



Sebastião Filipe Pedrosa

Licenciado em Ciências da Engenharia Electrotécnica  
e de Computadores

# Sistema EMG para controlo MIDI

MESTRADO EM ENGENHARIA ELECTROTÉCNICA E DE  
COMPUTADORES

Universidade NOVA de Lisboa  
Julho, 2023



## Sistema EMG para controlo MIDI

**Sebastião Filipe Pedrosa**

Licenciado em Ciências da Engenharia Electrotécnica e de  
Computadores

**Orientadora:** Anikó Katalin Horváth da Costa,  
Professora Auxiliar, FCT-NOVA

### **Júri:**

**Presidente:** Pedro Miguel Figueiredo Amaral,  
Professor Auxiliar, FCT-NOVA

**Arguentes:** Bruno João Nogueira Guerreiro,  
Professor Auxiliar, FCT-NOVA

**Orientadora:** Anikó Katalin Horváth da Costa,  
Professora Auxiliar, FCT-NOVA

## **Sistema EMG para controlo MIDI**

Copyright © Sebastião Filipe Pedrosa, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.



*Dedico esta dissertação  
ao meu pai, Jorge dos Santos Pedrosa,  
e a minha mãe, Anabela da Graça Jorge*



## Agradecimentos

Pela sua disponibilidade e acompanhamento absolutamente surreais que me tem fornecido desde a primeira aula de Sistemas Lógicos 1 até ao fim desta dissertação, quero agradecer a minha orientadora Anikó Katalin Horváth da Costa.

Por ser, aos meus olhos, uma autêntica referência de disciplina e trabalho árduo, quero agradecer ao meu primo Ludovic Santos. Por me ter posto na linha sempre que foi preciso, quero agradecer ao meu irmão Alexis Pedrosa. Por todos os agoniantes, mas muito necessários “então e como é que tá a tese?”, quero agradecer ao meu pai Jorge dos Santos Pedrosa e a minha mãe Anabela da Graça Jorge. Por todo o apoio que me deram, muitas vezes sem eu sequer pedir nem saber que precisava, quero agradecer a minha tia Sélia Jorge e a minha avó Irene Jorge.

Pela sua sugestão em utilizar um filtro de média, desbloqueando uma barreira a superar desta dissertação, quero agradecer a minha ex-colega de trabalho Júlia Pinheiro. Por me ter introduzido ao Ableton Live, aos sintetizadores, e indirectamente inspirar o tema desta dissertação, quero agradecer ao meu ex-colega de curso Ricardo Crespo.



# Resumo

---

O *Musical Instrument Digital Interface* (MIDI) é um standard usado já desde os inícios dos anos 80, e tem vindo a ter cada vez mais uso na comunidade musical. Trata-se de um protocolo usado para interligar dispositivos de áudio, instrumentos musicais, computadores, interfaces, e outros controladores MIDI. Um controlador MIDI é um dispositivo que envia e recebe mensagens MIDI para controlar parâmetros (mensagens de *control change*), representar e tocar notas musicais (*note\_on*). Estes controladores existem nos mais diversos formatos de forma simular e adaptar toda e qualquer performance musical.

Sejam teclados MIDI para simular pianos ou as superfícies de contacto de uma bateria electrónica, controladores MIDI podem fazer uso de toda e qualquer grandeza para converter fenómenos físicos em novas formas de expressão musical. Para esta dissertação, o objectivo é, em vez de teclas ou pancadas em superfícies, disparar notas com gestos da mão. Para detectar estes gestos da mão, será monitorizada a actividade causada pela mesma, ou seja, a actividade muscular ao nível do antebraço.

Assim, o objectivo desta dissertação é implementar um controlador MIDI capaz de converter gestos da mão em performance musical.

Palavras-chave: MIDI, EMG, reconhecimento de padrões, machine learning, classificador



# Abstract

---

The Musical Instrument Digital Interface (MIDI) is a standard that has been in use since the early 1980s, and has been increasingly used in the music community. It is a protocol used to interconnect audio devices, musical instruments, computers, interfaces, and other MIDI controllers. A MIDI controller is a device that sends and receives MIDI messages to control parameters (control change messages), represent and play musical notes (note\_on). These controllers exist in the most diverse formats in order to simulate and adapt any musical performance.

Whether it's MIDI keyboards to simulate pianos or the contact surfaces of an electronic drum set, MIDI controllers can make use of any and all magnitudes to convert physical phenomena into new forms of musical expression. For this dissertation, the objective is, instead of keys or tapping on surfaces, to trigger notes with hand gestures. To detect these hand gestures, the activity caused by them will be monitored, that is, the muscle activity at the level of the forearm.

Thus, the aim of this dissertation is to implement a MIDI controller capable of converting hand gestures into musical performance.

Keywords: MIDI, EMG, pattern detection, machine learning, classifier



# Índice

<b>RESUMO.....</b>	<b>VII</b>
<b>ABSTRACT .....</b>	<b>IX</b>
<b>ÍNDICE.....</b>	<b>XI</b>
<b>LISTA DE FIGURAS .....</b>	<b>XIII</b>
<b>LISTA DE TABELAS .....</b>	<b>XV</b>
<b>ACRÓNIMOS E SIGLAS .....</b>	<b>XVII</b>
<b>1. INTRODUÇÃO.....</b>	<b>1</b>
1.1. INTRODUÇÃO E MOTIVAÇÃO.....	1
1.2. OBJECTIVOS E CONTRIBUIÇÕES.....	2
1.3. ESTRUTURA DO DOCUMENTO .....	3
<b>2. ESTADO DA ARTE.....</b>	<b>5</b>
2.1. MUSICAL INSTRUMENT DIGITAL INTERFACE E INSTRUMENTOS ELECTRÓNICOS .....	5
2.2. SENSORES BIOMÉTRICOS MIOGRÁFICOS.....	10
<i>Actividade muscular e potenciais biológicos .....</i>	<i>10</i>
<i>Sistema baseado em sensores de electromiograma.....</i>	<i>12</i>
2.3. RASTREIO DE MOVIMENTOS CORPORAIS.....	13
2.3.1. <i>Soluções outside-in .....</i>	<i>15</i>
2.3.2. <i>Soluções inside-out .....</i>	<i>17</i>
2.3.3. <i>Soluções inside-in .....</i>	<i>20</i>
2.3.4. <i>Soluções baseadas em sinais biométricos EMG.....</i>	<i>24</i>
<b>3. PROPOSTA .....</b>	<b>33</b>
3.1. POSIÇÃO DOS SENSORES .....	34
3.2. LEITURA DE SINAL .....	35
3.3. ACONDICIONAMENTO DE SINAL E DETECÇÃO DE ATIVIDADE GESTUAL .....	36
3.4. TREINO DE CLASSIFICADORES.....	36
3.5. CONVERSÃO DE GESTOS PARA MENSAGENS MIDI.....	36
<b>4. IMPLEMENTAÇÃO .....</b>	<b>39</b>
4.1. ARQUITECTURA.....	39
4.1.1. HARDWARE.....	39

4.1.2.	SOFTWARE.....	40
4.2.	CAPTAÇÃO EMG .....	42
4.2.1.	LEITURA DOS SENSORES NO ARDUINO .....	42
4.2.2.	TRANSMISSÃO PARA O RASPBERRY PI.....	43
4.3.	PROCESSAMENTO DE SINAL .....	45
4.4.	EXTRACÇÃO DE CARACTERÍSTICAS E DETECÇÃO .....	46
4.5.	CLASSIFICAÇÃO DE GESTOS .....	49
4.6.	PROCESSAMENTO MIDI .....	50
<b>5.</b>	<b>DISCUSSÃO DE RESULTADOS .....</b>	<b>53</b>
5.1.	LEITURA DA INFORMAÇÃO EMG.....	53
5.2.	PROCESSAMENTO DO SINAL .....	53
5.3.	DETECÇÃO E CLASSIFICAÇÃO .....	54
5.4.	CONVERSÃO PARA MIDI.....	54
5.5.	VALIDAÇÃO DOS DADOS .....	55
<b>6.</b>	<b>CONCLUSÃO E TRABALHO FUTURO.....</b>	<b>59</b>
6.1.	TRABALHO FUTURO.....	59
<b>7.</b>	<b>REFERÊNCIAS.....</b>	<b>61</b>
<b>A.</b>	<b>MATRIZES DE CONFUSÃO .....</b>	<b>67</b>
A.1.	RANDOM FOREST.....	67
A.2.	SUPPORT VECTOR MACHINES .....	68
A.3.	REDE NEURONAL .....	69

