gestos e os sons que equilibrem intuitividade e profundidade expressiva;
- Executar a validação técnica do sistema por meio de testes de desempenho e medição de latência.

A justificativa para este projeto é multidimensional. Do ponto de vista social, buscou-se ampliar o acesso à criação musical ao oferecer uma alternativa de baixo custo que dispensa dispositivos especializados, favorecendo pessoas sem conhecimento técnico em música e indivíduos com limitações motoras. Do ponto de vista acadêmico e tecnológico, preencheu-se a lacuna identificada no estado da arte por meio da combinação de visão monocular acessível (utilização de uma única lente de câmera), arquitetura de *software* de baixa latência na linguagem C++ em conjunto com o *framework* JUCE, e o uso de MIDI 2.0 para integração com as principais Estações de Trabalho de Áudio Digital (*Digital Audio Workstations* - DAWs). Essa combinação evidencia inovação ao propor um sistema que une facilidade de uso a requisitos de execução musical de nível profissional. Ademais, o Aerochord insere-se no campo de IHC como um caso de uso prático para técnicas de reconhecimento gestual em tempo real, contribuindo com implementações que podem orientar desenvolvimentos futuros.

Para conduzir esta pesquisa e o subsequente desenvolvimento, adotou-se uma abordagem dividida em três frentes: a revisão bibliográfica para fundamentação teórica em bases como *Google Scholar*, na Conferência Internacional sobre Novas Interfaces para Expressão Musical (*New Interfaces for Musical Expression* - NIME) e na Biblioteca Digital da *Association for Computing Machinery* (ACM); a engenharia de *software*, englobando a implementação estruturada de captura de imagem, processamento de inteligência artificial e geração de eventos musicais (MIDI); e, por fim, a etapa de avaliação experimental, voltada à coleta e análise de métricas de desempenho e tempo de resposta do sistema.

O presente documento está estruturado da seguinte forma: no Capítulo 2, apresentam-se os conceitos fundamentais de IHC e hiperinstrumentos que sustentam a proposta; no Capítulo 3, detalham-se os aspectos de computação musical e o protocolo MIDI, justificando a adoção do MIDI 2.0; o Capítulo 4 examina criticamente o estado da arte em interfaces gestuais musicais e identifica a lacuna que motiva o Aerochord; o Capítulo 5 descreve em detalhes a arquitetura modular desenvolvida e a implementação do mapeamento entre gesto e som; o Capítulo 6

apresenta a metodologia de testes aplicada, juntamente com a análise empírica dos resultados obtidos em termos de latência e estabilidade; por fim, o Capítulo 7 conclui o trabalho, resumizando as contribuições alcançadas, discutindo as limitações encontradas e apontando direções para trabalhos futuros.

2 INTERAÇÃO GESTUAL E HIPERINSTRUMENTOS

Neste capítulo, a fundamentação teórica que orientou o projeto (*design*) e os objetivos do Aerochord é aprofundada. Inicialmente, na seção 2.1, o campo da Interação Humano-Computador (IHC) é explorado, destacando-se o paradigma da interação gestual como uma modalidade de controle expressiva e natural, capaz de resgatar a corporeidade e a nuance da execução musical (*performance*). Em seguida, na seção 2.2, são examinadas as principais ferramentas e tecnologias de detecção de gestos, com ênfase em abordagens de visão computacional que fundamentaram a solução técnica adotada no projeto. Por fim, na seção 2.3 apresenta-se e discute-se o conceito de hiperinstrumento, *framework* que define a natureza do instrumento materializado pelo Aerochord: um sistema musical gestual de alta expressividade, acessível e extensível. Assim, é apresentado um panorama organizado das bases teóricas e tecnológicas que sustentam o desenvolvimento do projeto, mostrando como os temas abordados — a interação gestual na IHC, as técnicas de detecção e o conceito de hiperinstrumento — se relacionam para fundamentar a arquitetura da solução.

2.1 INTERAÇÃO MULTIMODAL E GESTUAL

A área de Interação Humano-Computador (IHC) tem como objetivo superar as restrições impostas por interfaces convencionais, como teclado e *mouse*, ao buscar meios de interação mais naturais, intuitivos e centrados nas capacidades humanas. À medida que sistemas computacionais se integram de forma crescente ao cotidiano, surge a demanda por interfaces que reflitam a fluidez e a riqueza da comunicação interpessoal, reduzindo a distância entre a intenção do usuário e a resposta do sistema. Esse contexto justificou a investigação de modalidades gestuais no projeto Aerochord, por se tratar de uma abordagem que aproxima a interação digital dos modos como as pessoas se comunicam entre si (ROGERS; SHARP; PREECE, 2023).

2.1.1 O Paradigma da Interação Multimodal

Sistemas multimodais são aqueles capazes de processar, de maneira coordenada, múltiplas modalidades de entrada, como fala, gestos, olhar e toque, de modo que cada canal contribua para uma interpretação mais precisa da intenção do usuário. Por meio de fusão de sinais e integração temporal e semântica, esses sistemas combinam fluxos sensoriais distintos para aumentar a robustez e a adaptabilidade da interface. Isso permite que falhas ou ambiguidades em um modo sejam compensadas por outro.

Essa coordenação entre canais também oferece flexibilidade, pois o usuário pode escolher a modalidade mais conveniente conforme o contexto e suas preferências. Além disso, promove maior naturalidade ao refletir padrões típicos da comunicação entre pessoas (OVIATT

et al., 2018; OVIATT *et al.*, 2019). Interfaces multimodais também conseguem se adaptar dinamicamente a mudanças ambientais, alternando ou combinando modos de entrada para manter a eficácia da interação. Esses fatores justificam o estudo aprofundado de modalidades específicas, como a gestual, dentro de um quadro multimodal mais amplo.

A multimodalidade confere vantagens centrais em termos de usabilidade: flexibilidade para o usuário selecionar o canal mais adequado, robustez ao mitigar falhas em uma modalidade por meio de outra, e maior naturalidade ao espelhar padrões comunicativos humanos. Tais benefícios reforçam a importância do estudo de interfaces gestuais, mesmo em projetos que foquem primariamente nessa modalidade, pois elas se inserem num ecossistema de interação que visa experiências mais ricas e resilientes (OVIATT *et al.*, 2019).

2.1.2 A Expressividade da Interação Gestual na Música

A interação gestual *mid-air* (gestos realizados no ar) refere-se a movimentos livres de mãos e braços no espaço, sem contato físico com superfícies, proporcionando potencial expressivo elevado, mas impondo desafios ergonômicos, perceptuais e técnicos. Entre esses desafios estão a fadiga muscular decorrente de gestos prolongados, a ausência de *feedback* tátil natural, as variações de iluminação e cenário que afetam a detecção, e a necessidade de latência muito baixa para manter a sensação de controle imediato. Além disso, torna-se necessário incorporar *feedback* audiovisual ou háptico artificial para confirmar o reconhecimento dos gestos (AIGNER *et al.*, 2012). Esses aspectos exigem projetos cuidadosos de detecção, filtragem de ruído e mecanismos de confirmação, de modo a preservar a fluidez e a confiança do usuário na interface gestual.

Contudo, habilitar controle por gestos não é suficiente; é imprescindível que o repertório gestual seja intuitivo e fácil de aprender. Metodologias formais de eliciação de gestos para interação *mid-air* são aplicadas com o objetivo de identificar movimentos que os usuários associam espontaneamente a determinadas ações. Esse raciocínio orientou o mapeamento no Aerochord, garantindo que os gestos implementados coincidisse com expectativas naturais e se mantivessem ergonômicos, evitando movimentos excessivamente amplos ou complexos que pudessem comprometer a experiência (VOGIATZIDAKIS; KOUTSABASIS, 2018).

No contexto musical, gestos e som estão intrinsecamente conectados. Em instrumentos tradicionais, a execução sonora já se baseia em atos gestuais refinados, nos quais movimentos sutis determinam nuances de timbre, dinâmica e articulação. Controlar sistemas digitais por meio de gestos busca resgatar essa corporeidade, permitindo manipulações contínuas e multidimensionais de vários parâmetros sonoros (posição, velocidade, aceleração, trajetória e intensidade) que interfaces convencionais dificilmente capturam com fidelidade. Estudos na área de controle gestual musical mostram que a riqueza e a precisão dos dados capturados influenciam diretamente a expressividade do intérprete, reforçando que a modalidade gestual é especialmente poderosa e natural para expressão sonora (WANDERLEY, 2001; CADOZ; WANDERLEY,

2000).

Em síntese, os princípios de IHC e multimodalidade discutidos nesta seção reforçam a importância de interfaces mais humanas. Dentre essas, a interação gestual *mid-air* se destaca como particularmente relevante para a música digital, pois tem o potencial de recuperar nuances e a corporeidade típicas da performance tradicional. É com base nessa fundamentação teórica que o Aerochord propôs e implementou um sistema de controle gestual via visão computacional, criando uma experiência musical intuitiva e acessível. Na seção seguinte, são detalhadas as tecnologias de detecção de gestos que viabilizam essa interação.

2.2 DISPOSITIVOS E FERRAMENTAS PARA DETECÇÃO DE GESTOS

Capturar e interpretar gestos humanos de forma precisa constitui um dos principais desafios na IHC, pois envolve traduzir movimentos físicos contínuos e tridimensionais em dados digitais compreensíveis por um sistema computacional. Conforme discutido por Preece, Sharp e Rogers (2023), a complexidade dessa tarefa decorre da variabilidade natural dos usuários, das condições ambientais e da necessidade de resposta em tempo real para aplicações interativas. Para “sensoriar” gestos, diversas tecnologias foram desenvolvidas, cada qual com benefícios e limitações próprias no que tange à precisão, ao custo e à dependência de *hardware* específico. É fundamental, portanto, compreender essas categorias de solução a fim de situar a escolha para o projeto Aerochord dentro de um contexto de acessibilidade e robustez.

Uma das abordagens pioneiras e mais influentes baseia-se em câmeras de profundidade, que utiliza luz infravermelha para medir a distância de cada ponto da cena até o sensor, permitindo segmentar o usuário do fundo e estimar a posição tridimensional das articulações. O Microsoft Kinect (Figura 1) tornou-se referência ao empregar essa técnica para rastreamento esquelético em tempo real. O método proposto por Shotton et al., baseado em florestas aleatórias (*random forests*) treinadas com grandes volumes de dados sintéticos, possibilita classificar cada pixel da imagem de profundidade em partes do corpo, gerando estimativas rápidas e precisas das juntas corporais (SHOTTON *et al.*, 2013). Apesar de sua eficácia, essa solução demanda *hardware* específico (sensor de profundidade dedicado) e, embora tenha impulsionado pesquisas em IHC, limita-se em termos de acessibilidade e portabilidade para projetos que buscam operar com dispositivos amplamente disponíveis.

Figura 1 – Dispositivo Microsoft Kinect, modelo do console Xbox 360



Fonte: Disponível em: <<https://pt.wikipedia.org/wiki/Kinect>>. Acesso em 23 jun. 2025

Outra categoria relevante envolve controladores dedicados ao rastreamento manual,

como o Leap Motion Controller (Figura 2). Zhang et al. (2019) demonstram sua aplicação em sistemas de teleoperação baseados em gestos, evidenciando a capacidade do dispositivo em detectar movimentos das mãos e dedos com alta precisão e baixa latência para aplicações em controle de robôs. Embora esse tipo de dispositivo ofereça alto desempenho para detecção de gestos precisos, ele também implica custo adicional de *hardware* especializado, além de restringir o espaço de interação ao volume monitorado pelo sensor, o que contrasta com a meta do Aerochord de empregar soluções de baixo custo e maior flexibilidade.

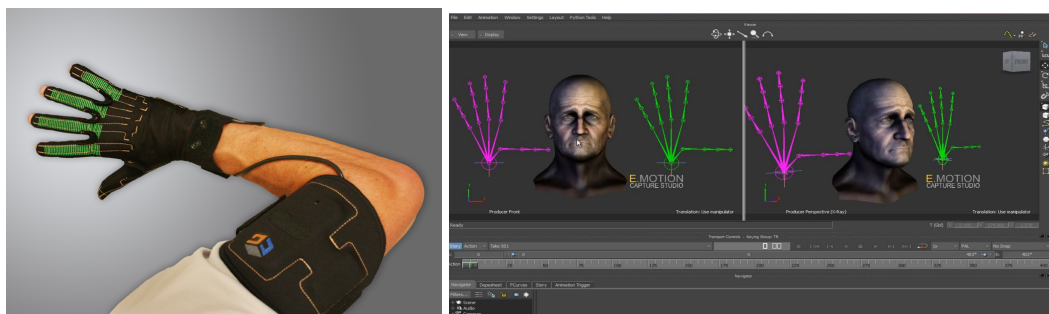
Figura 2 – Controlador Leap Motion em uso para rastreamento de movimentos das mãos



Fonte: Disponível em: <<https://documentation.ls-group.fr/interact/LeapMotion/>>. Acesso em 23 jun. 2025

No campo de sensores vestíveis (*wearables*), destacam-se as chamadas luvas de dados (*data gloves*) (Figura 3), que integram sensores de flexão e Unidades de Medição Inercial (IMUs) para medir diretamente os movimentos das mãos. A revisão feita por Dipietro et al. (2008), aponta como pontos positivos a alta fidelidade na captura de ângulos articulares e ausência de problemas de oclusão visual, mas também ressalta desvantagens como o custo elevado, necessidade de calibração constante e caráter invasivo da interface, que pode prejudicar a naturalidade da interação. Essa categoria, embora tecnicamente robusta, também se afasta do objetivo do Aerochord de oferecer expressão musical sem impor acessórios ao usuário.

Figura 3 – Luva de dados CyberGlove III projetada pela CyberGlove Systems



(a) Sensor vestível CyberGlove III (b) *Software* interpretando os comandos enviados pela CyberGlove III

Fonte: Disponível em: <<https://www.cyberglovesystems.com/cyberglove-iii/>>. Acesso em 23 jun. 2025

Adicionalmente, avanços em inteligência artificial e visão computacional têm viabilizado o rastreamento de gestos utilizando apenas câmeras RGB comuns, como *webcams*. Rautaray e Agrawal (2012) apresentam uma revisão de técnicas de reconhecimento de gestos baseadas em visão, destacando fases de detecção, rastreamento e reconhecimento que exploram desde segmentação de pele até modelos de *deep learning* para extrair características das mãos em imagens bidimensionais (2D). Essa abordagem mostra-se promissora para aplicações acessíveis, uma vez que aproveita *hardware* amplamente disponível e avanços em algoritmos que toleram variações de iluminação e cenários diversos, conectando-se diretamente ao objetivo do Aerochord de democratizar o controle gestual musical.