# Lista de Figuras

FIGURA 1.1: INTERFACE DO SOFTWARE ABLETON LIVE 9 SUITE COM CLIP MIDI APRESENTADO .....	3
FIGURA 2.1: ADDICTIVE DRUMS 2 DA XLN AUDIO E MASSIVE DA NATIVE INSTRUMENTS.....	5
FIGURA 2.2: MONTAGEM BÁSICA DE UM CONTROLADOR MIDI POR USB .....	6
FIGURA 2.3: ESTRUTURA DE UMA MENSAGEM MIDI.....	7
FIGURA 2.4: INTERFACE GRÁFICA DO SOFTWARE ABLETON LIVE 9 SUITE.....	9
FIGURA 2.5: TROCA DE IÕES NA MEMBRANA EM FUNÇÃO DO TEMPO .....	10
FIGURA 2.6: DIFERENTES FASES DO POTENCIAL AO NÍVEL DA MEMBRANA .....	11
FIGURA 2.7: SINAL MEDIDO APÓS QUATRO CONTRACÇÕES MUSCULARES DO BÍCEPS .....	11
FIGURA 2.8: DIAGRAMA DE BLOCOS DA PLACA SHIELD-EKG-EMG .....	12
FIGURA 2.9: MONTAGEM NO BRAÇO PARA MEDIÇÃO DE ACTIVIDADE MUSCULAR. ....	12
FIGURA 2.10: PLACA SHIELD-EKG-EMG.....	13
FIGURA 2.11: SENSORES USADOS PARA A CAPTURA 3D.....	16
FIGURA 2.12: DIAGRAMA DE BLOCOS DOS UMIs DA MONTAGEM NA LUVA .....	18
FIGURA 2.13: ACCELEGLOVE.....	19
FIGURA 2.14: ORIENTAÇÃO DOS EIXOS RELATIVAMENTE AOS DEDOS .....	20
FIGURA 2.15: CINCO MOVIMENTOS DEFINIDOS .....	21
FIGURA 2.16: CYBER GLOVE USADA NO PROJECTO CYBER COMPOSER .....	23
FIGURA 2.17: DIFERENTES FASES DE RECONHECIMENTO GESTUAL EM SISTEMAS DE CONTROLO EMG .....	25
FIGURA 2.18: LUVA EXOESQUELETO DE UM SÓ SENSOR.....	28
FIGURA 2.19: SEIS GESTOS DIFERENCIADOS E AS TRÊS POSIÇÕES DE BRAÇO.....	29
FIGURA 2.20: RESULTADOS DA FASE DE AVALIAÇÃO .....	31
FIGURA 3.1: DIAGRAMA DE BLOCOS DA MONTAGEM FINAL PROPOSTA.....	34
FIGURA 3.2: ESQUEMA DA POSIÇÃO DOS SENSORES NO BRAÇO DIREITO.....	35
FIGURA 4.1: LIGAÇÃO ENTRE ARDUINO UNO E RASPBERRY PI.....	39
FIGURA 4.2: ARQUITECTURA DO CIRCUITO FINAL .....	40
FIGURA 4.3: ESQUEMA FUNCIONAL DA LÓGICA DO CONTROLADOR .....	41
FIGURA 4.4: ESQUEMA DE COMUNICAÇÃO ENTRE COMPONENTES.....	42
FIGURA 4.5: POSIÇÃO DOS SENSORES NO BRAÇO DIREITO .....	43
FIGURA 4.6: PROTOCOLO DE COMUNICAÇÃO PARA UM CICLO DE LEITURA DOS TRÊS VALORES EMG .....	44
FIGURA 4.7: SINAIS EMG ANTES (AZUL) E DEPOIS (VERMELHO) DO FILTRO DE MÉDIA .....	45
FIGURA 4.8: CAPTAÇÃO DE QUATRO MOVIMENTOS DISTINTOS (CANAL 0).....	47
FIGURA 4.9: CAPTAÇÃO DE QUATRO MOVIMENTOS DISTINTOS (CANAL 1).....	47
FIGURA 4.10: CAPTAÇÃO DE QUATRO MOVIMENTOS DISTINTOS (CANAL 2). ....	48

FIGURA 4.11: QUATRO GESTOS ESCOLHIDOS PARA DETECÇÃO E CLASSIFICAÇÃO.....	49
FIGURA 4.12: FLUXOGRAMA DO PROCESSO EXECUTADO PELO CONTROLADOR MIDI IMPLEMENTADO.....	50
FIGURA 4.13: PERFORMANCE MUSICAL EM ABLETON LIVE 9.....	52
FIGURA 4.14: FORMATO DA MATRIZ DE CONFUSÃO DE UM CLASSIFICADOR.....	56
FIGURA 4.15: EXEMPLO DE MATRIZ DE CONFUSÃO.....	56

# Lista de Tabelas

TABELA 2.1: POSSÍVEIS MENSAGENS MIDI RELEVANTES PARA O PROBLEMA APRESENTADO .....	8
TABELA 2.2: DESEMPENHO DE DIFERENTES ALGORITMOS TESTADOS (ADAPTADO DE [42]).....	27
TABELA 4.1: RESULTADOS DE PRECISÃO DOS CLASSIFICADORES ESCOLHIDOS.....	50
TABELA 4.2: MAPEAMENTO DE GESTOS PARA NOTAS MUSICAIS.....	51
TABELA 4.3: TABELA DE CONFUSÃO PARA O CLASSIFICADOR RANDOM FOREST .....	55
TABELA 4.4: TABELA DE CONFUSÃO PARA O CLASSIFICADOR DE SUPPORT VECTOR MACHINES .....	55
TABELA 4.5: TABELA DE CONFUSÃO PARA O CLASSIFICADOR DE REDE NEURONAL.....	55
TABELA 4.6: MÉTRICAS DE DESEMPENHO PARA RANDOM FOREST .....	57
TABELA 4.7: MÉTRICAS DE DESEMPENHO PARA SUPPORT VECTOR MACHINES.....	57
TABELA 4.8: MÉTRICAS DE DESEMPENHO PARA REDE NEURONAL .....	57
TABELA A.1: MATRIZ DE CONFUSÃO PARA O GESTO DE POLEGAR LEVANTADO.....	67
TABELA A.2: MATRIZ DE CONFUSÃO PARA O GESTO DE MÃO FECHADA.....	67
TABELA A.3: MATRIZ DE CONFUSÃO PARA O GESTO DE MÃO ABERTA .....	67
TABELA A.4: MATRIZ DE CONFUSÃO PARA O GESTO DE TESOURA .....	68
TABELA A.5: MATRIZ DE CONFUSÃO PARA O GESTO DE POLEGAR LEVANTADO.....	68
TABELA A.6: MATRIZ DE CONFUSÃO PARA O GESTO DE MÃO FECHADA.....	68
TABELA A.7: MATRIZ DE CONFUSÃO PARA O GESTO DE MÃO ABERTA .....	69
TABELA A.8: MATRIZ DE CONFUSÃO PARA O GESTO DE TESOURA .....	69
TABELA A.9: MATRIZ DE CONFUSÃO PARA O GESTO DE POLEGAR LEVANTADO.....	69
TABELA A.10: MATRIZ DE CONFUSÃO PARA O GESTO DE MÃO FECHADA .....	70
TABELA A.11: MATRIZ DE CONFUSÃO PARA O GESTO DE MÃO ABERTA.....	70
TABELA A.12: MATRIZ DE CONFUSÃO PARA O GESTO DE TESOURA .....	70



## Acrónimos e siglas

AVC	Acidente vascular cerebral
BMS	Bit mais significativo
DSP	Digital signal processor
ECG	Electrocardiograma
EEG	Electroencefalograma
EMG	Electromiograma
EOG	Electrooculograma
FN	False negative
FP	False positive
MAV	Mean absolute values
MIDI	Musical Instrument Digital Interface
MISO	Master In Slave Out
MOSI	Master Out Slave In
RMS	Root Mean Square
SCLK	Serial Clock
SRAR	Sistema de referência de altitude e rumo
SSC	Slope sign changes
SVM	Support vector machine
TN	True negative
TNRG	True negative rate for grasping
TP	True positive
TPRG	True positive rate for grasping
UART	Asynchronous Receiver/Transmitter
UMI	Unidade de medição inercial
WL	Waveform lengths
ZC	Zero crossings

# 1. Introdução

## 1.1. Introdução e motivação

Música é uma forma de arte muito comum no nosso quotidiano. Sendo muito acessível para qualquer ouvinte, esta também é acessível, mesmo que na sua forma mais simples, para qualquer pessoa independentemente do seu nível de experiência. Hoje em dia, esta já não está limitada a ser executada apenas acusticamente. Sintetizadores, *samplers* e *drum machines* são apenas alguns exemplos de instrumentos electrónicos. Estes instrumentos electrónicos, tipicamente recebem input e comunicam entre si utilizando o protocolo Musical Instrument Digital Interface (MIDI).