Entre as ferramentas de visão computacional para detecção de gestos, destaca-se o MediaPipe, *framework open source* desenvolvido pelo Google, que oferece esteiras de processamento (*pipelines*) pré-configuradas de *machine learning* para rastreamento de mãos e poses corporais em tempo real. Conforme descrito por Lugaresi et al. (2019), o MediaPipe permite construir grafos de processamento de vídeo que interpretam o fluxo capturado pela *webcam* e entregam, em tempo real, coordenadas tridimensionais (3D) estimadas dos pontos-chave das mãos, com desempenho otimizado para várias plataformas. A documentação oficial do projeto também fornece instruções detalhadas para a personalização e integração desses módulos em aplicações específicas (Google, 2025). Por combinar alta precisão, baixa latência e depender apenas de *hardware* onipresente, o MediaPipe consolidou-se como a base tecnológica do Aerochord, atendendo perfeitamente aos requisitos de expressividade musical e acessibilidade pretendidos.

2.3 HIPERINSTRUMENTOS MUSICAIS

A evolução dos instrumentos musicais na era digital pode ser entendida como uma extensão da longa história de refinamento das interfaces físicas entre o artista e o som, agora estendendo-se para além das restrições mecânicas e acústicas tradicionais em direção a sistemas computadorizados que ampliam as possibilidades expressivas. Conforme argumentado por Wanderley (2001), instrumentos digitais não sofrem as mesmas limitações físicas de tubos, cordas ou membranas, permitindo uma diversidade enorme de métodos de produção sonora. No entanto, para que esses sistemas ofereçam controle refinado comparável ao das famílias acústicas clássicas, estratégias de projeto cuidadosas de mapeamento e de interface devem ser definidas.

2.3.1 Definição e Características

Em 1986, Tod Machover fundou o grupo Hyperinstruments no *MIT Media Lab*, formalizando o conceito de hiperinstrumento como uma extensão tecnológica da prática musical. Segundo o autor, esse tipo de sistema é composto por três elementos fundamentais: (1) uma

interface sensorial para o *performer*, capaz de capturar gestos, toques ou outras ações; (2) um módulo de processamento, que analisa e interpreta os dados de performance em tempo real; e (3) um sistema de geração ou modulação sonora, que pode envolver síntese digital, modelagem física ou controle de sintetizadores via protocolos como MIDI (MACHOVER, 1992). No contexto do Aerochord, esses três componentes alinham-se da seguinte forma: a interface são as mãos do usuário capturadas por visão computacional; o módulo de processamento é o *software* em C++ que extrai pontos-chave e gestos (via MediaPipe); e o gerador sonoro pode ser qualquer sintetizador compatível com MIDI, preferencialmente MIDI 2.0 acionado via *Universal MIDI Packets*, com contingência para MIDI 1.0.

Machover (1992) enfatiza também a filosofia de aumentação em vez de mera substituição: não se trata apenas de criar timbres novos, mas de capturar nuances da performance humana (velocidade, aceleração, pressão, gestos sutis) que muitas vezes ficam fora do alcance de controles convencionais, e usar esses dados para controlar parâmetros sonoros de forma rica e multidimensional. Essa perspectiva de ampliar a expressividade natural do *performer* reflete o objetivo de hiperinstrumentos de explorar dimensões de controle além do que um instrumento acústico isolado permitiria, usando o poder computacional para traduzir sutilezas gestuais em transformações sonoras complexas.

2.3.2 Evolução e Exemplos Notáveis

Historicamente, desvincular o gesto da produção sonora mecânica precede a era dos computadores. Instrumentos como o Theremin (Figura 4), criado em 1920, e o Ondes Martenot, desenvolvido em 1928, romperam com a necessidade de contato físico direto, permitindo o controle de altura e volume ou de timbre e efeito de modulação de frequência (*vibrato*) por gestos. No caso do Theremin, a proximidade das mãos em relação às antenas determinava as frequências geradas, enquanto o Ondes Martenot combinava um teclado com um anel deslizante para manipulação contínua de som. Esses instrumentos pioneiros anteciparam, em termos conceituais, a performatividade expandida dos hiperinstrumentos computadorizados, ao sugerirem novas formas de mediação gestual e controle expressivo (MANNING, 2013).

Figura 4 – Léon Theremin, criador do Theremin, demonstrando seu instrumento



Fonte: Disponível em: <https://en.wikipedia.org/wiki/Leon_Theremin>. Acesso em 23 jun. 2025

A formalização do termo “hiperinstrumento” ocorreu em 1986, com o trabalho do grupo Hyperinstruments de Tod Machover. Um dos projetos mais emblemáticos desse período foi o Hypercello (Figura 5), desenvolvido em 1991 por Machover em colaboração com o violoncelista Yo-Yo Ma, em que sensores no arco e no violoncelo capturavam parâmetros de performance (como pressão, velocidade e posição) e os convertiam em dados interpretados em tempo real para modificar aspectos do som, incluindo timbre, espacialização e efeitos. Esse modelo demonstra a aplicação prática dos três componentes definidos por Machover: o intérprete interage fisicamente (sensores no instrumento acústico), o computador processa os dados em tempo real, e o sistema sonoro responde de acordo com mapeamentos definidos, muitas vezes enriquecidos por algoritmos de síntese e processamento que ampliam o leque expressivo do instrumento original (MACHOVER, 1992).

Em complemento aos projetos iniciais de Machover, Manning (2013) descreve também o *Hyperbow Controller* (Figura 6), um arco de violino desenvolvido para uso com um violino elétrico. Esse dispositivo incorpora extensômetros (*strain gauges*) que medem as forças aplicadas no arco e acelerômetros que capturam variações de aceleração e inclinação durante o movimento. Essas informações gestuais são convertidas em múltiplos sinais de controle, que são processados em tempo real para modificar parâmetros como timbre, dinâmica e espacialização, sem comprometer a interação física familiar ao violinista. O *Hyperbow* exemplifica a filosofia de aumentação em hiperinstrumentos, ampliando as possibilidades expressivas do gesto de arco tradicional ao traduzi-lo em comandos sonoros multidimensionais via síntese digital ou processamento computacional.

Figura 5 – Tod Machover e seu Hypercello



Fonte: Disponível em: <<https://opera.media.mit.edu/projects/trilogy.html>>. Acesso em 23 jun. 2025

Figura 6 – Hyperbow sendo utilizado em um violoncelo



Fonte: Disponível em: <https://www.researchgate.net/figure/Hyperbow-and-acoustic-acoustic-cello-C-Roberto-Aimi-Interestingly-although-the_fig1_221164872>. Acesso em 23 jun. 2025

Em tempos mais recentes, surgiram exemplos que empregam interfaces tangíveis e visuais para enriquecer a performance musical. Um exemplo é o Reactable (Figura 7), desenvolvido na *Universitat Pompeu Fabra*, que consiste em uma superfície interativa sensível ao toque e à manipulação de objetos físicos, com visualização em tempo real para múltiplos usuários colaborarem musicalmente. Essa plataforma une o gesto manual direto a *feedback* audiovisual, configurando uma forma contemporânea de hiperinstrumento que amplia a expressividade e a interação social em performance ao vivo (BOVERMANN *et al.*, 2016). Embora existam também controladores gestuais baseados em *hardware* dedicado, a ênfase recente em visão computacional e sensores amplamente disponíveis reflete a busca por acessibilidade, alinhando-se diretamente ao projeto Aerochord, que visa empregar câmeras comuns para capturar gestos e oferecer controle musical sensível sem exigir dispositivos especializados.

Figura 7 – Reactable



Fonte: Disponível em: <<https://en.wikipedia.org/wiki/Reactable>>. Acesso em 23 jun. 2025

2.3.3 Hiperinstrumentos e Interação Gestual

A interação gestual ocupa papel central no *design* de hiperinstrumentos, pois permite capturar a expressividade natural do corpo do *performer* de modo contínuo e multidimensional. Conforme argumenta Wanderley (2001), o gesto musical, especialmente em instrumentos acústicos, já é uma forma de controle multiparamétrico, envolvendo aspectos como sincronização rítmica (*timing*), articulação e dinâmica. Transportar essa riqueza para interfaces digitais exige que o sistema capture variações sutis de velocidade, aceleração e posição, assegurando um mapeamento claro e responsivo entre gesto e som.

No Aerochord, a adoção da visão computacional com MediaPipe exemplificou perfeitamente essa proposta. Ao extrair coordenadas 3D das mãos em tempo real, o sistema preserva a latência e a precisão necessárias para a expressividade musical. Entretanto, o verdadeiro desafio superado esteve no *mapeamento*: isto é, na definição de regras que traduzem os dados gestuais em parâmetros sonoros de forma musicalmente coerente e criativa. Wanderley (2001) destaca que um mapeamento eficaz deve respeitar propriedades perceptivas e ergonômicas, oferecer consistência e previsibilidade (para aprendizado do *performer*), mas também permitir explorabilidade para descobertas sonoras. No contexto de hiperinstrumentos, o mapeamento implementado no projeto não foi apenas linear, envolvendo fusão de múltiplas dimensões gestuais que dialogam diretamente com as extensões proporcionadas pelo MIDI 2.0.

2.4 CONSIDERAÇÕES SOBRE O CAPÍTULO

Ao longo deste capítulo, fundamentou-se o projeto Aerochord em pilares teóricos que se articulam de modo coeso: inicialmente, a partir dos princípios da IHC que enfatizam a busca por interfaces mais naturais e centradas no usuário; em seguida, pela caracterização da interação gestual como uma modalidade de controle especialmente poderosa e expressiva; e, finalmente, pela concepção dos hiperinstrumentos como modelo ideal para explorar essa expressividade no domínio musical. Essa jornada conceitual reforça que o Aerochord não é uma iniciativa isolada, mas sim o resultado de uma proposta fundamentada em princípios consolidados da computação musical e da IHC.

A relevância contemporânea desse enquadramento revela-se na convergência entre a ambição histórica dos hiperinstrumentos de Machover, originalmente dependentes de sensores dedicados, *hardware* especializado e ambientes de pesquisa de alto custo, e o advento de ferramentas de visão computacional *open source*, como o MediaPipe, que operam com *hardware* genérico (por exemplo, uma simples *webcam*). Enquanto Machover (1992) descreveu hiperinstrumentos cuja expressividade exigia infraestrutura complexa, o panorama atual oferece camadas de percepção corporal em tempo real, robustas e acessíveis (LUGARESI *et al.*, 2019). Foi precisamente nessa interseção (entre o conceito expressivo de hiperinstrumento e a democratização tecnológica) que o Aerochord se estabeleceu, viabilizando a mesma amplitude de

controle de projetos pioneiros de maneira acessível a criadores sem necessidade de equipamentos exclusivos.

Os hiperinstrumentos representam a convergência de pesquisas em interação gestual, processamento de sinais e *design* de interface musical, cujo objetivo é ampliar as capacidades expressivas do *performer*. Desde pioneiros como Theremin e Ondes Martenot até plataformas modernas baseadas em visão computacional e interfaces tangíveis (como o Reactable), observa-se um fio condutor claro: a busca por interfaces que respondam ao corpo do artista com sensibilidade e precisão enquanto oferecem liberdade criativa ampliada pelo poder computacional. As escolhas de tecnologia realizadas no Aerochord foram integralmente orientadas pelos princípios da expressividade musical e da usabilidade performática.

Entretanto, mesmo com o suporte teórico consolidado e a disponibilidade de tecnologia acessível, o sucesso de qualquer hiperinstrumento gestual repousa no desafio central do mapeamento gestual, isto é, da tradução entre movimento corporal e transformação sonora. Como enfatizado em Wanderley 2001, o mapeamento constitui o “coração” de um instrumento digital gestual: ele deve ser suficientemente intuitivo para iniciantes e profundo o bastante para permitir aos músicos experientes explorar dimensões expressivas mais sutis (velocidade, aceleração, forma de movimento, variações dinâmicas).

O Capítulo 5 detalha como o Aerochord abordou e superou esse desafio na prática, descrevendo a arquitetura de *software* implementada e as estratégias projetadas para equilibrar usabilidade e riqueza expressiva.

3 COMPUTAÇÃO E MÚSICA

Este capítulo apresenta a base teórica que serviu de apoio para o desenvolvimento da solução computacional do Aerochord. Na Seção 3.1, discutem-se os fundamentos da computação musical, incluindo representação digital do som, síntese sonora e processamento de áudio digital. Em seguida, a Seção 3.2 detalha o protocolo MIDI (Interface Digital para Instrumentos Musicais, do inglês *Musical Instrument Digital Interface*), apresentando seu propósito, funcionamento e as mensagens mais relevantes para o controle expressivo do sistema. Por fim, na Seção 3.3, revisam-se os principais conceitos de teoria musical aplicados ao mapeamento gestual — escala cromática, oitavas, articulações (notas curtas e destacadas, *staccato*; ou notas fluidas e ligadas, *legato*) e efeitos como modulação de afinação e frequência (*pitch bend* e *vibrato*), sustentação (*sustain*) e dinâmica — que fundamentaram o *design* do instrumento.

3.1 FUNDAMENTOS DA COMPUTAÇÃO MUSICAL

A computação musical concentra-se na aplicação de algoritmos e modelos computacionais para criar, analisar e modificar sons de maneira flexível e precisa. Conforme destacado por Miranda (2002), esse campo abrange desde a geração programada de timbres até a modelagem de estruturas musicais complexas e sistemas de execução (*performance*) interativa. Dodge e Jerse (1997) enfatizam o papel do computador na realização de tarefas sonoras avançadas, enquanto Moore (1990) evidencia a importância da representação digital do som, permitindo seu armazenamento e processamento eficientes.

O processo inicia-se com a conversão de um sinal analógico em dados numéricos, por meio de dois estágios principais: a amostragem, que mede a amplitude do sinal em instantes regulares, e a quantização, que mapeia essas medições para valores discretos (ROSSING, 2002). Essa sequência de valores numéricos serve de base para todas as operações subsequentes.

A síntese sonora gera sons por meio de modelos computacionais em vez de gravações acústicas. Na síntese aditiva, ondas senoidais puras são somadas para formar timbres; na síntese subtrativa, sinais ricos em harmônicos são filtrados para moldar o espectro sonoro. A modulação de frequência (FM) altera dinamicamente a frequência de uma portadora por uma moduladora, produzindo timbres expressivos, enquanto a síntese granular emprega pequenos fragmentos de áudio (os “grãos”) para criar texturas sonoras singulares (MANNING, 2013).

O processamento de áudio digital manipula gravações pré-existentes por meio de técnicas como filtragem, equalização, compressão, reverberação e aplicação de efeitos especiais, aprimorando a qualidade sonora e possibilitando transformações criativas (ZÖLZER, 2011). Essas operações são largamente empregadas em Estações de Trabalho de Áudio Digital (*Digital Audio Workstations* - DAWs) para modelar o som conforme a intenção artística.