Essencialmente, MIDI é um protocolo utilizado para descrever e controlar performance musical. Um músico pode gravar a sua performance com um controlador MIDI, e com isso obter uma sequência de notas, cada uma com a sua duração e força associada. Essa sequência de notas MIDI pode ser por fim usado em qualquer instrumento electrónico capaz de processar mensagens MIDI. Por exemplo, a informação de um ritmo de bateria pode ser enviada para um módulo de uma *drum machine* para que esta, utilizando amostras sonoras que armazena nela mesma, reproduza efectivamente aquilo que o ouvinte comum reconhece como um ritmo de bateria. Ou seja, enquanto o MIDI descreve toda a informação de uma determinada performance, não guarda nenhum dado sobre o som em si que vai ser reproduzido. Adicionalmente, existe mais informação que pode ser armazenada para além de notas musicais. Um músico pode por exemplo querer mudar um determinado parâmetro a meio de uma performance (a frequência de corta de um filtro de um sintetizador, por exemplo). Para tal, são usados uns dispositivos chamados de controladores de MIDI.

Dito isto, com o uso de controladores MIDI, surgem várias vantagens:

- Fácil manipulação e mapeamento.

- Possibilidade de transposição de uma determinada melodia para qualquer instrumento.
- Possibilidade de automação de parâmetros.
- Personalização, no sentido em que permite mapeamentos personalizados ao gosto do músico.
- A possibilidade de lançar clips MIDI sincronizados permite a um músico de essencialmente conduzir uma “orquestra”.

O que é proposto, é um controlador MIDI que em vez de receber input através de botões e potenciômetros, receberá input dos movimentos da mão do operador possibilitando uma operação mais intuitiva.

Além de contar como nova forma de expressão, pode também ter o seu contributo no tratamento de crianças com necessidades especiais e pacientes de doenças com efeitos ao nível de capacidades motoras como pacientes de acidentes vascular cerebrais (AVC). Existem muitos estudos que indicam fortes progressos nestes casos através do uso de terapia musical em crianças com necessidades especiais [1][2], adultos com osteoartrite [3], bem como pacientes de AVC [4].

Dito isto, uma nova forma de expressão musical, bem como o uso total de um membro, pode trazer também uma nova abordagem, possibilitando novos métodos de terapia musical, dando ênfase na coordenação olho-mão.

## 1.2. Objectivos e contribuições

O objectivo desta dissertação passará por desenvolver um controlador que receberá sinais do braço usando sensores biométricos para medir a flexão e posição do braço e converter esse comportamento para mensagens de MIDI capazes de ser recebidas por qualquer software que seja capaz de as interpretar.

Dito isto, neste projecto será feita a captação dos movimentos e posição dos membros através de sensores biométricos, cujo sinal será enviado para um controlador Raspberry Pi 4, que por sua vez irá interpretar estes sinais e enviar a correspondente informação MIDI para um computador onde este será processado por um software que faça uso do mesmo como o Ableton Live, apresentado na Figura 1.1.

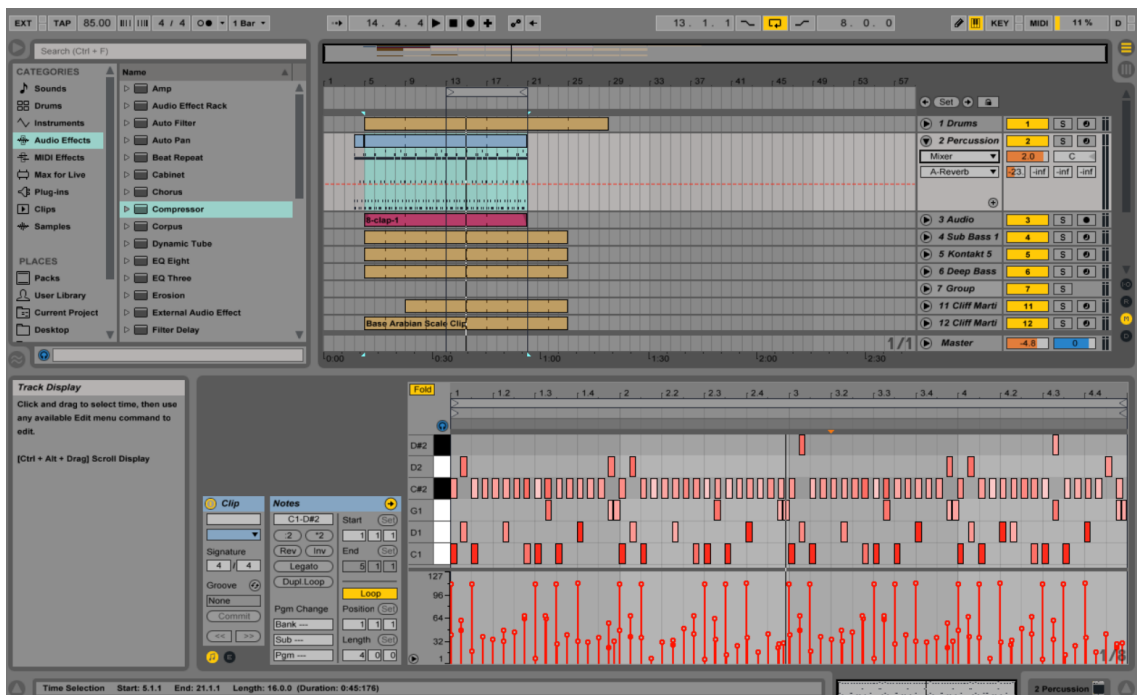


Figura 1.1: Interface do software Ableton Live 9 Suite com clip MIDI apresentado

### 1.3. Estrutura do documento

Este documento encontra-se dividido em seis secções. Nesta secção foram definidos os objectivos e motivações desta dissertação.

Na secção 2 são apresentadas tecnologias usadas para esta tese, bem como um enquadramento teórico sobre sinais e sensores electro miográficos. São também apresentados projectos de interesse que serviram de ponto de partida para definir a solução final para o controlador final.

Na secção 3 é feita uma proposta inicial para a arquitectura e fluxo de sinal desde a actividade muscular até ao som musical resultante. São definidas a arquitectura do sistema, posicionamento dos sensores, leitura e acondicionamento do sinal, treino de classificadores e conversão para mensagens MIDI. Nesta secção são também evidenciadas possíveis falhas no planeamento que foram alteradas a medida que o projecto foi sendo desenvolvido.

Na secção 4 é apresentada a implementação final, evidenciando problemas que foram encontrados e como foram resolvidos, como por exemplo, a escolha do classificador mais adequado para esta aplicação.

Na secção 0 são discutidos os resultados finais em cada fase da implementação, bem como a validação dos testes dos classificadores, concluindo com uma validação completa de todo o sistema.

Finalmente, na secção 6 são discutidos desenvolvimentos futuros a melhorar no sistema de forma a obter melhores resultados nos classificadores, bem como melhoramentos mais funcionais como a transmissão MIDI por cabo MIDI em vez de USB.

## 2. Estado da arte

Este capítulo foca-se em introduzir os conceitos essenciais para esta dissertação. Começando pela estrutura do MIDI e a forma como interagem os seus dispositivos, as diferentes formas de captação de informação sobre movimentos corporais, bem como projectos propriamente ditos que tentaram previamente abordar este tema.

### 2.1. Musical Instrument Digital Interface e instrumentos electrónicos

MIDI é um protocolo desenvolvido em 1982, para possibilitar a comunicação entre dispositivos MIDI, sintetizadores e outros instrumentos electrónicos. Definindo diversos eventos de performance e controlo, mensagens MIDI essencialmente transportam toda essa informação de uma performance musical [5]. Ou seja, estas convertem a informação de uma performance musical para o domínio digital sem afectar o som propriamente dito. Essas mensagens são por sua vez usadas por instrumentos electrónicos que por sua vez recebem mensagens MIDI e utilizam-nas para gerar som conforme estiverem dimensionados. Este pode ser um ficheiro de áudio (*samplers* e *drum machines*), ou pode ser mesmo a geração de um sinal como uma onda quadrada ou sinusoidal. Quando estes instrumentos funcionam inteiramente por software estes chamam-se instrumentos virtuais[6].



Figura 2.1: Addictive Drums 2 da XLN Audio e Massive da Native Instruments

Por exemplo, na Figura 2.1 estão apresentados dois instrumentos virtuais. A esquerda está o Addictive Drums 2, que é essencialmente um software que serve para simular uma caixa de ritmos. Conforme a nota que receber poderá reproduzir o som de um timbalão, de uma tarola, ou até mesmo um prato. Por exemplo, por defeito, a nota C1 de um teclado MIDI está mapeada para o som de bombo. Por outro lado, a direita está representado o Massive da Native Instruments. Este instrumento virtual trata-se de um sintetizador digital com uma vasta e discutivelmente completa gama de efeitos e configurações. Mapeando estas configurações a um controlador MIDI, tornam-se então mais facilmente configuráveis em tempo real. É importante referir que MIDI não usa notação musical, mas sim mensagens binárias genéricas. Por consequente, estas podem ser lidas e/ou enviadas seja em que contexto for [6].

Na Figura 2.2 está representada uma montagem simples em que um controlador Novation Impulse 25 está a enviar mensagens MIDI para uma interface áudio, que por sua vez envia para o computador.



Figura 2.2: Montagem básica de um controlador MIDI por USB

É importante referir que os oito controladores centrais contidos no quadrado amarelo serão os controladores de interesse. Por exemplo, na imagem está seleccionado o controlador 21 (logo o primeiro do canto superior esquerdo) que está neste momento com o valor 116.

Uma mensagem MIDI é composta por três bytes. O primeiro byte é o byte de estado, tem o seu bit mais significativo (BMS) a '1' e serve para distinguir o tipo de mensagem. Os outros dois bytes são bytes de dados e têm os seus respectivos BMS definidos a '0'. Mensagens MIDI dividem-se em dois grupos, as mensagens de sistema e as mensagens de canal. Dispositivos MIDI estão normalmente a escuta num canal específico, que pode ir de 0 a 15. Ou seja, uma mensagem MIDI pode ser do tipo canal em que apenas os dispositivos a escuta nesse canal específico a vão receber, ou então pode ser uma mensagem de sistema, onde todos os dispositivos irão receber essa mensagem independente do canal. O primeiro byte está sempre definido a '1' para o diferenciar como byte de estado, os três seguintes definem o tipo de mensagem, e por fim os restantes quatro definem o canal em questão. Na Figura 2.3, está apresentada a estrutura básica de uma mensagem MIDI.

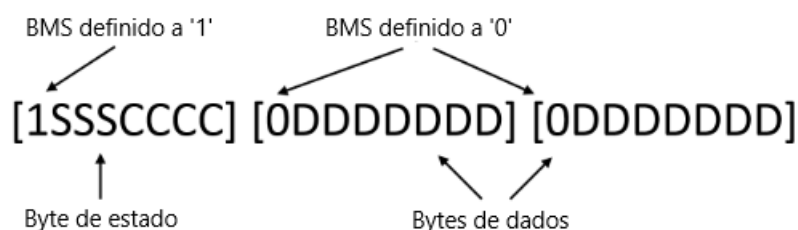


Figura 2.3: Estrutura de uma mensagem MIDI (adaptado de [5])

Entrando em mais detalhe, estas podem ser de cinco tipos:

- *Channel voice*;
- *Channel mode*;
- *System common*;
- *System real-time*;
- *System exclusive*;

Destas cinco, as que são mais importantes para o âmbito desta dissertação, são as mensagens de *channel voice*. Estas são usadas na transmissão de informação em tempo real e transmitem qualquer mudança num determinado controlador. Ou seja, se uma nota for premida ou se um controlo for alterado, serão enviadas mensagens MIDI de *channel voice* com essas informações. Dentro deste grupo ainda é possível dividir em mais mensagens. As que são de maior interesse para esta tese serão as mensagens de *control change*.

**Tabela 2.1: Possíveis mensagens MIDI relevantes para o problema apresentado**

Byte de estado	Tipo de mensagem	Dados necessários	
		Byte de dados 1	Byte de dados 2
1000xxxx	Note off	Nota (0-127)	Velocidade (0-127)
1001xxxx	Note on	Nota (0-127)	Release Velocity (0-127)
1010xxxx	Polyphonic aftertouch	Nota (0-127)	Aftertouch value
1011xxxx	Control change	Identificação do controlador	Valor (0-127)
1100xxxx	Program change	Identificação do programa	
1101xxxx	Aftertouch	Valor de aftertouch (0-127)	
1110xxxx	Pitch bend	Pode usar um byte (0-127) ou dois bytes (0 - 16383)	
1111xxxx	System	Número de bytes de dados variável	

As mensagens de *control change* essencialmente servem para mudar parâmetros de um determinado dispositivo MIDI. Tal como foi visto na Figura 2.1, existe toda uma gama de parâmetros para os quais estas mensagens podem ser mapeadas. Uma frequência de corte de um filtro, uma panorâmica ou até mesmo o estado de um oscilador<sup>1</sup>. Estas mensagens ainda se subdividem em dois tipos, controlo contínuo e controlo discreto. No controlo contínuo entende-se toda uma gama de valores de 0 a 127 para determinar um certo parâmetro (e.g.

<sup>1</sup> O NI Massive permite ter até três osciladores a gerarem um sinal em simultâneo, dependendo do seu estado, se estiverem ligados ou desligados.

uma frequência de corte). No caso do controlo discreto entendem-se apenas dois valores, ou seja, activo ou desactivo. Estes são normalmente usados como interruptores para ligar ou desligar certos efeitos, ou outros instrumentos. Tal como foi referido anteriormente, mensagens MIDI dividem-se em três bytes. Como tal uma mensagem de *control change*, tem por norma o mesmo byte de estado onde é definido o canal e o tipo de mensagem (será *channel voice*), um byte onde é identificado o parâmetro a alterar, e por fim um byte a definir o novo valor para esse parâmetro [5], [6].

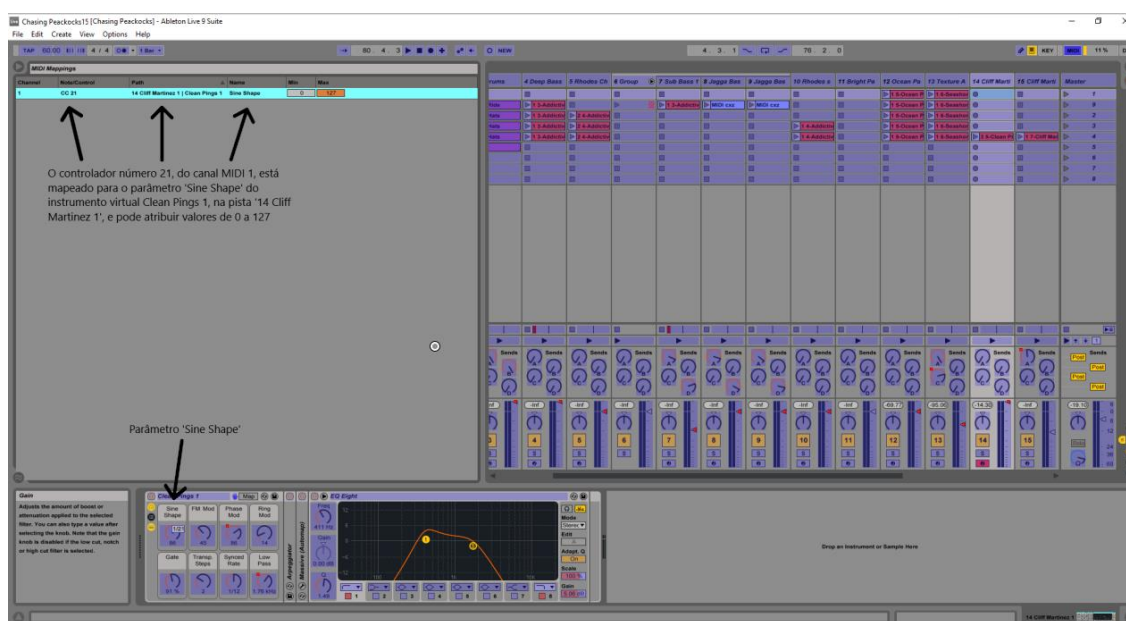


Figura 2.4: Interface gráfica do software Ableton Live 9 Suite para mapeamento de um parâmetro no software para um controlador no hardware (Novation Impulse 25).

Na Figura 2.4, está representada a forma como são associados controladores do hardware aos parâmetros que se pretendem controlar no software Ableton Live 9 Suite. Neste caso, observa-se que o controlador 21 (primeiro controlo que foi visto no teclado da Figura 2.2) está mapeado para o mapeado para o parâmetro 'Sine Shape', do sintetizador 'Clean Pings 1', da pista 14 'Cliff Martinez 1'. Possibilitando então um controlo mais intuitivo com hardware.

Concluindo, a montagem pretendida para esta dissertação será algo semelhante, excepto que em vez de um controlador MIDI Novation Impulse 25,

será utilizado um controlador dimensionado para receber input dos gestos da mão e do braço do utilizador. Adicionalmente, será utilizado um controlador Raspberry Pi 4, pelas suas capacidades de processamento de MIDI e pela sua performance a altas frequências de amostragem.