As aplicações da computação musical incluem a criação de instrumentos digitais equipados com sensores e controladores, a composição assistida por computador que apoia o desenvolvimento de ideias melódicas e estruturais, e sistemas interativos que respondem em tempo real aos gestos do intérprete (*performer*). Adicionalmente, métodos de análise e recuperação de informação musical permitem indexar e buscar repertórios com base em atributos como melodia, ritmo e timbre (COPE, 2001).

No contexto deste trabalho, esses fundamentos habilitaram a captação e interpretação de gestos das mãos e seu mapeamento para mensagens MIDI coerentes. A etapa de quantização assegurou precisão na conversão dos sinais de movimento, enquanto a síntese e o processamento digital compõem o ambiente sonoro no *software* receptor. Ao integrar esses elementos, o hiperinstrumento desenvolvido amplia as possibilidades expressivas do intérprete, tornando o gesto das mãos parte ativa da criação musical.

3.2 O PROTOCOLO MIDI (MUSICAL INSTRUMENT DIGITAL INTERFACE)

Desde seu surgimento em meados de 1983, o protocolo MIDI estabeleceu-se como padrão onipresente para comunicação de eventos musicais entre controladores, sintetizadores e sequenciadores, permitindo que equipamentos de fabricantes distintos interagissem de forma consistente e interoperável (AVANZINI; FASCHI; LUDOVICO, 2023; Association of Musical Electronics Industry; MIDI Manufacturers Association, 1996). Essa universalidade viabilizou a criação de um ecossistema diversificado de *hardware* e *software*, moldando práticas de produção, *performance* e composição musical ao longo de décadas. Porém, à medida que as demandas por expressividade mais refinada, controle de alta precisão e interação em tempo real evoluíram, especialmente em contextos de *performance* gestual, tornaram-se evidentes limitações do MIDI 1.0, tais como resolução restrita de parâmetros, comunicação predominantemente unidirecional e ausência de mecanismos de autoconfiguração (MIDI Manufacturers Association; Association of Musical Electronics Industry, 2023c).

Em cenários de interfaces musicais gestuais de baixa latência e alta expressividade, como o projeto Aerochord, essas restrições tornam-se especialmente críticas: o mapeamento fino de gestos a parâmetros sonoros exige curvas de resposta suaves, confirmação de capacidades do dispositivo receptor e sincronização precisa de eventos. Foi essa necessidade que motivou a adoção do MIDI 2.0 no projeto, cuja arquitetura revisada introduz inovações concebidas para atender a tais exigências sem abandonar a compatibilidade retroativa com o ecossistema existente. Nesta seção, são apresentados inicialmente os fundamentos e as limitações do protocolo MIDI 1.0. Em seguida, discute-se a arquitetura técnica e as novas capacidades do MIDI 2.0, demonstrando como elas fundamentaram a escolha desse protocolo como pilar tecnológico do Aerochord.

3.2.1 Fundamentos e Limitações do MIDI 1.0

O protocolo MIDI 1.0 opera como um sistema de transmissão de instruções musicais, transportando mensagens que indicam eventos sem veicular o áudio propriamente dito (AVANZINI; FASCHI; LUDOVICO, 2023; Association of Musical Electronics Industry; MIDI Manufacturers Association, 1996). Os eventos mais comuns incluem ativação e desativação de notas (*Note On* e *Note Off*), intensidade do toque (*Velocity*), mudanças de parâmetros de som (*Control Change*), troca de instrumentos (*Program Change*), dobra de afinação (*Pitch Bend*) e mensagens exclusivas de fabricantes (*System Exclusive* ou *SysEx*). Cada mensagem é composta por um byte de estado (com bit alto indicando tipo de mensagem e canal) seguido de um ou dois bytes de dados (com 7 bits significativos), refletindo a concepção original de comunicação serial a 31,25 kbps. Essa taxa de transferência definia limites para o número de eventos simultâneos que poderiam ser enviados sem comprometer significativamente a latência ou gerar congestionamento na linha.

Ao unificar equipamentos de fabricantes distintos, o MIDI 1.0 estabeleceu um ecossistema interoperável, permitindo que controladores e DAWs compartilhassem mensagens comuns. Essa compatibilidade retroativa sustenta uma vasta base instalada de dispositivos e aplicou-se também ao Aerochord, que implementou a capacidade de enviar mensagens em modo legado a sintetizadores sem suporte a MIDI 2.0, assegurando que usuários com equipamentos existentes possam integrar o sistema sem barreiras.

Entretanto, para aplicações que demandam expressividade refinada e baixa latência a concepção original do MIDI 1.0 revela limitações críticas. A resolução de 7 bits em *Control Changes* e *Velocity* implica apenas 128 passos discretos, frequentemente resultando em saltos audíveis em variações contínuas; o *Pitch Bend* afeta todas as notas de um mesmo canal simultaneamente, impedindo modulações individuais em acordes polifônicos; a comunicação é primordialmente unidirecional; o número restrito de 16 canais por porta limita distinções granulares; e a ausência de *timestamps* embutidos impede compensações automáticas de variações no tempo de chegada dos dados (*jitter*)¹. Esses fatores se tornam especialmente significativos no Aerochord, em que a fluidez do mapeamento de gestos exigiu curvas suaves e sincronização precisa que ultrapassam o escopo do padrão legado (MIDI Manufacturers Association; Association of Musical Electronics Industry, 2023c).

3.2.2 Arquitetura Técnica do MIDI 2.0

O surgimento do MIDI 2.0 representa uma resposta direta às limitações identificadas no protocolo legado, oferecendo compatibilidade retroativa ao introduzir mecanismos bidirecionais que permitem autoconfiguração e descoberta de capacidades entre instrumentos (MIDI Manufacturers Association; Association of Musical Electronics Industry, 2023c). Essa evolução foi moti-

¹ Variação no atraso da transmissão de dados sobre uma rede ou barramento, causando irregularidades e possíveis problemas de sincronização.

vada pela necessidade de suportar demandas de expressividade refinada em tempo real, onde a fluidez gestual é um requisito central.

No cerne da nova arquitetura estão os Pacotes MIDI Universais (*Universal MIDI Packets* - UMP), que redefinem o formato de mensagens para 32, 64 ou 128 bits, com campos que incluem tipo de mensagem, grupo, canal/comando e, opcionalmente, *timestamp* embutido (MIDI Manufacturers Association; Association of Musical Electronics Industry, 2023f). Essa estrutura ampliada possibilita transportar dados de alta resolução e atuar com controladores por nota individual (*per-note controllers*)². No Aerochord, cada evento gestual é marcado com um *timestamp* no momento exato da captura, propagado internamente por todo o *pipeline* até o envio do pacote MIDI. Esse *timestamp* é usado para medir a latência *end-to-end* do sistema (percentis P50/P95) — ele não é serializado como mensagem de *Jitter-Reduction Timestamp* dentro do próprio pacote UMP transmitido ao dispositivo receptor, ficando restrito à instrumentação interna do Aerochord.

A descoberta de capacidades passa a ser realizada por meio da Consulta de Capacidades MIDI (*MIDI Capability Inquiry* - MIDI-CI), em que um dispositivo envia mensagens de busca e recebe detalhes sobre o suporte a MIDI 2.0 e propriedades configuráveis (MIDI Manufacturers Association; Association of Musical Electronics Industry, 2023d). Esse protocolo de negociação inicial (*handshake*) transforma a comunicação "cega" em uma negociação inteligente: o Aerochord identifica automaticamente se o receptor aceita mensagens UMP, dispensando configurações manuais extensas.

Em sequência, os Perfis (*Profiles*) e a Troca de Propriedades (*Property Exchange*) permitem compartilhar conjuntos padronizados de regras. Os perfis definem como interpretar mensagens em contextos específicos (ex: mapeamentos para *per-note controllers*), enquanto o *Property Exchange* possibilita consultar e ajustar dinamicamente parâmetros remotos, como curvas de resposta, sem sair da sessão (MIDI Manufacturers Association; Association of Musical Electronics Industry, 2023a; MIDI Manufacturers Association; Association of Musical Electronics Industry, 2023b).

A arquitetura também prevê flexibilidade de transporte: pacotes UMP podem trafegar via USB, *Bluetooth* ou rede, ao mesmo tempo em que existe um *fallback* para encapsular mensagens em conexões MIDI 1.0 legadas (MIDI Manufacturers Association; Association of Musical Electronics Industry, 2023f). Essa adaptabilidade garante operação em múltiplas plataformas, mantendo a funcionalidade básica mesmo em dispositivos antigos.

Por fim, a especificação prevê até 16 grupos independentes, cada um com 16 canais, permitindo separar fluxos de controle distintos. Como o Aerochord, em sua versão atual, é um instrumento monofônico de voz única — em que notas, volume, sustentação, timbre, vibrato e *pitch bend* moldam o mesmo som —, todos os eventos das duas mãos são roteados para um único grupo e canal. Essa escolha garante que os macrocontroles globais (volume e sustentação,

² Controladores que permitem ajustar parâmetros como volume, afinação e timbre de cada nota individualmente, diferentemente dos controles comuns que afetam todo o canal.

operados pela mão de controle) e as modulações por nota (operadas pela mão de articulação) incidam efetivamente sobre as notas em execução, uma vez que, na especificação, cada grupo constitui uma porta lógica de 16 canais independente. Os grupos adicionais permanecem reservados para a evolução polifônica e multivoz do sistema, discutida entre os trabalhos futuros (Capítulo 7).

3.2.3 Novas Capacidades Expressivas do MIDI 2.0

Ao redefinir o formato e introduzir mecanismos de maior resolução, o MIDI 2.0 abre espaço para ganhos expressivos inimagináveis no padrão legado, os quais se refletiram diretamente no Aerochord (MIDI Manufacturers Association; Association of Musical Electronics Industry, 2023c; MIDI Manufacturers Association; Association of Musical Electronics Industry, 2023f).

A alta resolução representa um avanço fundamental: valores de *Velocity* passam a ter 16 bits e *Control Changes* podem alcançar 32 bits, eliminando saltos audíveis. No Aerochord, onde pequenas alterações de gesto modulam contínua e simultaneamente vários efeitos, essa granularidade estendida assegurou que o mapeamento refletisse nuances sutis sem artefatos (MIDI Manufacturers Association; Association of Musical Electronics Industry, 2023f).

Outra capacidade expressiva é o suporte a controle por nota individual. Diferentemente do MIDI 1.0, o MIDI 2.0 permite enviar mensagens de afinação, modulação ou pressão após o toque (*aftertouch*) para notas específicas dentro de um acorde. O Aerochord adota essa abordagem ao emitir a dobra de afinação como *Per-Note Pitch Bend (UMP MT4 Opcode 0x6)*, direcionada à nota sustentada corrente em vez de afetar o canal inteiro. Na versão atual, monofônica, isso garante que o *bend* incida exatamente sobre a nota em execução; mais importante, prepara a arquitetura para a evolução polifônica prevista nos trabalhos futuros (Capítulo 7), na qual cada nota de um acorde poderá ser modulada de forma independente — algo inviável no padrão legado.

O transporte de blocos estruturados de dados torna-se mais robusto, permitindo enviar configurações e *presets* de sintetizadores em formato padronizado. A padronização desses mecanismos facilita a portabilidade e a experimentação colaborativa, integrando-se à calibração dinâmica em tempo real (MIDI Manufacturers Association; Association of Musical Electronics Industry, 2023b).

3.2.4 Persistência de Performances e Arquivos

Registrar performances gestuais com fidelidade é fundamental para edição e análise detalhada. Em instrumentos como o Aerochord, cada gesto traduzido em MIDI 2.0 contém dados de alta resolução e *timestamps* que documentam a evolução temporal exata dos controles.

O formato *MIDI Clip File*, baseado em UMP, oferece suporte à gravação de sequências contendo mensagens de alta resolução e *timestamps* embutidos (MIDI Manufacturers Association; As-

sociation of Musical Electronics Industry, 2023e). Em complemento, o *Container File* permite agrupar múltiplos clipes e metadados, formando um pacote coeso.

No contexto de avaliação do Aerochord, a persistência de eventos (implementada através do *eval mode*, exportando para arquivos tabulares) e os *logs* precisos de eventos *end-to-end* desempenharam um papel vital na medição de latência e de precisão do sistema.

3.2.5 Justificativa Final para a Arquitetura do Aerochord

Em síntese, as capacidades do MIDI 2.0 — alta resolução, controle por nota individual, autoconfiguração via MIDI-CI e *timestamps* — atenderam diretamente aos requisitos de expressividade e robustez que nortearam o Aerochord (MIDI Manufacturers Association; Association of Musical Electronics Industry, 2023f; MIDI Manufacturers Association; Association of Musical Electronics Industry, 2023d; MIDI Manufacturers Association; Association of Musical Electronics Industry, 2023b; MIDI Manufacturers Association; Association of Musical Electronics Industry, 2023e).

Em termos de aplicação concreta, a distância entre os dedos pôde controlar filtros sonoros suavemente sem saltos audíveis. Ao sustentar uma nota, gestos de oscilação e inclinação geraram modulação contínua direcionada a essa nota (via *Per-Note Pitch Bend*), em vez de afetar todo o canal. Além disso, ao conectar-se internamente, o Aerochord integrou uma porta virtual em modo MIDI 2.0, configurando parâmetros (*setup*) de forma transparente.

O suporte a MIDI 2.0 não foi meramente um recurso opcional, mas o elemento central para viabilizar o controle gestual de alta precisão alcançado. A adoção do padrão potencializou a missão de democratização, oferecendo uma exploração fina com resposta sonora refinada.

3.3 TEORIA MUSICAL APLICADA

Esta seção apresenta os fundamentos da teoria musical que serviram de base para o mapeamento gestual no sistema Aerochord. Serão abordados os conceitos de escala cromática e oitavas (BENWARD; SAKER, 2009). Em seguida, noções de articulação, como *staccato* e *legato* (DUCKWORTH, 2007). Por fim, revisam-se efeitos como *vibrato*, *pitch bend*, *sustain* e dinâmica, implementados a partir das variações nos gestos captados (DODGE; JERSE, 1997).

3.3.1 Escala Cromática e Oitavas

A escala cromática consiste na divisão da oitava em doze semitons iguais, representando a menor unidade de distância entre duas notas no sistema temperado. Cada semitom equivale a um aumento fixo de frequência, criando um sistema padronizado (BENWARD; SAKER, 2009).

No contexto do MIDI, cada nota é representada por um número inteiro de 0 a 127. A nota Dó central (C4 ou MIDI 60) serve como referência, onde aumentos de valor representam semitons sucessivos. As oitavas são formadas por doze semitons (relação de frequência 2:1),

percebidas como uma repetição da mesma nota em um registro mais agudo (DUCKWORTH, 2007). Essa percepção invariável foi explorada no Aerochord por meio dos gestos de seleção.

Do ponto de vista do *design* gestual, o mapeamento suporta usuários destros e canhotos sem qualquer configuração prévia: os papéis de cada mão (controle vs. articulação) são atribuídos pela ordem de entrada em quadro, e não pela lateralidade anatômica do usuário (ver Seção 5.4). A altura vertical do quadro de visão da câmera foi dividida em doze zonas discretas (semitons) ativadas pela **mão de articulação** (a segunda mão a entrar em quadro). Simultaneamente, movimentos verticais macro da **mão de controle** (a primeira a entrar) determinavam a transição entre oitavas. Isso possibilitou ao usuário navegar por registros diferentes de forma ergonômica, escolhendo livremente qual mão usar para cada papel.

3.3.2 Articulações

As articulações musicais determinam como as notas se conectam. O *staccato* indica notas breves e destacadas. No Aerochord, esse efeito é reproduzido por movimentos rápidos e secos, como um fechamento súbito dos dedos seguido de uma liberação imediata, gerando notas curtas (DUCKWORTH, 2007).

Por outro lado, o *legato* indica uma execução fluida. Em termos técnicos, significa a emissão de um *Note On* antes do *Note Off* da nota anterior, criando sobreposição. No Aerochord, isso foi interpretado permitindo o deslizamento da mão de articulação fechada entre as zonas verticais de nota. Essa abordagem exigiu lógica computacional precisa para gerir as fronteiras, utilizando zonas mortas (*dead zones* - áreas de tolerância de entrada) e configurações de estabilização do sinal para evitar ativações duplas indesejadas (BENWARD; SAKER, 2009).

3.3.3 Efeitos de expressividade

A expressividade musical reflete a resposta a nuances da execução. O *vibrato* é a modulação periódica na frequência da nota. No MIDI 2.0, seu controle opera com alta granularidade (MIDI Manufacturers Association, 2025). No Aerochord, esse efeito foi gerado a partir de pequenos movimentos oscilatórios laterais da mão de articulação durante a ativação da nota.

O *pitch bend* altera a afinação de forma contínua. Tratado como *Per-Note Pitch Bend* (atributo por nota) em MIDI 2.0, o sistema interpretou deslocamentos angulares de inclinação da mão de articulação para realizar transições melódicas fluídas, mapeando a rotação diretamente para o valor de afinação da nota em andamento (MIDI Manufacturers Association, 2025).

O *sustain* refere-se à capacidade de manter o som após o término da ação física. A manutenção da **mão de controle** aberta (em repouso vertical temporário) foi mapeada para acionar e desativar o *sustain*, garantindo que as notas ativadas pela **mão de articulação** pudessem permanecer soando de maneira análoga ao pedal de sustentação de um piano.

A dinâmica diz respeito à intensidade sonora (comumente tratada na música tradicional sob termos como *piano*, *forte* e *crescendo*). No protocolo MIDI, a dinâmica do ataque da nota é controlada pelo *velocity*. No projeto, a velocidade de fechamento dos dedos da mão de articulação (distância mensurada em janelas de tempo) foi calculada pelo *software* e mapeada para a intensidade da nota, permitindo ataques percussivos suaves ou agressivos conforme a intenção do usuário (BENWARD; SAKER, 2009).

3.4 CONSIDERAÇÕES SOBRE O CAPÍTULO

Ao longo deste capítulo, foram apresentados os fundamentos da computação musical e explorado o protocolo MIDI. Demonstrou-se como o MIDI 2.0 reverteu obstáculos de baixa resolução por meio de pacotes universais (UMP) e consultas automáticas (MIDI-CI).

A compreensão dessas especificações técnicas constituiu o alicerce sobre o qual a arquitetura de *software* do Aerochord se apoiou. A capacidade de endereçar dados de alta resolução e controlar a afinação (*pitch bend*) de cada nota individualmente foi o elemento-chave que viabilizou o mapeamento fluido implementado.

Com os conceitos de IHC, teoria musical e protocolo MIDI firmemente estabelecidos, o Capítulo 4 aprofundará a revisão da literatura, examinando soluções existentes para delimitar exatamente o cenário onde a proposta de solução (detalhada no Capítulo 5) foi implementada.

4 ESTADO DA ARTE EM INTERFACES MUSICAIS GESTUAIS

Uma análise do estado da arte em interfaces musicais gestuais deve ir além da simples catalogação de tecnologias e dispositivos: é preciso centrar a investigação nos paradigmas de interação que essas soluções viabilizam e nos *trade-offs* implícitos em cada abordagem. Essa perspectiva não se limita a enumerar sensores, bibliotecas ou *frameworks*, mas busca compreender como, historicamente e atualmente, diferentes formas de captação gestual, mapeamento e entrega de *feedback* procuram equilibrar três pilares cruciais: a fidelidade na captura dos gestos, a expressividade resultante no domínio sonoro e a acessibilidade de uso para perfis diversos de usuários.

Este capítulo segue uma narrativa estruturada em etapas complementares. Primeiramente, é explorado como paradigmas clássicos — de sistemas baseados em contato físico dedicado a gestos livres em espaço tridimensional ou a capturas por visão monocular — moldaram o que hoje se entende como possibilidades e limitações em interfaces musicais gestuais. Em seguida, discute-se o desafio central do mapeamento gesto-som, investigando tanto estratégias discretas e contínuas, bem como filosofias de *design* orientadas a priorizar o movimento (*movement-first*) ou priorizar o som (*sound-first*). Na sequência, exploram-se métodos de avaliação, perfis de usuário e cenários de aplicação, situando cada abordagem em contextos reais e em requisitos de usabilidade, latência e expressividade. Por fim, apresenta-se uma síntese crítica das principais soluções existentes, destacando suas potencialidades e limitações, com o objetivo de identificar lacunas e oportunidades que informaram o desenvolvimento prático do Aerochord.

O projeto Aerochord surgiu nesse contexto como uma solução que aproveitou os avanços recentes em Inteligência Artificial e técnicas de processamento em tempo real para maximizar a expressividade musical a partir de gestos, mantendo-se universalmente acessível. Ao longo deste capítulo, a caracterização dos paradigmas vigentes e a análise crítica de suas forças e fraquezas servirão de base para justificar e fundamentar as escolhas de arquitetura, mapeamento e avaliação que distinguem a implementação prática do sistema. Dessa forma, este Estado da Arte estabelece o alicerce conceitual para a justificativa de inovação do projeto e para a definição de suas metas atingidas em termos de fidelidade de captura, qualidade do mapeamento e inclusão de usuários em cenários diversos.

4.1 PARADIGMAS DE INTERAÇÃO EM INTERFACES MUSICAIS GESTUAIS

Nesta seção, analisamos as principais formas de interação gestual em interfaces musicais, agrupadas conforme o tipo de experiência proporcionada ao *performer*. O foco está em

revelar como as tecnologias empregadas moldam diferentes paradigmas de controle expressivo, revelando os compromissos entre fidelidade na captura, potencial expressivo e acessibilidade e como essas lições orientaram o projeto do Aerochord.

4.1.1 Interação de Alta Fidelidade com Contato Físico

O paradigma de alta fidelidade com contato físico é tradicionalmente exemplificado pelas luvas de dados (*data gloves*), que empregam sensores de flexão nos dedos e, frequentemente, Unidades de Medição Inercial (IMUs) para estimar orientação e aceleração espacial (DIPIETRO; SABATINI; DARIO, 2008; ROVAN; WANDERLEY; DUBNOV, 1997). Essa combinação possibilita captar nuances finas, tais como a taxa de curvatura das juntas digitais, a velocidade de flexão e pequenas variações de posição, o que viabiliza mapeamentos multiparamétricos e contínuos sobre síntese sonora ou processamento de efeitos em cenários que exigem controle detalhado de múltiplos graus de liberdade. Em contextos de pesquisa e *performance* profissional, essa fidelidade extrema oferece sensibilidade para traduzir sutilezas de expressão em variações sonoras muito precisas.

Entretanto, o alto custo de *hardware* especializado e a complexidade de calibração — ajustar sensores de flexão para cada mão, compensar derivações de IMUs e atualizar configurações para diferentes execuções — restringem essa abordagem a nichos com recursos dedicados. O caráter invasivo de exigir que o *performer* “vista” o dispositivo impõe barreiras físicas e psicológicas, pois a sensação de restrição das mãos e a fadiga em sessões prolongadas podem comprometer a espontaneidade e o conforto durante a *performance*. Além disso, a dependência de manutenção e a suscetibilidade a falhas de sensor por mau encaixe ou interferência elevam o risco de instabilidade em tempo real.

Neste cenário, percebe-se que as luvas de dados funcionam como um instrumento de precisão extrema. Por outro lado, o Aerochord priorizou acessibilidade e simplicidade, dispensando vestimentas tecnológicas específicas e custos elevados. Ao invés de buscar os mesmos níveis de detalhamento das luvas, o Aerochord concentrou-se em atingir uma fidelidade gestual suficiente para permitir expressividade musical, utilizando visão computacional e aprendizado de máquina com equipamentos comuns. Com isso, reduziu-se a necessidade de calibração complexa e ampliou-se o potencial de uso em contextos diversos.

4.1.2 Gestos Livres em Volume 3D

O paradigma de gestos livres em volume 3D explora o rastreamento corporal em espaço sem contato físico, permitindo movimentos amplos e performáticos. O Microsoft Kinect popularizou essa abordagem ao oferecer rastreamento esquelético completo em tempo real por meio de câmeras de profundidade acessíveis ao consumidor (SHOTTON *et al.*, 2013). Essa tecnologia permitiu que intérpretes acionassem eventos sonoros por gestos expansivos, como movimentos

ampos de braço para modular filtros ou disparar *samples*, conferindo dimensão cênica às apresentações (LEESE; MCDONALD, 2012). Contudo, o Kinect impunha limitações em relação à zona de captura, com restrições de distância mínima e máxima, sensibilidade a condições de iluminação e resolução insuficiente para detalhes finos de mãos e dedos, o que restringe a exatidão em mapeamentos que exijam precisão manual.

Em sequência, dispositivos como o Leap Motion Controller refinaram o paradigma ao focar em um “espaço de trabalho virtual” para rastreamento manual com infravermelho de curto alcance, proporcionando maior precisão nos gestos das mãos (RAUTARAY; AGRAWAL, 2012). Nesse contexto, tornou-se viável conjugar gestos amplos capturados por câmeras com sensor de profundidade com manipulações finas detectadas por sensores dedicados, atendendo a aplicações que requerem simultaneamente controle de parâmetros amplos e sutis. Ainda assim, ambas as abordagens dependem de *hardware* específico, o que implica custos adicionais e limita a flexibilidade de configuração do ambiente de *performance*, além da suscetibilidade a ruído de luz ambiente e restrições físicas de posicionamento.

O Aerochord avançou esse paradigma ao utilizar apenas câmeras RGB padrão, ampliando a flexibilidade da área de *performance*. Algoritmos robustos de detecção de pose monocular (baseada em uma única lente) permitiram reconhecer gestos em espaços variados e sob condições de uso diversificadas, sem exigir sensores proprietários. Dessa forma, manteve-se o espírito de gestos livres em 3D, mas reduziu-se a dependência de *hardware* dedicado, privilegiando acessibilidade e adaptabilidade a diferentes configurações de palco ou estúdio, sem descuidar da necessidade de precisão e confiabilidade de rastreamento para uso musical em tempo real.

4.1.3 Detecção de Pose com Visão Monocular

O paradigma de detecção de pose com visão monocular representa a fronteira mais recente em acessibilidade para interfaces musicais gestuais, viabilizada por avanços em redes neurais profundas e *frameworks* como MediaPipe (LUGARESI *et al.*, 2019). Hoje, é possível estimar posições articulares em 2D e até reconstruções 3D aproximadas a partir de uma única câmera RGB, alcançando precisão crescente para rastrear articulações principais do corpo e detectar *landmarks* de mãos em tempo real. Essa abordagem democratiza o acesso, pois qualquer câmera convencional integrada a computadores portáteis ou *smartphones* pode servir como sensor principal.

Diversos protótipos acadêmicos exploram essa base. Projetos como o Hyper-Harp demonstram sistemas de harpa virtual baseados em rastreamento de pose monocular (AMERICO, 2020). Implementações de controle de som em tempo real por gestos revelam a viabilidade inicial, mas evidenciam desafios de otimização de latência e robustez a variações de iluminação, que podem comprometer a estabilidade do rastreamento em contextos de *performance* ao vivo (HASSAN; DESSOUKY; EL-LATIF, 2019). Estudos preliminares indicam que a latência *end-to-end*

em *pipelines* de visão monocular pode variar tipicamente entre 30 ms e 60 ms em *hardware* comum (LUGARESI *et al.*, 2019; HASSAN; DESSOUKY; EL-LATIF, 2019).

O trabalho Integrando a Dança e a Música em uma Arquitetura de *Deep Learning* ilustra a integração de detecção monocular com redes profundas para geração musical, empregando OpenPose para classificar estilos e gerar sequências MIDI (LIMA, 2021). Embora esse sistema demonstre a robustez das tecnologias para análise *off-line*, seu foco recai sobre tarefas com menor exigência de latência. Em contraste, o Aerochord propôs e implementou um mapeamento direto e instantâneo, exigindo que cada nuance de movimento do *performer* fosse refletida imediatamente no comportamento sonoro. Esse contraste reforçou a necessidade de otimizações de *software*, culminando na escolha de C++/JUICE e *pipelines* de processamento assíncronos, visando atender a requisitos de latência determinística.

No Aerochord, a implementação em C++ foi uma decisão de engenharia estratégica que visou minimizar a latência ao longo de todo o *pipeline*, da captura à saída sonora. No caminho de detecção, esse ganho apoia-se no acesso direto à câmera via V4L2, na inferência nativa do MediaPipe e em filas livres de bloqueio (*lock-free*) entre os módulos; já o *framework* JUICE atua na etapa final de transporte e formatação das mensagens MIDI. Essa escolha distinguiu o projeto de muitos protótipos acadêmicos desenvolvidos em linguagens interpretadas, garantindo atualização imediata dos parâmetros sonoros assim que o gesto é reconhecido. Desse modo, o Aerochord aliou a acessibilidade intrínseca do paradigma monocular à robustez de engenharia necessária para atender seus usuários em cenários de *performance* em tempo real.