## 2.2. Sensores biométricos miográficos

Sensores biométricos servem para medir potenciais biológicos gerados por actividade humana. Existem vários tipos, visto que este pode variar no seu tipo consoante o que se pretender medir. Segundo Reilly *et al.* [7], alguns destes incluem:

- Electrocardiograma (ECG) mede a actividade no coração;
- Electroencefalograma (EEG) mede a actividade no cérebro;
- Electrooculograma (EOG) mede a actividade nos olhos;
- Electromiograma (EMG) mede a actividade no tecido muscular;

Para o objectivo desta dissertação, o mais importante é o electromiograma, por possibilitar a medição de actividade ao nível do braço.

### Actividade muscular e potenciais biológicos

Ao estimular o nervo motor, é causada a libertação de neurotransmissores pelo nervo. Isto desencadeia um processo de troca de iões  $\text{Na}^+$  e  $\text{K}^+$  (bomba sódio-potássio) na membrana, despolarizando e repolarizando em sequência.

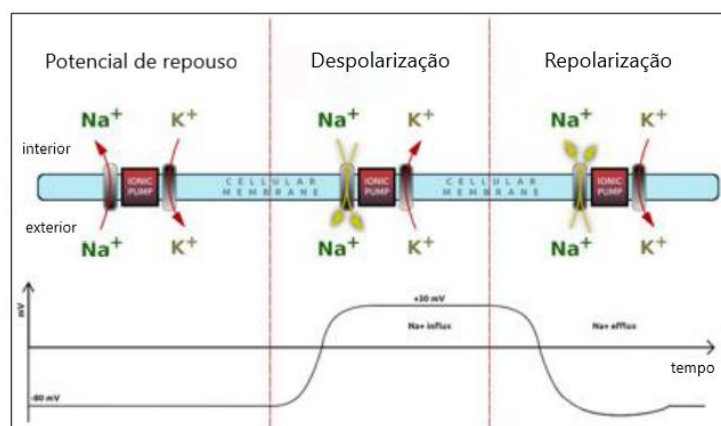
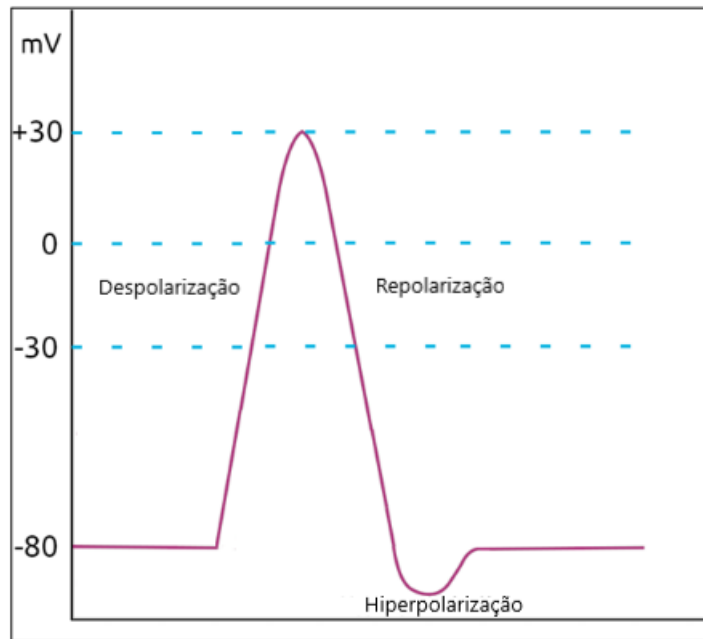
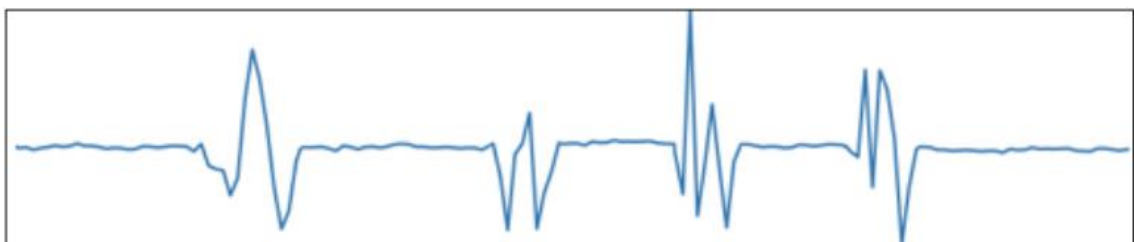


Figura 2.5: Troca de iões na membrana em função do tempo (adaptado de [8])



**Figura 2.6:** Diferentes fases do potencial ao nível da membrana (adaptado de [8])

Este processo gera um potencial variante no tempo cujo comportamento pode ser observado na Figura 2.5 e na Figura 2.6. É esta actividade que é medida pelos sensores EMG. Com isto, usando três eléctrodos (configuração bipolar com um extra para rejeição de modo comum), é possível detectar este processo e sobretudo a forma como se repete. Como é de esperar, pode ver-se pela Figura 2.7, que o sinal real será muito mais instável.



**Figura 2.7:** Sinal medido após quatro contrações musculares do bíceps (retirado de [8])

Sistema baseado em sensores de electromiograma

Em qualquer sistema de aquisição de dados, por norma, existe sempre uma fase de acondicionamento de sinal. Note-se que o sinal medido na membrana estará na ordem dos milivolts. Isto é bastante pequeno, pelo que é necessário algum processamento antes de entrar no controlador. É então para isto que existem placas já dimensionadas para esse tratamento. É o caso, por exemplo, das placas de desenvolvimento Shield-EKG-EMG da Olimex, cujos andares de processamento estão descritas na Figura 2.8 [9].

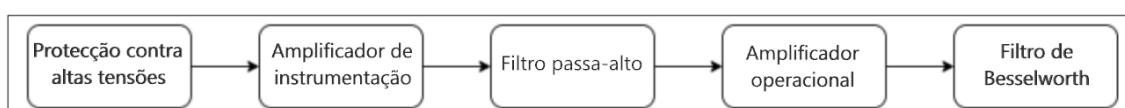


Figura 2.8: Diagrama de blocos da placa Shield-EKG-EMG<sup>2</sup> (adaptado de [29])

Estes poderão ser usados em dois pontos de interesse nesta dissertação, nomeadamente medição do sinal EMG de cada dedo e o sinal EMG do braço (bíceps e tríceps), seguindo as disposições apresentadas na Figura 2.9.

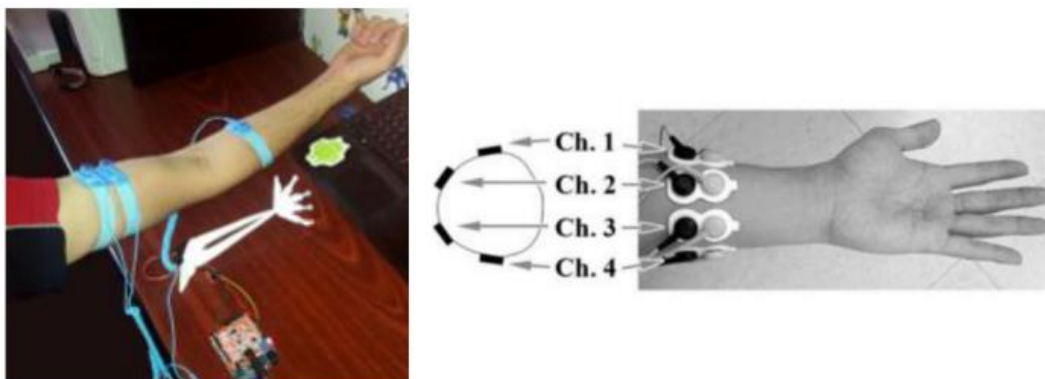


Figura 2.9: Montagem no braço para medição de actividade muscular no bícep (esquerda) e no antebraço para medição da actividade em cada dedo (direita).

<sup>2</sup> É importante referir que ainda existe um andar de rejeição de altas frequências antes do amplificador de instrumentação para estabilizar o sinal, e um filtro passa-alto mesmo antes do filtro de Besselworth. Decidiu-se não incluir na imagem apenas por questões de formatação e facilidade de leitura.

Concluindo, nesta secção foram analisados o que são sensores biométricos EMG de uma perspectiva de qual o fenómeno que é efectivamente medido. Adicionalmente, também se abordou uma montagem base para implementar um sistema de aquisição de dados baseado nos mesmos para retirar toda informação pretendida. Dito isto, será utilizado precisamente esta placa anteriormente referida e apresentada na Figura 2.10, para receber e acondicionar todo o sinal EMG e enviar para uma placa externa Raspberry Pi 4.