Esses três paradigmas — desde a fidelidade extrema das *data gloves*, passando pelos gestos livres com *hardware* dedicado, até a detecção monocular acessível — delineiam um espectro de *trade-offs* entre precisão, expressividade e inclusão. A próxima seção examina como cada paradigma influencia as estratégias de mapeamento gesto-som.

4.2 MAPEAMENTO GESTO-SOM

O mapeamento entre gestos e parâmetros sonoros é frequentemente descrito como o “coração” de qualquer instrumento musical digital, pois funciona como ponte conceitual que traduz a intenção expressiva do *performer* em variações sonoras audíveis. Mesmo que uma interface gestual ofereça captura de alta fidelidade ou acessibilidade ampliada, sem um mapeamento bem concebido o resultado tende a ser artificial ou pouco expressivo.

4.2.1 Estratégias de Mapeamento

Uma distinção fundamental em estratégias de mapeamento é entre mecanismos discretos e contínuos. Nos mapeamentos discretos, gestos específicos disparam eventos pontuais, como o envio de um comando de ativação de nota (*Note On*) ou a execução de um *sample* previamente definido, resultando em respostas binárias. Por outro lado, mapeamentos contínuos

permitem que variações graduais em um gesto modulem parâmetros sonoros de forma fluida, como a distância entre dois dedos controlando volume.

Em trabalhos clássicos, discute-se que a combinação de sensores deve contemplar ambos os tipos de mapeamento, de modo a oferecer tanto a capacidade de acionar eventos precisos quanto a modulação contínua (WANDERLEY *et al.*, 2000; CADOZ; WANDERLEY, 2000). A literatura também explora estratégias híbridas, em que eventos discretos iniciam processos sonoros, como o começo de uma base harmônica contínua (*pad*), e gestos contínuos modulam os parâmetros internos desse processo (WANDERLEY *et al.*, 2000).

No contexto do Aerochord, essa distinção foi adotada de forma integrada: a arquitetura desenvolvida oferece mapeamentos discretos (através da Máquina de Estados) para o disparo de notas, e mapeamentos contínuos para modulação fluida de parâmetros como a dobra de afinação (*Pitch Bend*), criando performances que transitam suavemente entre precisão rítmica e exploração tímbrica.

4.2.2 Filosofias de Design de Mapeamento: *Movement-First* vs. *Sound-First*

Duas filosofias de *design* de mapeamento são frequentemente contrapostas na literatura: a abordagem orientada ao movimento (*movement-first*) e a abordagem orientada ao som (*sound-first*). Na filosofia *movement-first*, predomina a exploração do movimento como ponto de partida: o desenvolvedor parte da liberdade cinemática do gesto, buscando capturar variações naturais do corpo para então associá-las ao domínio sonoro.

Por outro lado, a filosofia *sound-first* parte da definição prévia de resultados sonoros desejados, buscando mapeamentos que garantam precisão, repetibilidade e controle previsível. O *design* estrutura o reconhecimento gestual de modo a satisfazer requisitos acústicos rigorosos. Essa abordagem prevalece em contextos em que a coerência sonora é prioritária (WANDERLEY *et al.*, 2000).

Em muitos projetos acadêmicos, observa-se que cada filosofia tende a ser aplicada isoladamente (AIGNER *et al.*, 2012; VOGIATZIDAKIS; KOUTSABASIS, 2018). No Aerochord, o mapeamento buscou equilibrar essas filosofias ao implementar a Máquina de Estados (FSM) orientada ao controle preciso (*sound-first*) para o gatilho das notas, e fluxos de dados contínuos para a modulação natural do gesto (*movement-first*), não comprometendo a coerência entre intenção e resultado sonoro.

4.3 AVALIAÇÃO, USUÁRIOS E CONTEXTOS DE USO

Avaliar interfaces requer ir além de especificações puramente técnicas, pois a qualidade depende de fatores perceptivos que influenciam diretamente a aceitação. Esta seção apresenta

métricas objetivas e subjetivas para mensurar desempenho e usabilidade.

4.3.1 Métricas e Metodologias de Avaliação

A avaliação de um Instrumento Musical Digital (IMD) difere da avaliação de uma interface convencional. Conforme argumenta O'Modhain (2011), o critério de sucesso não se limita à completude ou rapidez de tarefas, mas estende-se ao potencial do instrumento para gerar uma experiência estética de qualidade.

No cerne da avaliação residem critérios que vão além da usabilidade. Em primeiro lugar, a expressividade. Em segundo lugar, a tocabilidade (*playability*), que engloba o quão satisfatório e fluido é o ato de tocar o instrumento. Por fim, a curva de aprendizado (*learnability*), que se diferencia no baixo custo de entrada (*low entry fee* - a facilidade para iniciantes) e no teto para virtuosidade (*high ceiling* - a profundidade para usuários avançados) (WANDERLEY; ORIO, 2002).

Para operacionalizar esses critérios, a avaliação adota abordagens mistas. Entre as métricas objetivas, destaca-se a medição da latência *end-to-end*, entendida como o tempo decorrido entre o gesto físico e a resposta sonora (LUGARESI *et al.*, 2019; HASSAN; DESSOUKY; EL-LATIF, 2019). O protocolo de avaliação do Aerochord ancorou-se nesta perspectiva objetiva de *performance*. A estabilidade do *pipeline* assíncrono e as avaliações empíricas de latência foram conduzidas rigorosamente para assegurar que a ferramenta atinja a tocabilidade necessária.

4.3.2 Perfis de Usuário e Cenários de Aplicação

As interfaces musicais atendem perfis diversificados. Soluções que dependem de *hardware* dedicado tendem a restringir-se a nichos com maior orçamento. Em contraste, abordagens que utilizam *webcams* demonstram grande potencial para aplicações de baixo custo e ambientes educacionais (AMERICO, 2020; VAMVAKOUSIS; RAMIREZ, 2016).

O Aerochord posicionou-se como ferramenta focada na democratização: projetado para atacar três barreiras interligadas. A barreira econômica foi enfrentada pela exigência de equipamento mínimo. A barreira da proficiência técnica foi contornada por meio de uma interface intuitiva. E a barreira da limitação motora foi abordada ao permitir controle através de gestos de alcance variado, reduzindo a necessidade exclusiva de motricidade fina extrema.

4.4 SÍNTESE CRÍTICA E POSICIONAMENTO DO AEROCHORD

A partir da análise detalhada, torna-se possível realizar uma síntese crítica do campo de interfaces musicais gestuais.

4.4.1 Mapeamento Comparativo das Abordagens

A Tabela 1 apresenta de forma estruturada os principais parâmetros considerados em cada abordagem.

Tabela 1 – Comparativo detalhado de paradigmas e sistemas em interfaces musicais gestuais

Sistema/Referência	Paradigma de Interação	Tecnologia de Captura	Hardware Requerido	Nível de Expressividade (Potencial)	Acessibilidade (Custo/Facilidade)	
Data Gloves (DIPIETRO; SABATINI; DARIO, 2008)	Contato de Alta Fidelidade	Físico	Sensores de flexão e IMUs nas luvas	Luvas especializadas; estação de calibração	Muito alto: captura de detalhes de articulações, mapeamentos multiparamétricos	Baixa: custo elevado, calibração complexa, restrição ergonômica
Kinect e similares (SHOTTON <i>et al.</i> , 2013)	Gestos Livres em Volume 3D	Câmera com sensor de profundidade para rastreamento esquelético	Câmera com sensor de profundidade dedicada	Moderado a alto para gestos amplos, limitado em detalhes de mãos	Moderada: custo de sensor, requisitos de ambiente controlado	
Leap Motion (ZHANG <i>et al.</i> , 2019; RAUTARAY; AGRAWAL, 2012)	Gestos Livres em Volume 3D Focado em Mãos	Sensor infravermelho de curto alcance	Dispositivo Leap Motion ou similar	Alto para gestos manuais finos	Moderada: <i>hardware</i> adicional e área de interação limitada	
Visão Monocular (MediaPipe) (LUGARESI <i>et al.</i> , 2019)	Detecção de Pose com Câmera RGB	<i>Framework</i> de visão (MediaPipe ou equivalente)	Webcam comum (integrada ou USB)	Moderado a bom: com filtragem e mapeamentos híbridos	Elevada: baixo custo, sem equipamento extra, mas sensível a condições ambientais	
EyeHarp (VAMVAKOUSIS; RAMIREZ, 2016)	Rastreamento de Olhar e Seleção por Dwell-Time	Rastreador ocular (<i>eye tracker</i>) ou <i>webcam</i> adaptada	<i>Eye tracker</i> dedicado ou configuração de <i>webcam</i>	Moderado para seleção e sequenciamento, limitado em dinâmica rápida	Alta para mobilidade reduzida, porém não ideal para <i>performance</i> ágil	
Sensores Inerciais (IMUs) (DIPIETRO; SABATINI; DARIO, 2008)	Rastreamento de Movimento Amplo	IMUs em dispositivos vestíveis (<i>wearables</i>) como relógios inteligentes (<i>smartwatches</i>)	Dispositivo com IMU integrado	Moderado para modulação ampla, baixo para detalhes manuais	Elevada: <i>hardware</i> leve e barato, mas insuficiente isoladamente para controle preciso de notas	

A análise revela tendências claras: existe uma correlação inversa entre o uso de *hardware* dedicado e o nível de acessibilidade.

4.4.2 A Lacuna Identificada e a Proposta de Valor do Aerochord

A síntese apresenta um *trade-off* persistente: de um lado, sistemas de alta precisão demandam investimentos significativos. De outro, abordagens baseadas em *webcam* não são concebidas com foco na minimização extrema de latência nem na exploração aprofundada de protocolos como o MIDI 2.0.

O Aerochord preencheu precisamente essa lacuna ao unir três pilares fundamentais. O primeiro pilar trata da máxima acessibilidade: exigindo apenas *webcam*. O segundo pilar aborda a baixa latência, por meio de uma arquitetura implementada em C++/JUICE. O terceiro

pilar corresponde ao compromisso com alta expressividade, integrando nativamente o protocolo MIDI 2.0.

4.5 TENDÊNCIAS E PERSPECTIVAS FUTURAS

Após a análise do panorama atual, torna-se evidente que vetores apontam para direções promissoras, como integração de Inteligência Artificial em mapeamentos e fusão com Realidade Aumentada e Virtual (AR/VR).

Espera-se que modelos de aprendizado de máquina possam capturar o vocabulário gestual de um músico, ajustando automaticamente curvas de resposta (CARAMIAUX *et al.*, 2015). A arquitetura do Aerochord, concebida de forma modular e desacoplada, revelou-se ideal para essa evolução.

Outra frente de inovação envolve a integração com ambientes de realidade mista. Nesse contexto, o Aerochord funciona como um controlador eficiente que pode ter suas saídas de dados direcionadas a *engines* de áudio ou jogos imersivos.

Em síntese, o Aerochord, ao preencher hoje a lacuna entre acessibilidade, baixa latência e expressividade, consolidou-se como uma plataforma cujas camadas podem ser estendidas para incorporar inovações, projetando-se como alicerce para pesquisas futuras.

5 DESENVOLVIMENTO DA PROPOSTA DE SOLUÇÃO

O desenvolvimento do Aerochord foi pautado pelo desafio central de conciliar duas cargas computacionais de naturezas distintas: o processamento intensivo e variável de visão computacional (inferência de redes neurais) e a exigência de latência mínima e determinística para a geração de comandos musicais. Para atingir esse objetivo, a arquitetura do sistema foi projetada sob o paradigma de esteira de processamento assíncrona (*pipeline*), desenvolvida em C++17 e integrando o *framework* JUCE para manipulação de eventos MIDI e a biblioteca MediaPipe para detecção de pose.

Este capítulo detalha a modelagem de requisitos, a arquitetura da solução e o fluxo de dados — desde a captura do movimento até a síntese do comando sonoro —, evidenciando as estratégias de engenharia de *software* adotadas para garantir estabilidade e precisão em tempo real.

5.1 REQUISITOS DO SISTEMA

Para orientar o desenvolvimento e avaliar o sucesso da solução, foram definidos os requisitos funcionais e não funcionais do sistema.

Os **Requisitos Funcionais** englobam as capacidades operacionais do Aerochord:

- Capturar fluxo de vídeo em tempo real a partir de uma *webcam* comum.
- Detectar e rastrear coordenadas tridimensionais de até duas mãos simultaneamente.
- Diferenciar papéis de controle e articulação com base na ordem de entrada das mãos em quadro.
- Traduzir movimentos espaciais em eventos musicais (ativação de notas, volume, seleção de oitava, *pitch bend*, *vibrato* e filtro).
- Gerar e transmitir comandos via protocolo MIDI 2.0 utilizando pacotes UMP nativamente.
- Realizar o *fallback* automático para o padrão MIDI 1.0 caso o dispositivo receptor não suporte o protocolo atualizado.

Os **Requisitos Não Funcionais** ditam as restrições e atributos de qualidade do sistema:

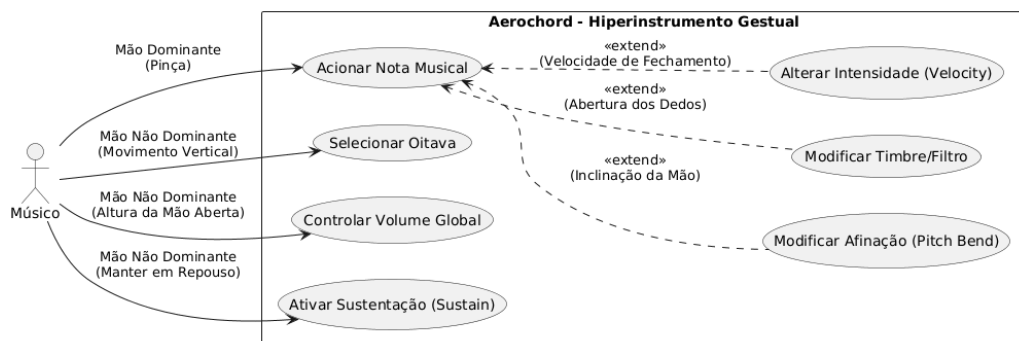
- **Desempenho (Latência):** O sistema deve garantir uma latência *end-to-end* estritamente inferior a 30 ms para assegurar a tocabilidade e evitar atrasos perceptíveis pelo músico.

- **Eficiência Computacional:** O consumo de CPU e RAM deve ser minimizado para permitir a execução paralela de Estações de Trabalho de Áudio Digital (DAWs) na mesma máquina.
- **Acessibilidade Tecnológica:** O sistema não deve exigir vestimentas (como luvas) ou sensores especializados além de uma câmera RGB convencional.

5.2 MODELAGEM DE CASOS DE USO

Para formalizar as interações entre o usuário e o hiperinstrumento, elaborou-se o Diagrama de Casos de Uso (Figura 8). Na modelagem, o único ator externo primário é o **Músico** (ou intérprete). Durante a performance musical, a interação ocorre por meio de gestos no espaço tridimensional capturados pela câmera; complementarmente, um painel gráfico opcional (disponível quando a visualização está ativa) oferece controles de configuração não-expressivos — alternância do modo *legato* e seleção de instrumento (*Program Change*) —, que não fazem parte do fluxo gestual de execução.

Figura 8 – Diagrama de Casos de Uso do Aerochord



Fonte: O Autor (2026).

O sistema prevê os seguintes casos de uso principais, que são acionados simultaneamente dependendo do papel de cada mão (controle ou articulação, atribuído por ordem de entrada em quadro — ver Seção 5.4):

- **Acionar Nota:** O caso de uso central, onde o músico realiza o fechamento dos dedos para engatilhar um evento sonoro.
- **Modular Afinação (*Pitch Bend*):** Ação contínua e simultânea à nota ativada, alterando a altura da nota através da inclinação da mão.
- **Controlar Timbre/Filtro:** Modulação contínua do som utilizando a extensão do dedo médio (distância da ponta ao pulso).
- **Controlar Volume e Oitava:** Ações de macrocontrole, operadas pela mão de controle, que configuram a região tonal e a intensidade global do som.

- **Acionar Sustentação (*Sustain*):** Manutenção do som após a liberação do gesto original.

5.3 ARQUITETURA DO SISTEMA

A arquitetura do Aerochord é composta por cinco módulos principais de processamento, além de um módulo opcional de visualização. Para evitar gargalos e garantir que a lentidão momentânea na inferência de vídeo não bloqueie a emissão de comandos MIDI, cada módulo opera em sua própria *thread* dedicada.

Figura 9 – Diagrama de Arquitetura do Aerochord: Módulos de processamento e comunicação assíncrona



Fonte: O Autor (2026).

A comunicação inter-processual é realizada exclusivamente por meio de filas de produtor único e consumidor único (*Single-Producer / Single-Consumer* - SPSC) do tipo *lock-free*. Esta estrutura, implementada por meio de *buffers* circulares estáticos e operações atômicas com barreiras de memória rigorosas, elimina a necessidade de *mutexes*. Na prática, isso impede o *thread blocking*, garantindo que o fluxo de dados musicais no *hot path* ocorra sem interrupções por sincronização, requisito fundamental em sistemas de áudio de baixa latência.

O fluxo contínuo de dados atravessa os seguintes estágios:

1. **Captura:** Obtenção de quadros da câmera.
2. **Detecção de Pose:** Extração de coordenadas espaciais das mãos.
3. **Análise de Gestos:** Tradução de coordenadas em eventos com significado semântico através de uma Máquina de Estados Finitos (*Finite State Machine* - FSM).
4. **Mapeamento:** Conversão de intenções gestuais em comandos musicais brutos.
5. **Geração MIDI:** Empacotamento e transmissão de protocolos musicais (MIDI 1.0/2.0).

5.4 CAPTURA E DETECÇÃO DE POSE

A primeira etapa do *pipeline* é gerenciada pelo módulo de captura (*CaptureModule*), responsável por interagir diretamente com o *hardware* de vídeo. Em ambientes Linux, o módulo utiliza a interface V4L2 (*Video4Linux2*) com mapeamento direto de memória (*memory*

mapping) para acessar os *buffers* da câmera, evitando cópias desnecessárias na memória do sistema.

Na sequência, o módulo de detecção (*PoseDetectionModule*) consome esses quadros e aciona o rastreador de pontos-chave das mãos (*Hand Landmarker*) do MediaPipe. Este módulo extrai 21 *landmarks* tridimensionais normalizados para até duas mãos simultaneamente. Para mitigar o *jitter* (pequenas oscilações naturais e ruídos na inferência da rede neural) que poderia resultar em ruído sonoro, o sistema aplica um filtro de Média Móvel Exponencial (EMA) sobre as coordenadas espaciais. Adicionalmente, um algoritmo de correspondência espacial por proximidade do pulso garante que as detecções não troquem de papel inadvertidamente entre os quadros, preservando a coerência da atribuição de cada mão ao longo do tempo.

A atribuição inicial dos papéis de cada mão (controle vs. articulação) é feita por **ordem de entrada em quadro**, e não pela classificação de lateralidade (esquerda/direita) do MediaPipe: a primeira mão detectada após um período sem nenhuma mão em quadro assume o papel de controle (volume/oitava); a segunda mão a entrar assume o papel de articulação (pinça/notas). Essa escolha de projeto decorre da baixa confiabilidade da classificação de lateralidade do MediaPipe quando apenas uma mão está visível, sem o contexto da outra mão para comparação — o que tornaria o rótulo bruto do modelo uma base instável para a divisão de papéis. Como efeito colateral desejável, esse critério torna o sistema utilizável tanto por destros quanto por canhotos sem qualquer configuração: o usuário simplesmente posiciona primeiro a mão que deseja usar para controle.

5.5 MÁQUINA DE ESTADOS E ANÁLISE GESTUAL

O módulo de análise (*GestureAnalysisModule*) atua como o motor lógico do sistema. Ele recebe as coordenadas filtradas e extrai métricas geométricas, como distâncias euclidianas e ângulos relativos, alimentando a Máquina de Estados Finitos (FSM) que identifica a intenção musical do usuário.

Visando acessibilidade e ergonomia irrestrita (suportando usuários destros e canhotos sem qualquer configuração prévia), o sistema divide as responsabilidades entre dois papéis funcionais, atribuídos por ordem de entrada em quadro conforme descrito na Seção 5.4:

- **Mão de Articulação e Modulação (segunda a entrar em quadro):** Controla o gatilho das notas através de um gesto de "pinça" (aproximação entre polegar e indicador). A FSM transita pelos estados de inatividade (*IDLE*), início de pinça (*PINCH_START*) e sustentação de pinça (*PINCH_HOLD*). Durante a sustentação, o módulo extrai variações contínuas, como o ângulo de inclinação da mão para modulação de afinação (*Pitch Bend*) e a taxa de oscilação lateral para controle de *Vibrato*.
- **Mão de Macrocontrole (primeira a entrar em quadro):** Opera majoritariamente de

forma contínua. A altura vertical da mão aberta traduz-se em controle global de volume. Um gesto de pinça acompanhado de movimento vertical aciona a transição de oitavas, enquanto manter a mão em repouso por um período pré-determinado alterna o estado do pedal de sustentação (*Sustain*).

Como o papel de cada mão é definido pela ordem de entrada e não pela lateralidade anatômica, o usuário escolhe livremente qual mão (esquerda ou direita) deseja usar para articulação, bastando posicioná-la em quadro depois da mão de controle.

Para garantir estabilidade, o módulo implementa lógicas de estabilização de sinal (*debounce*, exigindo estabilidade por quadros consecutivos) e *histerese* (diferença de limiares) nas transições de estado, prevenindo engatilhamentos duplos e acidentais das notas.

5.6 ESTRATÉGIAS DE MAPEAMENTO GESTO-SOM

O módulo de mapeamento (*MappingModule*) consome os eventos gestuais semânticos e os traduz em comandos MIDI, atuando como a interface entre o movimento físico e a teoria musical.

A Tabela 2 descreve o conjunto de mapeamentos implementados, detalhando cada gesto básico, a ação musical correspondente e a justificativa para a escolha do gesto em termos ergonômicos e perceptivos, respeitando a lógica de lateralidade.

Tabela 2 – Mapeamento Gesto–Som Implementado no Aerochord

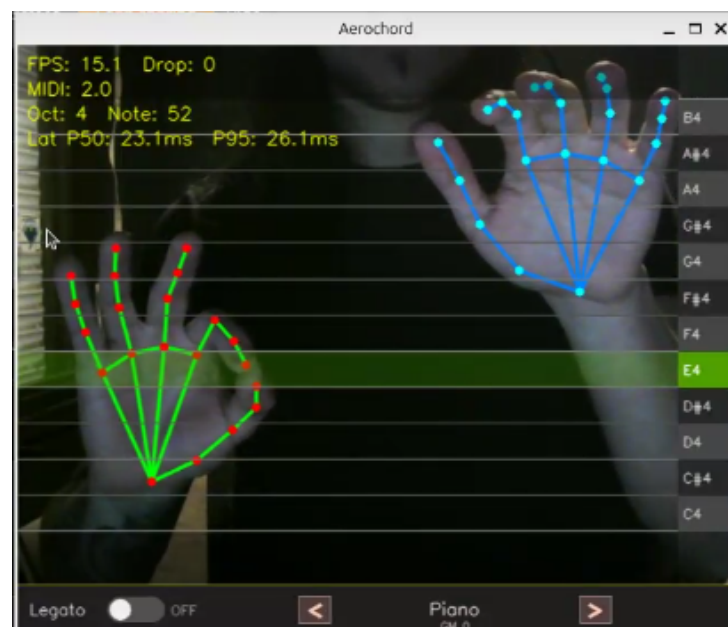
Função Musical	Mão (ordem de entrada)	Gesto Utilizado	Justificativa
Seleção de Oitava	1ª (controle)	Pinça + movimento vertical.	Gesto modal claro que separa o controle de oitava de outras funções, evitando acionamentos acidentais.
Volume Global	1ª (controle)	Mão aberta + movimento vertical.	Mapeamento intuitivo (cima/baixo = mais/menos) que utiliza um gesto natural de “repouso”.
Disparo de Nota	2ª (articulação)	Pinça (polegar + indicador) em zonas verticais discretas.	Gesto ergonômico e preciso para iniciar o som. A altura da mão define a nota de forma intuitiva.
Parada de Nota	2ª (articulação)	Soltar o gesto de pinça.	Ação de abrir a mão é a contraparte mecânica e imediata ao gesto de disparo.
Timbre/Filtro	2ª (articulação)	Distância entre a ponta do dedo médio e o pulso (base da mão).	Gesto robusto para rastreamento que oferece grande amplitude expressiva sem influenciar a afinação.
<i>Pitch Bend</i>	2ª (articulação)	Inclinação (rotação) da mão.	Movimento angular é ortogonal ao movimento de seleção de nota, minimizando conflitos de leitura.
<i>Vibrato</i>	2ª (articulação)	Oscilação leve lateral.	Mimetiza mecanicamente o <i>vibrato</i> de instrumentos de corda acústicos.
Intensidade (<i>Velocity</i>)	2ª (articulação)	Velocidade temporal de fechamento da pinça.	Imita a física de instrumentos de percussão e teclas (tocar mais rápido gera um ataque mais intenso).
Sustentação (<i>Sustain</i>)	1ª (controle)	Mão aberta mantida parada por um intervalo curto (alterna liga/desliga).	Gesto de “repouso” que evita conflito com os controles contínuos de volume e oitava, análogo ao pedal de um piano.

Observa-se que a mão de controle foi minimamente utilizada, o que abre margem para expansões futuras de controle (como acionamento de acordes inteiros ou mudança de progra-

mas). O mapeamento discrimina claramente ações discretas (ativação de nota) e modulações contínuas.

O sistema divide a área de atuação da câmera em 12 zonas verticais (Figura 10), correspondentes aos semitons de uma oitava completa. O mapeamento incorpora zonas mortas (*dead zones* — áreas de fronteira sem resposta) e preenchimento (*padding*) nas extremidades superior e inferior do quadro visual. Isso fixa as notas mais graves e agudas, garantindo que o *performer* não precise esticar o braço até os limites físicos do alcance da câmera para tocar as notas extremas da escala.

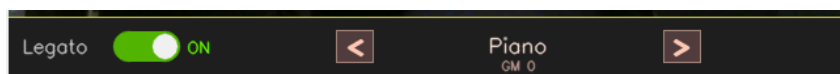
Figura 10 – Interface do Aerochord em execução: esqueleto das mãos detectadas, guia das 12 zonas verticais (semitons) na margem direita com a nota ativa destacada, zonas mortas sombreadas nas extremidades e HUD de telemetria no canto superior esquerdo



Fonte: O Autor (2026).

Um recurso de destaque na implementação é o modo de ligação (*legato*). Quando ativado, o mapeador permite a transição de notas de forma fluida (deslizando a mão de articulação ainda fechada entre as zonas verticais). A engenharia do módulo garante que o comando da nova nota (NOTE_ON) seja emitido milissegundos antes do encerramento da nota anterior (NOTE_OFF), criando a sobreposição de áudio necessária para que os sintetizadores engatilhem transições fluídas, mimetizando instrumentos de sopro. O modo *legato* e a seleção de instrumento (*Program Change*) são acionados pelo painel de controle exibido abaixo do quadro de vídeo (Figura 11).

Figura 11 – Painel de controle não-expressivo exibido abaixo do quadro de vídeo: alternância do modo *legato* (ativado) e seletor de instrumento General MIDI



Fonte: O Autor (2026).

5.7 GERAÇÃO DE PROTOCOLO E SUPORTE NATIVO A MIDI 2.0

A etapa final reside no módulo de comunicação, responsável por formatar e entregar os comandos musicais ao sistema operacional. O Aerochord inova ao implementar suporte ao padrão MIDI 2.0 através da formatação de Pacotes MIDI Universais (*Universal MIDI Packet - UMP*).

Durante a inicialização, o sistema seleciona o transporte de saída em uma ordem de preferência. O caminho primário para o MIDI 2.0, em ambientes Linux, é o suporte nativo a UMP do *ALSA Sequencer* (disponível em *kernel* recente em conjunto com a *alsa-lib* compatível): o módulo abre uma porta virtual de saída em modo MIDI 2.0, à qual sintetizadores e Estações de Trabalho de Áudio Digital podem se conectar. Estando esse transporte disponível, o sistema constrói pacotes UMP de 64 bits e os despacha diretamente por essa porta. Isso garante resoluções de 32 bits para mensagens de controle contínuo, além de permitir o *Per-Note Pitch Bend* (dobro de afinação individualizada por nota ativa), elevando drasticamente o teto de expressividade instrumental da aplicação. A seleção desse transporte é registrada no *log* de inicialização do sistema (Figura 12).

Figura 12 – *Log* de inicialização confirmando a seleção do transporte primário: criação da porta virtual “Aerochord UMP” e ativação do modo MIDI 2.0 via UMP nativo do ALSA Sequencer

```
[22:36:50.623046] [INFO] [MidiGenerationModule] Dispositivo aberto: Midi Through Port-0
[22:36:50.623098] [INFO] [MidiGenerationModule] ALSA UMP ativo – porta "Aerochord UMP" (client:130 port:0). Conecte um sintetizador via aconnect.
[22:36:50.623105] [INFO] [MidiGenerationModule] MIDI 2.0 ativo via ALSA Sequencer UMP.
[22:36:50.623216] [INFO] [MidiGenerationModule] Auto-conexão: "Aerochord UMP" -> "FLUID Synth (47236)" (128:0).
```

Fonte: O Autor (2026).