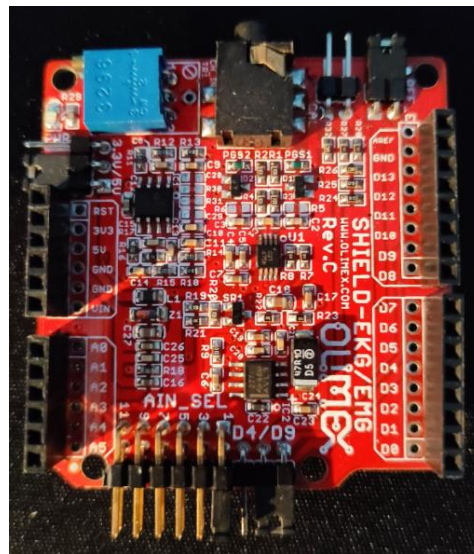


Figura 2.10: Placa Shield-EKG-EMG

### 2.3. Rastreo de movimentos corporais

Nesta secção serão analisados projectos que abordam o conceito de rastreo de movimentos da mão independente da secção de MIDI. É importante sublinhar esta dependência, visto que a captação da informação da mão é inteiramente independente da conversão da mesma para mensagens MIDI.

Para simplificar, esta dissertação irá focar-se apenas na captura de movimentos do braço e da mão. Por rastreo de movimentos corporais entende-se a captura e registo de informação sobre os movimentos de um ser humano. Essa informação é, normalmente, simplificada e retratada através de pontos chave como as articulações, mãos, pés e cabeça. Dipietro *et al.* [10] e faz um

levantamento de várias luvas, descrevendo que sensores, funcionalidades, vantagens e desvantagens.

Segundo Dipietro *et al.* e Sturman *et al.* [11], as primeiras implementações deste conceito datam de 1977 com a Sayre Glove [12] e de 1983 com a Digital Entry Data Glove [13]. A Sayre Glove usava tubos flexíveis com uma fonte de luz numa ponta e uma célula fotoelétrica na outra. Esta configuração era montada ao longo dos dedos. Com isto, a medida que os dedos dobravam, menos luz era emitida para a célula e menos tensão era gerada pela célula. Assim, era possível de correlacionar a postura da mão com a tensão de saída da célula fotoelétrica. Por outro lado, A Digital Data Glove usava sensores de toque ou proximidade para detectar se o polegar estava a tocar noutra parte do corpo, extensómetros para medir a flexão dos dedos, dois sensores de nível para detectar a inclinação da mão e dois sensores de inércia para medir a torção do braço e a flexão do pulso. Com isto, pode-se facilmente perceber que existem inúmeras formas de abordar este problema<sup>3</sup>. Por isso, para facilitar a compreensão, poderá ser benéfico a categorização destas abordagens. Segundo M. Velte *et al.* [14], os vários métodos de rastreio de movimentos do corpo humano podem ser classificados conforme as suas abordagens em três categorias, nomeadamente *outside-in*, *inside-out* e *inside-in*.

Os métodos *outside-in* são essencialmente métodos que usam sistemas ópticos. Nestes sistemas são usados sensores de imagem, como sensores RGB ou infravermelhos, de forma a captar uma ou mais imagens, e determinar as várias secções do corpo bem como as suas articulações. Esta abordagem divide-se ainda em duas, nomeadamente *outsider-in* com marcadores e sem marcadores no corpo. No caso de existirem marcadores, cria-se a necessidade para um fato para posicionar os ditos marcadores, bem como o risco de oclusão dos mesmos (a meio de um movimento um braço ficar tapado pelo outro). Por outro lado, o uso de marcadores alivia também a necessidade de mais câmaras para a obtenção de mais dados. Um exemplo de um sistema sem marcadores bastante conhecido é, por exemplo, o sistema Microsoft Kinect. Este é um sistema que usa uma câmara RGB e um sensor de profundidade baseado em infravermelhos<sup>4</sup>.

---

<sup>3</sup> É recomendada a leitura de [7] e [8] para mais exemplos como estes dois.

<sup>4</sup> Ainda inclui um conjunto de 4 microfones para possibilitar reconhecimento vocal.

Depois existem as soluções *inside-out*. Nestas são usados sistemas electromagnéticos, que consistem num conjunto de *transceivers* atados ao corpo, que emitem para um controlador, dados relativos à sua posição relativamente ao receptor (neste caso o controlador). Enquanto esta abordagem pode ser vantajosa pela sua baixa latência e ao facto de não estar sujeito a oclusões mecânicas, apresenta alguma limitação na sua área de captura relativamente ao ponto do controlador que recebe o sinal dos transmissores. Adicionalmente está também sujeito a interferência no caso de haver material metálico na proximidade [14].

Por fim, existe ainda a abordagem *inside-in*. Nesta abordagem são usados sensores espalhados pelo corpo ou parte do corpo, como extensómetros, de forma a registar, por exemplo, a amplitude da mão capturando a extensão de cada dedo. Esta abordagem também não é muito susceptível a oclusões mecânicas e permite a captura em tempo real. No entanto, têm tendência a ser muito invasivos, apresentam amostragens mais baixas, e não aproveitam totalmente a posição global do utilizador. Além disso, segundo A. Moreira et al. [15], sensores como extensómetros podem muitas vezes sofrer de saturação e interferência no sinal, deslocação dos sensores no fato, bem como longos processos de calibração.

Concluindo, foram apresentadas as diferentes formas de captura bem como as suas vantagens e desvantagens. Pode se entender que, no final, o ideal será sempre uma junção de duas capturas para corrigir certos defeitos, como será visto nas secções seguintes. Dito isto, nas próximas secções, serão vistas em mais detalhe soluções que abordem estas ideias.

### 2.3.1. Soluções *outside-in*

Apesar de não representar o foco desta tese na sua totalidade, é importante na mesma fazer uma análise deste tipo de abordagem. Nesta secção serão apresentadas duas soluções *outside-in*, ou seja, soluções baseadas em processamento de imagem.

### Microsoft Kinect

Em 2010, a Microsoft lançou o Microsoft Kinect. Esta trata-se de uma ferramenta de captura de movimentos que tem como principal objetivo servir de input para computadores Windows e para a Xbox 360. Além da captura do corpo humano, esta permite o controlo através de comandos de voz e gestos. Apresentado na Figura 2.11, este sistema é constituído por:

- Uma câmara RGB;
- Um sensor de profundidade constituído por um emissor e um receptor de infravermelhos;
- Conjunto de 4 microfones.

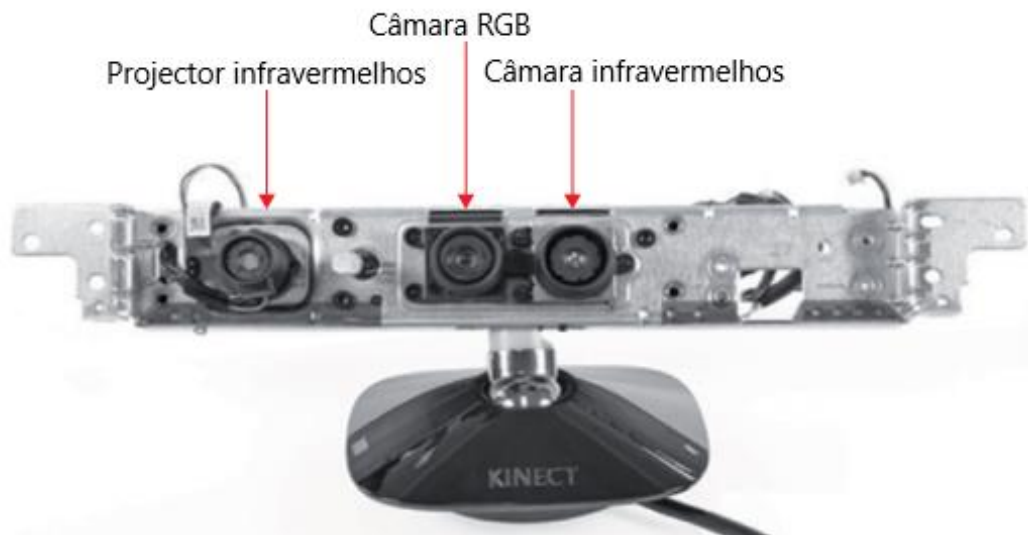


Figura 2.11: Sensores usados para a captura 3D (adaptado de [16])

Adicionalmente, o seu software está especializado para reconhecer a forma humana, utilizando informação recolhida através luz infravermelho que envia para depois receber a mesma com as deformações causadas pelo ambiente. Dessas deformações é feita então uma aproximação do ambiente captado. Reconhece até 6 pessoas em simultâneo, mas apenas consegue rastrear os movimentos de duas com toda a informação dos membros e articulações das quais consegue diferenciar, conseguindo distinguir até 20 articulações. A cada

uma destas articulações são atribuídas três coordenadas cartesianas e um estado de rastreamento. Existem três estados diferentes:

- TRACKED para quando a articulação está efectivamente a ser captada e rastreada.
- INFERRED para quando esta está a ser estimada com base posição das restantes articulações.
- NOT\_TRACKED para quando não só não está a ser rastreada como nem sequer está a ser estimada, ou seja, não existe qualquer informação sobre essa articulação.

### 2.3.2. Soluções inside-out

Como foi referido em 2.1, estas soluções funcionam a base de localização relativa de transmissores posicionados em pontos chave ao longo das articulações do corpo.

#### *Rastreio de movimentos com base num sistema de referência de altitude e rumo*

Em 2014, Moreira *et al.* [15] determinaram que o uso de extensómetros tinha demasiados problemas, como saturação de sinal e deslocação dos sensores. Adicionalmente, o uso de extensómetros requer que sejam usados em grande quantidade para conseguir retirar informação sobre a postura da mão correctamente. Relativamente aos sensores, foram então definidos os seguintes requisitos:

- Aquisição de dados em tempo real da flexão/extensão e abdução/adução dos cinco dedos;
- Calibração automática dos sensores;
- Portabilidade e baixo custo;

Por consequente, para cumprir estes requisitos, foram usadas onze unidades de medição de inércia com nove graus de liberdade, colocadas na parte de trás da luva. Estas unidades são constituídas por dois sensores:

- Um giroscópio de três eixos (L3GD20) [17];
- Um combinado de um acelerómetro com o um magnetómetro (LSM303DLHC) [18];

A resultante destes três sensores é uma unidade de medição inercial (UMI), que ao combinar as nove leituras, consegue fazer a leitura da orientação de forma rápida (proporcionada pelo giroscópio), com a referência proporcionada pelo acelerómetro e pelo magnetómetro. Por outras palavras, a junção destes dois sensores irá formar um sistema referência de altitude e rumo (SRAR). Por fim, estas onze unidades estão todas ligadas a um controlador STM32F407VG [19]. Este controlador foi escolhido por pertencer a uma família de controladores (Cortex-M4 da ARM) que proporciona altas frequências de amostragem (até 180MHz), *digital signal processor* (DSP) embutido com operações de aritmética de ciclo único e uma corrente de alimentação baixa. Com isto, a comunicação entre os sensores e o computador será garantida através do protocolo *Universal Asynchronous Receiver/Transmitter* (UART) a 512kbps. É importante referir que a junção dos UMIs é feita através de um *bus* I2C. Para isto, são então utilizados dois *multiplexers* TCA9548A [20] da Texas Instruments. Dito isto, um *multiplexer* ficará encarregue de receber do 1 ao 6, e o outro receberá os restantes. Finalmente, na Figura 2.12 está representada num diagrama de blocos toda a montagem previamente explicada [15].

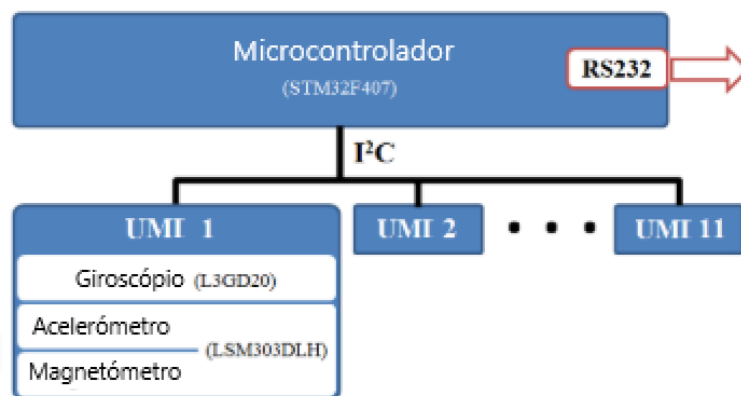


Figura 2.12: Diagrama de blocos dos UMIs da montagem na luva (adaptado de [15]).

Para o implementar SRAR com o sinal recebido das UMIs, é usado o algoritmo de gradiente descendente desenvolvido por Madwick *et al.* [21]. É importante referir que enquanto o autor deste projecto decidiu usar este algoritmo, existem ainda outros métodos como [22], [23] e [24]. Dito isto, este

algoritmo foi escolhido pela sua eficiência computacional, performance a baixas amostragens e pelos seus algoritmos de compensação de desvios ao nível do giroscópio [15] [21].

Para retirar conclusões em relação a sua performance, a luva foi testada relativamente a sua estabilidade dos sensores, precisão do SRAR, amostragem do sinal de saída e comparação do movimento da mão com o resultado que foi animado em computador.

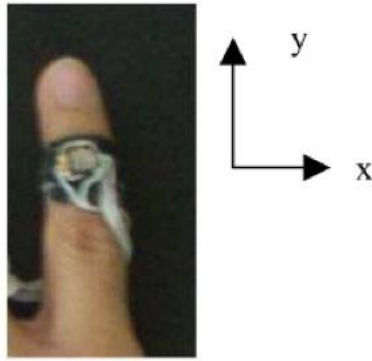
#### *AcceleGlove*

O problema que a AcceleGlove [25]-[27] tenta corrigir, em relação aos seus precedentes, é o desvio de calibração causado pela adaptação da mão do utilizador. Enquanto tem a vantagem de não precisar de um sistema de rastreamento externo, tem a desvantagem de não conseguir detectar quando uma determinada postura é feita na horizontal.



**Figura 2.13: AcceleGlove (retirado de [26])**

Apresentada na Figura 2.13, esta funciona utilizando um conjunto de seis acelerómetros montados em cada dedo e na parte de trás da palma da mão. Com isto, em vez de a “dobra” nas articulações dos dedos através de alterações na tensão de saída de um conjunto de extensómetros, esta mede a proximidade dos dedos em relação a sua posição de origem.



**Figura 2.14: Orientação dos eixos relativamente aos dedos (retirado de [27])**

Quanto aos acelerómetros propriamente ditos, estes são constituídos por uma massa suspensa por uma mola. Depois existem sensores capacitivos distribuídos ao longo dos eixos X e Y (segundo a orientação apresentada na Figura 2.14) de forma a conseguir retirar uma medição proporcional ao deslocamento da massa relativamente ao seu ponto de descanso. Com isto, medindo o deslocamento da massa (seja por aceleração ou por desvio em relação ao vector gravitacional), é possível determinar a sua posição angular absoluta. Com isto, através dos dados do eixo do Y consegue-se determinar a flexão de cada dedo, e com o eixo do X a abertura dos dedos, a rotação e direcção horizontal do pulso.

### 2.3.3. Soluções inside-in

Como foi referido em 2.1, estas soluções funcionam a base de localização relativa de transmissores posicionados em pontos chave ao longo das articulações do corpo. Nesta próxima secção serão abordados projectos que façam uso desta abordagem.

#### *Music Glove*

A Music Glove é um sistema que apesar de não incidir muito com MIDI, incide na captura de gestos da mão de forma a converter esses mesmos gestos numa forma de expressão musical. Tal como este tema, também tem por

objectivo a facilitação na terapia para recuperação de capacidades motoras em pacientes de doenças como AVC.

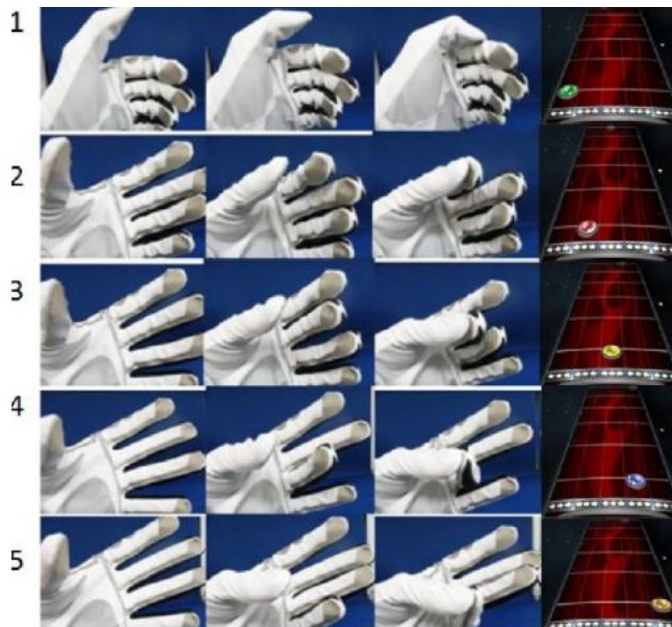


Figura 2.15: Cinco movimentos definidos (retirado de [28]).

Trata-se de um sistema audio-visual que junta uma luva que regista movimentos simples da mão e regista esses movimentos como eventos, enviando o sinal para um controlador (neste caso o computador) para este ser processado por um software. Foram definidos cinco movimentos para gerar cinco notas diferentes apresentados na Figura 2.15. Para a experiência, os autores utilizaram o jogo *open source*, Frets on Fire [28].

Para efectivamente, detectar estes movimentos, a luva contém seis terminais eléctricos, cinco em cada dedo e um na parte lateral do indicador. Ao tocar como polegar em cada um dos dedos, é gerada uma corrente que é reconhecida pelo controlador como um evento.