Como mecanismo complementar de interoperabilidade, o sistema também implementa a negociação por Consulta de Capacidades (*Capability Inquiry - MIDI-CI*): quando o transporte UMP nativo não está disponível, é realizado um *handshake* que envia mensagens de descoberta (*Discovery*) para verificar se o dispositivo receptor anuncia suporte à nova especificação. Sobre essa negociação foi prototipada ainda a Troca de Propriedades (*Property Exchange*), prevista para a calibragem dinâmica de parâmetros.

Caso nenhum transporte MIDI 2.0 esteja disponível — ou o equipamento/*software* receptor não suporte o novo padrão —, o sistema aplica um mecanismo de *fallback* transparente, truncando matematicamente os dados para os 7 bits originais e roteando a comunicação via arquitetura legado (MIDI 1.0), assegurando total compatibilidade com tecnologias anteriores e garantindo a universalidade da solução.

A porta MIDI 2.0 criada pelo Aerochord torna-se imediatamente visível e conectável a sintetizadores e Estações de Trabalho de Áudio Digital do sistema operacional, como evidenciado na Figura 13.

Figura 13 – Porta virtual “Aerochord UMP” disponível para conexão no subsistema de áudio do sistema operacional (saída do utilitário `aconnect -l`)

```
cubano@cubano-ubuntu:~/Aerochord$ aconnect -l
client 0: 'System' [type=kernel]
  0 'Timer'
    Connecting To: 144:0
  1 'Announce'
    Connecting To: 144:0, 129:0
client 14: 'Midi Through' [type=kernel]
  0 'Midi Through Port-0'
    Connected From: 129:1
client 128: 'FLUID Synth (47236)' [type=user,pid=47236]
  0 'Synth input port (47236:0)'
    Connected From: 130:0
client 129: 'JUCE' [type=user,pid=47927]
  0 'announcements'
    Connected From: 0:1
  1 'Midi Through Port-0'
    Connecting To: 14:0
client 130: 'Aerochord' [type=user,pid=47927]
  0 'Aerochord UMP'
    Connecting To: 128:0
client 144: 'PipeWire-System' [type=user,pid=2269]
  0 'input'
    Connected From: 0:1, 0:0
client 145: 'PipeWire-RT-Event' [type=user,pid=2269]
  0 'input'
```

Fonte: O Autor (2026).

6 TESTES E ANÁLISE DE RESULTADOS

Após o desenvolvimento e a implementação da arquitetura do Aerochord, tornou-se imperativo submeter o sistema a um processo de validação empírica. Este capítulo detalha os testes realizados para aferir se a ferramenta atingiu as metas propostas nos requisitos funcionais e não funcionais.

A avaliação foi dividida em três frentes complementares. Na Seção 6.1, analisam-se os dados objetivos de desempenho do sistema operando em tempo real, com foco na medição estatística da latência *end-to-end* e no consumo de recursos da máquina. Na Seção 6.2, discute-se a estabilidade do rastreamento gestual e a qualidade percebida das articulações e mapeamentos. Por fim, na Seção 6.3, apresenta-se um teste piloto de usabilidade conduzido com um voluntário, visando avaliar a facilidade de uso e a curva de aprendizagem do hiperinstrumento.

6.1 AVALIAÇÃO DE DESEMPENHO TÉCNICO E LATÊNCIA

O requisito técnico mais crítico para qualquer Instrumento Musical Digital é a manutenção de uma latência imperceptível para o intérprete, garantindo a sensação de causalidade imediata entre o gesto físico e o evento sonoro. Conforme os estudos levantados sobre interfaces baseadas em visão computacional, a latência *end-to-end* nesse paradigma varia tipicamente entre 30 ms e 60 ms (LUGARESI *et al.*, 2019; HASSAN; DESSOUKY; EL-LATIF, 2019). Portanto, estabelecer e alcançar uma meta abaixo da barreira dos 30 milissegundos (ms) torna-se o marco ideal para superar as limitações técnicas comuns da área e preservar a tocabilidade em aplicações musicais interativas.

Para medir essa métrica, o Aerochord foi executado em uma estação de trabalho de alto desempenho, equipada com processador AMD Ryzen 7 7800X3D, 32 GB de memória RAM DDR5, placa de vídeo NVIDIA GeForce RTX 4070 Ti e sistema operacional Ubuntu 24.04.4 LTS. A captura visual foi realizada por meio de uma *webcam* Logitech C505, que possui uma resolução de 720p (1280x720 pixels) e taxa de captura de 30 quadros por segundo.

O sistema operou utilizando o modo analítico da aplicação (`-eval`). Neste modo, o *software* registra *timestamps* exatos no momento em que um quadro de vídeo é capturado (entregue pelo subsistema de vídeo) e, em seguida, registra o momento exato em que a mensagem MIDI 2.0 correspondente é enviada para o sequenciador do sistema operacional. Cabe ressaltar que essa métrica isola a latência do *pipeline* de *software* — da captura do quadro ao envio da mensagem MIDI —, não incluindo a latência intrínseca da câmera (exposição do sensor e transferência de dados via barramento USB) nem a renderização de áudio do sintetizador receptor, ambas independentes da implementação.

O teste de carga contínua consistiu na execução da aplicação por um período prolongado de interação ativa, gerando um conjunto de dados (*dataset*) com 7.821 amostras de eventos musicais consecutivos.

6.1.1 Resultados de Latência

A análise estatística do arquivo de registro gerado pelo sistema demonstrou resultados altamente satisfatórios, superando a meta estabelecida. Os dados recolhidos foram:

- **Amostras totais:** 7.821 eventos.
- **Latência Média:** 22,62 ms.
- **Percentil 50 (P50 - Mediana):** 22,29 ms.
- **Percentil 95 (P95):** 24,73 ms.
- **Amostras abaixo de 30 ms:** 99,2%.

O indicador P95 de 24,73 ms atesta que 95% de todos os eventos musicais gerados pelo sistema — desde a captura do quadro pelo *software* até o envio da mensagem MIDI — ocorreram em menos de 24,8 milissegundos. Este valor, associado a uma taxa de sucesso de 99,2% das amostras abaixo da tolerância máxima de 30 ms, prova a eficácia da arquitetura concorrente baseada em filas *lock-free* implementada em C++.

6.1.2 Consumo de Recursos do Sistema

O desempenho de baixa latência seria inviável se o consumo de recursos computacionais saturasse o sistema operacional, resultando no descarte de quadros (*frame drops*). Durante os testes em tempo real, a aplicação consumiu em média **25,27% de CPU** (medida em relação à capacidade total da máquina) e utilizou **145,0 MB de Memória RAM**, considerando o *pipeline* central de processamento (captura, detecção, análise, mapeamento e geração MIDI), sem a janela de visualização opcional.

Estes números demonstram que a integração da biblioteca MediaPipe nativa em C++ oferece um impacto computacional significativamente leve. O sistema preserva recursos suficientes para que o mesmo computador possa, simultaneamente, executar uma Estação de Trabalho de Áudio Digital (DAW) e processar sintetizadores virtuais pesados, cumprindo na íntegra a premissa de viabilidade de execução em *hardware* comercial de uso comum.

6.1.3 Distribuição e Expressividade dos Eventos

A análise do *dataset* de 7.821 amostras também revelou a distribuição por tipo de mensagem MIDI gerada durante a execução:

- **Eventos de Nota (*Note On / Note Off*):** 822 amostras (10,5%).
- **Controladores Contínuos (*Control Change - CC*):** 4.805 amostras (61,4%).
- **Dobra de Afinação por Nota (*Per-Note Pitch Bend*):** 2.194 amostras (28,1%).

Este perfil de distribuição, onde aproximadamente 89,5% das mensagens consistem em dados de modulação contínua (CC e Afinação), evidencia matematicamente a filosofia de *design* focada no movimento (*movement-first*). O Aerochord não funciona como um mero disparador de notas estáticas; a maioria dos dados processados reflete a expressividade contínua do corpo do intérprete moldando o som no decorrer do tempo.

6.2 ESTABILIDADE E AVALIAÇÃO QUALITATIVA

Além dos dados quantitativos, uma avaliação qualitativa do uso prolongado permitiu inferir a estabilidade e a ergonomia do instrumento sob condições reais. Observou-se que a detecção do gesto de "pinça" (gatilho de nota da mão de articulação) obteve uma taxa de precisão qualitativa de aproximadamente 90%. As falsas detecções (falsos positivos ou negativos) mostraram-se residuais e estão diretamente ligadas a condições do ambiente externo, nomeadamente ao enquadramento excessivamente descentrado da mão ou a condições de iluminação desfavoráveis, fatores inerentes a abordagens por visão monocular. Em condições de iluminação estável, o sistema mostrou-se altamente responsivo.

O modo de transição fluida de notas (*legato*), suportado pela sobreposição temporal de eventos através do algoritmo de programação, desempenhou a sua função de maneira satisfatória, assegurando transições limpas. Foram registradas falhas esporádicas no acoplamento das notas apenas em ocasiões onde os movimentos do utilizador excederam o limiar de tolerância algorítmico nas zonas de fronteira (zonas mortas).

Quanto à divisão de papéis entre as mãos, a alocação dos macro-controles (seleção de oitavas e volume) para a **mão de controle** (a primeira a entrar em quadro) revelou-se natural. Constatou-se, contudo, que tocar articulações com a mão de articulação enquanto se realizam saltos simultâneos de oitava com a mão de controle exige coordenação e prática prévia por parte do executante. Por outro lado, performances confinadas a uma única oitava, utilizando ambas as mãos predominantemente para timbre e intensidade, demonstraram uma fluidez e curva de aprendizagem bastante acessíveis.

6.3 TESTE PILOTO DE USABILIDADE: ESTUDO DE CASO

A fim de verificar a facilidade de uso e avaliar o baixo custo de entrada (que preconiza a utilização da interface por indivíduos sem formação musical), realizou-se um teste piloto empírico de usabilidade. O voluntário, designado como Participante 1 (P1), tem 25 anos de idade,

atualmente é estudante de Análise e Desenvolvimento de Sistemas e encaixa-se perfeitamente no perfil-alvo de usuários não-especialistas: não possui treino musical formal nem contato rotineiro com instrumentos acústicos tradicionais, embora seja um ouvinte assíduo de música.

6.3.1 Metodologia do Teste e Curva de Aprendizagem

Para garantir o rigor metodológico da avaliação qualitativa, a sessão com o participante foi guiada por um protocolo de tarefas dividido em três etapas principais:

1. **Familiarização Livre (Aclimatização):** Inicialmente, o participante teve um período livre para explorar o sistema, com o objetivo de compreender a área de captura da câmera e o mapeamento básico entre os gestos das mãos e o retorno sonoro.
2. **Tarefas Estruturadas (Controle e Precisão):** Em seguida, foi solicitado que o participante executasse ações intencionais e específicas. Isso incluiu o acionamento isolado e repetido de eventos sonoros (para testar a responsividade), a variação na velocidade dos movimentos (para avaliar a dinâmica e a sensibilidade do sistema) e a manutenção de posturas contínuas no espaço (para testar a manipulação de parâmetros expressivos, como modulação).
3. **Improvisação Musical (Uso Ecológico):** Por fim, o participante foi instruído a utilizar o *Aerochord* de forma contínua em um contexto de improvisação musical, visando avaliar a fadiga física, o conforto ergonômico e a expressividade geral do instrumento em um cenário de uso real.

Foi a partir da execução deste roteiro que as percepções e os *feedbacks* do participante foram coletados e analisados. O teste foi conduzido num ambiente não controlado (doméstico). Foi fornecida a P1 uma breve instrução verbal, com duração inferior a 5 minutos, explicando a separação funcional de papéis entre as mãos (a mão de articulação executa notas e a mão de controle controla a região das oitavas e o volume), atribuídos automaticamente pela ordem de entrada em quadro.

Após a explicação inicial, P1 iniciou a exploração livre do hiperinstrumento. Sem qualquer necessidade de intervenção do investigador ou consulta a diagramas, o participante foi capaz de gerar sequências melódicas intencionais, controlando o volume e transitando entre oitavas logo nos primeiros minutos de utilização. O sucesso imediato de P1 em tocar o *Aerochord*, em contraste direto com a longa curva de aprendizagem requerida por instrumentos de cordas ou sopro convencionais, fornece indícios preliminares de que o sistema possui uma barreira técnica de entrada baixa, ainda que testes mais amplos e robustos sejam necessários para a comprovação estatística dessa hipótese.

6.3.2 Feedback Qualitativo e Limitações Identificadas

Após a sessão exploratória, o participante forneceu suas impressões verbais sobre a experiência (retorno sensorial / *feedback* qualitativo). O sistema foi descrito como altamente reativo e de fácil compreensão mecânica.

Entretanto, durante o relato final, P1 fez a seguinte observação crucial sobre a profundidade funcional da aplicação:

"Hoje vejo que só é possível tocar notas soltas no software. Seria interessante poder tocar acordes."

Este apontamento evidencia não apenas a assimilação rápida do funcionamento da ferramenta por parte do participante, mas também identifica uma limitação funcional arquitetônica da versão atual do Aerochord: o mapeamento de um único rastreador de pinça por mão restringe a execução à monofonia ou a melodias de nota única. O desejo orgânico do utilizador de expandir o escopo sonoro para execuções harmônicas completas (acordes polifônicos) será discutido como o principal foco de trabalho futuro do sistema.

7 CONSIDERAÇÕES FINAIS

Este trabalho propôs o desenvolvimento e a avaliação do Aerochord, um hiperinstrumento musical gestual fundamentado em visão computacional e no protocolo MIDI 2.0. A motivação central da pesquisa originou-se na observação de uma lacuna significativa no estado da arte: a ausência de sistemas que aliassem acessibilidade — dispensando *hardware* de alto custo —, baixa latência e expressividade de nível profissional. Ao longo dos capítulos, a transição da fundamentação teórica para a implementação prática demonstrou que é possível democratizar a execução musical sem abdicar do rigor técnico.

A pesquisa cumpriu os seus objetivos delineados no Capítulo 1. A revisão bibliográfica consolidou o entendimento sobre a interação humano-computador aplicada à música e a evolução dos hiperinstrumentos. A adoção do protocolo MIDI 2.0 provou ser um diferencial crítico: a modulação de alta resolução (controle contínuo de 32 bits e *velocity* de 16 bits) eliminou os saltos audíveis do padrão legado, conferindo resposta sonora refinada já na versão atual. Adicionalmente, o sistema emprega a dobra de afinação por nota individual (*Per-Note Pitch Bend*) — recurso inexistente no MIDI 1.0 — que, embora na versão monofônica atual seja equivalente em efeito audível à dobra por canal, prepara a arquitetura para a expressividade polifônica prevista nos trabalhos futuros. A arquitetura de *software*, construída em linguagem C++ e utilizando o *framework* JUCE em conjunto com a biblioteca MediaPipe, materializou esses conceitos numa solução robusta e funcional.

7.1 SÍNTESE DOS RESULTADOS

A avaliação empírica do Aerochord demonstrou indícios fortes da viabilidade técnica e prática da proposta no ambiente testado. Em termos de desempenho, a arquitetura baseada em *pipelines* assíncronos e filas *lock-free* foi capaz de manter a latência *end-to-end* rigorosamente controlada. O percentil 95 (P95) de 24,73 milissegundos atesta que o sistema opera consistentemente abaixo da barreira perceptível de 30 milissegundos, um marco notável para uma aplicação que realiza inferência de redes neurais em tempo real sobre imagens de vídeo.

Simultaneamente, o consumo de recursos da máquina revelou-se altamente eficiente (consumindo cerca de 25% de CPU e aproximadamente 145 MB de memória RAM num ambiente de teste padrão), sugerindo a sustentabilidade da execução do Aerochord em paralelo com Estações de Trabalho de Áudio Digital (DAWs) num mesmo computador de uso doméstico.

Na perspectiva da usabilidade, o teste piloto empírico demonstrou que o hiperinstrumento possui uma barreira técnica de entrada muito baixa. O modelo de interação centrado na distinção ergonômica entre a mão de articulação (para notas e efeitos) e a mão de controle (para macro-controles como volume e oitavas), com os papéis atribuídos pela ordem de entrada em

quadro, mostrou-se natural. Essa separação de papéis permitiu que um performer sem formação musical formal compreendesse o sistema e gerasse melodias expressivas com menos de cinco minutos de instrução preliminar.

7.2 LIMITAÇÕES DO ESTUDO

Apesar dos resultados positivos, o sistema apresenta limitações inerentes à tecnologia escolhida e às decisões iniciais de *design*. A utilização de visão monocular (captura por uma única lente de *webcam*) torna a aplicação dependente de boas condições de iluminação e de um enquadramento visual claro. Situações de oclusão mecânica das mãos ou movimentos excessivamente rápidos que gerem borrão de movimento (*motion blur*) podem induzir falhas pontuais no rastreamento dos *landmarks*.

Adicionalmente, a lógica de acionamento musical desenvolvida na versão atual da Máquina de Estados Finitos restringe a execução a uma única nota por vez na mão de articulação (monofonia). O mapeamento de um único gesto de pinça global impede, momentaneamente, a formulação de blocos harmônicos simultâneos no mesmo canal.

Outra limitação metodológica importante refere-se ao escopo da testagem. A medição de desempenho foi validada em um único setup de *hardware* de alto desempenho, e a avaliação qualitativa contou com uma amostragem de apenas um participante. Além disso, o cálculo de latência desconsiderou o tempo intrínseco de captura de vídeo pela câmera. Portanto, os dados atuais não fornecem amostragem suficiente para generalizações universais em equipamentos com menos recursos computacionais.

7.3 TRABALHOS FUTUROS

As limitações identificadas e o *feedback* qualitativo obtido durante os testes abrem um leque de oportunidades promissoras para a continuidade deste trabalho. Entre as principais diretrizes para evolução do Aerochord, destacam-se:

- **Implementação de Polifonia e Acordes:** Conforme apontado diretamente pelo participante do teste piloto empírico (que expressou o forte desejo de tocar acordes em vez de notas soltas), o desenvolvimento de uma lógica para reconhecer o fechamento independente de múltiplos dedos permitirá a execução de harmonias completas, elevando o nível de virtuosidade da aplicação.
- **Mapeamentos Inteligentes via *Machine Learning*:** Integrar algoritmos que aprendam o vocabulário gestual individual do intérprete, ajustando automaticamente as curvas de sensibilidade e as zonas mortas (*dead zones*) em tempo real, personalizando ainda mais a experiência ergonômica.

- **Fusão Multimodal:** Combinar o rastreamento da *webcam* com Unidades de Medição Inercial presentes em dispositivos vestíveis comuns, como relógios inteligentes (*smartwatches*), para mitigar as falhas de oclusão visual da lente monocular e aumentar a precisão dos gestos.
- **Exploração em Ambientes Imersivos:** Expandir a saída de dados do sistema para controlar *engines* sonoros em ambientes de Realidade Virtual e Realidade Aumentada (VR/AR), tirando proveito da arquitetura assíncrona de baixa latência já estabelecida.
- **Testagem em Larga Escala e Múltiplos Setups:** Conduzir um processo de avaliação mais amplo e robusto, com uma amostragem maior de participantes e medições de desempenho em equipamentos com diferentes configurações de *hardware* e sistemas operacionais, incluindo o tempo de captura de vídeo na latência total.

Em suma, o Aerochord consolida-se como um passo significativo em direção à democratização da musicalidade mediada por computador. O equilíbrio alcançado entre o baixo custo estrutural, o alto desempenho computacional e a inovação tecnológica conferida pelo protocolo MIDI 2.0 prova que a expressão musical gestual digital não precisa ser um privilégio exclusivo de laboratórios e estúdios avançados, podendo ser acessível, responsiva e profundamente expressiva para qualquer pessoa com um computador e uma câmera.

REFERÊNCIAS

- AIGNER, R. *et al.* Understanding mid-air hand gestures: A study of human preferences in usage of gesture types for hci. **Microsoft Research Technical Report**, MSR-TR-2012-111, 2012.
- AMERICO, G. B. **Hyper-Harp: Um Instrumento Musical Digital Baseado em Gestos com Câmera RGB**. Dissertação (Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Computação)) — Universidade Federal do Rio Grande do Sul, 2020. Orientador: Prof. Dr. Leandro Krug Wives.
- Association of Musical Electronics Industry; MIDI Manufacturers Association. **Complete MIDI 1.0 Detailed Specification**. [S.l.], 1996. Document Version 96.1, Third Edition; referência normativa para MIDI 1.0. Disponível em: <<https://www.midi.org>>.
- AVANZINI, F.; FASCHI, V.; LUDOVICO, L. A. Título extraído do pdf smc2023_midi.pdf. **Proceedings of the Sound and Music Computing Conference**, 2023. Disponível em SMC2023_MIDI.pdf.
- BENWARD, B.; SAKER, M. **Music in Theory and Practice**. 8th. ed. [S.l.]: McGraw-Hill Education, 2009.
- BOVERMANN, T. *et al.* **Musical Instruments in the 21st Century: Identities, Configurations, Practices**. 1. ed. [S.l.]: Springer, 2016. ISBN 9811029504,9789811029509.
- CADOZ, C.; WANDERLEY, M. M. Gesture - music. In: _____. **Trends in Gestural Control of Music**. Paris: IRCAM – Centre Pompidou, 2000. p. 71–94.
- CARAMIAUX, B. *et al.* Form follows sound: Designing interactions from sonic memories. In: **Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems**. New York, NY, USA: Association for Computing Machinery, 2015. (CHI '15), p. 3943–3952. ISBN 9781450331456. Disponível em: <<https://doi.org/10.1145/2702123.2702515>>.
- COPE, D. **The Algorithmic Composer**. Middleton, Wisconsin: A-R Editions, Inc., 2001. ISBN 978-0895795185.
- DIPIETRO, L.; SABATINI, A.; DARIO, P. A survey of glove-based systems and their applications. **Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on**, v. 38, p. 461 – 482, 08 2008.
- DODGE, C.; JERSE, T. A. **Computer Music: Synthesis, Composition, and Performance**. New York: Schirmer Books, 1997. ISBN 978-0028646822.
- DUCKWORTH, W. **A Creative Approach to Music Fundamentals**. 9th. ed. [S.l.]: Cengage Learning, 2007.
- Google. **MediaPipe**. 2025. <<https://ai.google.dev/edge/mediapipe>>. Acessado em: 15 jun. 2025.

HASSAN, S. M.; DESSOUKY, M. I.; EL-LATIF, A. M. A. A real-time gesture-based sound control system using rgb cameras. **International Journal of Advanced Computer Science and Applications (IJACSA)**, v. 10, n. 12, p. 413–420, 2019.

LEESE, R.; MCDONALD, A. **Kinect-based Gestural Control of Audio Effects**. 2012. Disponível em repositórios de pesquisa de interfaces musicais.

LIMA, M. C. W. d. P. **Integrando a Dança e a Música em uma Arquitetura de Deep Learning**. Dissertação (Mestrado) — Universidade de Caxias do Sul, 2021. Trabalho de Conclusão de Curso (Bacharel em Ciência da Computação).

LUGARESI, C. *et al.* Mediapipe: A framework for building perception pipelines. **CoRR**, abs/1906.08172, 2019.

MACHOVER, T. **Hyperinstruments: A Progress Report 1987-1991**. [S.l.], 1992.

MANNING, P. **Electronic and computer music** -. fourth edition. [S.l.]: Oxford University Press, 2013. ISBN 9780199746392,0199746397.

MIDI Manufacturers Association. **The History of MIDI**. 2025. Disponível em: <<https://www.midi.org/the-history-of-midi>>. Acessado em: 04 jun. 2025.

MIDI Manufacturers Association; Association of Musical Electronics Industry. **Common Rules for MIDI-CI Profiles**. [S.l.], 2023. MIDI Association Document: M2-102-U, Version 1.1; Draft Date: 11-May-2023; Published: 15-Jun-2023. Disponível em: <<https://www.midi.org>>.

_____. **Common Rules for MIDI-CI Property Exchange**. [S.l.], 2023. MIDI Association Document: M2-103-UM, Version 1.1; Draft Date: 11-May-2023; Published: 15-Jun-2023. Disponível em: <<https://www.midi.org>>.

_____. **MIDI 2.0 Specification Overview**. [S.l.], 2023. MIDI Association Document: M2-100-U, Version 1.1; Draft Date: 11-May-2023; Published: 15-Jun-2023. Disponível em: <<https://www.midi.org>>.

_____. **MIDI Capability Inquiry (MIDI-CI)**. [S.l.], 2023. MIDI Association Document: M2-101-UM, Version 1.2; Draft Date: 11-May-2023; Published: 15-Jun-2023. Disponível em: <<https://www.midi.org>>.

_____. **MIDI Clip File Specification**. [S.l.], 2023. MIDI Association Document: M2-116-U, Version 1.0; Draft Date: 11-May-2023; Published: 15-Jun-2023. Disponível em: <<https://www.midi.org>>.

_____. **Universal MIDI Packet (UMP) Format and MIDI 2.0 Protocol**. [S.l.], 2023. MIDI Association Document: M2-104-UM, Version 1.1.2; Draft Date: 27-Oct-2023; Published: 2023. Disponível em: <<https://www.midi.org>>.

MIRANDA, E. R. **Computer Sound Design: Synthesis Techniques and Programming**. Oxford; Boston: Focal Press, 2002. ISBN 978-0240516931.

MOORE, F. R. **Elements of Computer Music**. Englewood Cliffs, N.J.: Prentice Hall, 1990. ISBN 978-0132637636.

O'MODHRAIN, S. A framework for the evaluation of digital musical instruments. **Computer Music Journal**, v. 35, p. 28–42, 03 2011.

OVIATT, S. *et al.* **The Handbook of Multimodal-Multisensor Interfaces, Volume 2: Signal Processing, Architectures, and Detection of Emotion and Cognition**. [S.l.]: Association for Computing Machinery and Morgan & Claypool Publishers, 2018. (ACM Books #21). ISBN 9781970001709.

_____. **The Handbook of Multimodal-Multisensor Interfaces: Language Processing, Software, Commercialization, and Emerging Directions, Volume 3**. [S.l.]: Association for Computing Machinery and Morgan & Claypool Publishers, 2019. (ACM Books #23). ISBN 9781970001747.

RAUTARAY, S. S.; AGRAWAL, A. Vision based hand gesture recognition for human-computer interaction: A survey. **Artificial Intelligence Review**, v. 37, n. 1, p. 1–54, 2012.

ROADS, C. **The Computer Music Tutorial**. Cambridge, Mass.: MIT Press, 1996. ISBN 978-0262181580.

ROGERS, Y.; SHARP, H.; PREECE, J. **Interaction Design: Beyond Human-Computer Interaction**. 6. ed. [S.l.]: Wiley, 2023. ISBN 111990109X, 9781119901099.

ROSSING, T. D. **The Science of Sound**. 3. ed. San Francisco: Addison-Wesley, 2002. ISBN 978-0805385652.

ROVAN, J.; WANDERLEY, M.; DUBNOV, S. Instrumental gestural mapping strategies as expressivity determinants in computer music performance. 12 1997.

SHOTTON, J. *et al.* Real-time human pose recognition in parts from single depth images. **Commun. ACM**, Association for Computing Machinery, New York, NY, USA, v. 56, n. 1, p. 116–124, jan. 2013. ISSN 0001-0782. Disponível em: <<https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/BodyPartRecognition.pdf>>.

VAMVAKOUSIS, Z.; RAMIREZ, R. The EyeHarp: A gaze-controlled digital musical instrument. **Frontiers in Psychology**, v. 7, 2016. ISSN 1664-1078. Disponível em: <<https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2016.00906>>.

VOGIATZIDAKIS, P.; KOUTSABASIS, P. Gesture elicitation studies for mid-air interaction: A review. **Multimodal Technologies and Interaction**, v. 2, n. 4, p. 65, 09 2018.

WANDERLEY, M. M. Gestural control of music. **International Workshop Human Supervision and Control in Engineering and Music**, IRCAM / McGill University, 2001.

WANDERLEY, M. M.; ORIO, N. Evaluation of input devices for musical expression: Borrowing tools from hci. **Comput. Music J.**, MIT Press, Cambridge, MA, USA, v. 26, n. 3, p. 62–76, set. 2002. ISSN 0148-9267. Disponível em: <<https://doi.org/10.1162/014892602320582981>>.

WANDERLEY, M. M. *et al.* The role of gestures in musical interaction. **Trends in Gestural Control of Music**, 2000.

ZHANG, W. *et al.* A gesture-based teleoperation system for compliant robot motion. **Applied Sciences**, v. 9, n. 24, 2019. ISSN 2076-3417. Disponível em: <<https://www.mdpi.com/2076-3417/9/24/5290>>.

ZöLZER, U. **DAFX: Digital Audio Effects**. Chichester, West Sussex; Hoboken, NJ: John Wiley & Sons, 2011. ISBN 978-0470665992.