Concluindo, este projecto apresenta algumas ideias em como definir mensagens de eventos. Enquanto não aparenta implementar nenhuma interface MIDI, a sua implementação é idêntica a de mensagens NOTE\_ON e NOTE\_OFF em MIDI. Visto que estes movimentos definidos mostraram ter um efeito muito significativo na recuperação de capacidades motoras de pacientes de AVC,

seriam fortes alternativas a gestos a manter na implementação desta dissertação, por exemplo, para mensagens MIDI de controlo discreto.

### *Cyber Composer*

O Cyber Composer é um dispositivo que já tenta automatizar a geração de som com base em gestos de ambas as mãos. Para facilitar na implementação, é usada já uma luva que regista toda a actividade ao nível da mão, chamada de Cyber Glove [29]. A Cyber Glove contém 18 extensómetros embutidos no tecido da luva em pontos chave da mão, de forma a conseguir adquirir toda a informação necessária para conseguir descrever a postura da mão. Estes extensómetros são essencialmente tiras flexíveis resistivas cuja resistência varia consoante o ângulo que esta faz. Estes estão posicionados pela mão da seguinte forma:

- Dois no polegar, sendo que um serve para medir a rotação do polegar (articulação inferior que leva o polegar ao dedo mindinho) e outro para a articulação superior onde este efectivamente dobra;
- Dois em cada um dos restantes dedos, sendo que um mede o angulo da articulação superior onde o dedo dobra, e outro para medir o angulo ao nível da junção do dedo com a mão;
- Um sensor entre cada dedo que mede a abertura entre cada dedo em separado;
- Dois sensores ao nível do pulso para medir a posição da palma em relação ao pulso;
- Um sensor que mede a rotação do dedo mindinho através da palma da mão tal como é feito com o polegar.

Na Figura 2.16 pode-se observar um exemplar da Cyber Glove, onde podem ser observadas as costuras onde estão embutidos os extensómetros.



Figura 2.16: Cyber Glove usada no projecto Cyber Composer (retirado de [30])

Adicionalmente, visto que a Cyber Glove não suporta a determinação da orientação 3D absoluta da mão por si só, foram adicionados um sensor FASTRAK 3D da Polhemus. Este sensor possibilita a obtenção de dados sobre a posição e orientação absoluta das mãos do utilizador [30][31]. Esta adição é interessante a ter em conta, pois acaba por corrigir uma desvantagem crítica que esta abordagem teria a partida, que é a obtenção de localização e orientação absoluta, que tal como foi referido em 2.2, era uma desvantagem crítica da abordagem *inside-in*.

Após a captação da informação dos movimentos e gestos das mãos, estes ainda devem ser digitalizados. Dito isto, estes movimentos são então convertidos em mensagens MIDI pela sua facilidade de operação e compatibilidade. Dito isto, o Cyber Composer é constituído por vários módulos, todos ligados a um controlador que processa toda a informação de cada um:

- Interface musical;
- Interface de Cyber Glove;
- Módulo de geração musical;
- Módulo de geração de melodia;

Na interface musical é onde são convertidos os "sinais de expressão" da mão para MIDI de forma a serem enviadas para um módulo MIDI como um módulo

Roland TD-11. Depois usou-se a Windows MIDI API para controlar input e output de MIDI do lado do computador. O módulo de interface CyberGlove envia os sinais de posição e orientação da mão (incluindo os dedos) para o controlador para estes serem convertidos em "sinais de expressão" pelos módulos de geração de música de fundo e de geração de melodia. O módulo de geração musical gera música de fundo conforme dois parâmetros definidos pelo utilizador, nomeadamente tempo e escala. Por fim, o módulo de geração de melodia, tal como o nome indica, gera melodias tendo por base os movimentos da mão [30].

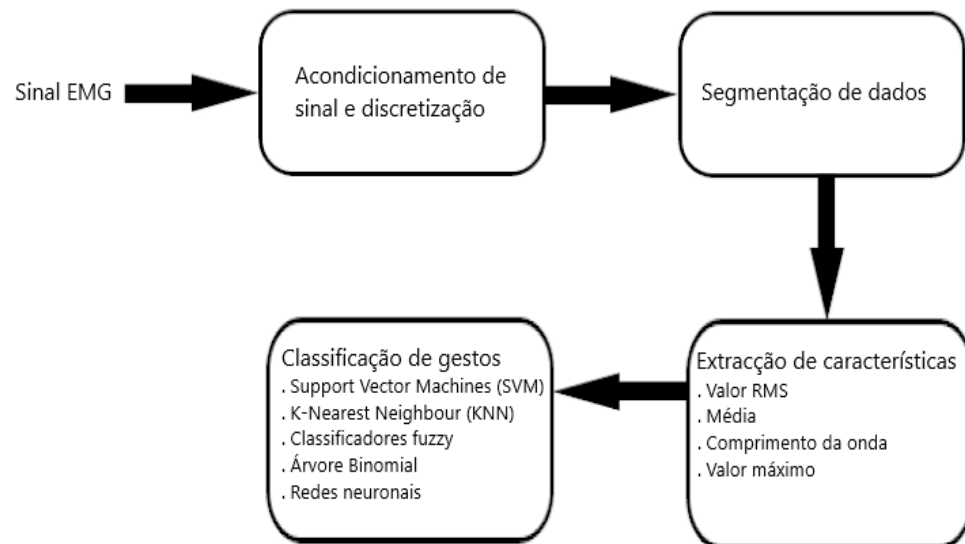
Quanto aos modos de operação, a ideia geral é uma geração e manipulação de parâmetros MIDI com base nos movimentos da mão. No entanto, a geração de áudio já é feita de forma automática com base em algoritmos e teoria musical. Visto que esta parte já não é relevante para o âmbito desta dissertação, para mais detalhes, é recomendada a leitura de [30].

#### 2.3.4. Soluções baseadas em sinais biométricos EMG

Sistemas de controlo EMG podem dividir-se em duas categorias, nomeadamente os sistemas com reconhecimento de padrões e sistemas sem reconhecimento de padrões. Por um lado, os sistemas que não reconhecem padrões servem para aplicações em que se pretende apenas reconhecer um valor em módulo na entrada seja em grandeza for. Por outro lado, sistemas de controlo por sensores EMG por reconhecimento de padrões já envolvem aplicações onde se pretende mesmo reconhecer a postura de uma determinada parte do corpo, como por exemplo, a postura da mão. Nesta secção serão estes sistemas que serão de maior interesse [32].

O reconhecimento de gestos da mão por sistemas EMG é dividido por várias fases. Em primeiro, vem a aquisição de dados, onde o sinal deve ser amplificado e filtrado do seu ruído. Em seguida, é necessário segmentar o sinal para apanhar apenas o sinal correspondente ao movimento efectuado, de forma a extrair as suas características. Para sistemas de tempo real, é importante manter esta janela de segmentação abaixo de 100ms de forma a não causar uma sensação de atraso por parte do utilizador [33]. Estas podem ser quaisquer características que descrevam a onda como o valor médio ou comprimento do transiente, em que

logicamente, a sua escolha estará sempre dependente da aplicação para a qual o sistema será adaptado. Finalmente, após a extracção de características vem a classificação de gestos conforme as características extraídas na fase anterior, resultando no diagrama de blocos apresentado na Figura 2.17 [32].



**Figura 2.17: Diferentes fases de reconhecimento gestual em sistemas de controlo EMG**

Devido a algumas limitações já próprias dos sistemas EMG [34], torna-se necessário em alguns casos incluir uma fase de correcção de erros de classificação [35]. Existem vários algoritmos que se podem seguir como:

- Maioria de voto [36]–[38];
- Velocidade móvel [39];
- Decisão baseada em velocidade de rampa [40]

O conceito de reconhecimento de padrões é algo que teve os seus primeiros passos nos anos 60 [39]. Infelizmente, a capacidade de processamento disponível na altura ainda não permitia aplicações em tempo real sem que houvesse grandes quantidades de hardware envolvidos. Dito isto, segundo [39], a primeira implementação de reconhecimento de padrões de sinais EMG em tempo real com alta performance e precisão veio em 1993 por Hudgins *et al.* [33]. Esta abordagem segue essencialmente o mesmo processo que foi anteriormente explicado. O sinal

EMG foi amostrado a 1kHz com uma janela de segmentação definida como sendo igual ou inferior a de 100ms. Para disparar esta janela, é monitorizado o valor médio absoluto, e quando este ultrapassa um certo valor, são analisadas as 200 amostras seguintes (200ms). Foram excluídas características do domínio da frequência pois essas exigem demasiado processamento. Dito isto, foram escolhidas as seguintes características:

- Valor médio absoluto;
- Declive do valor médio absoluto;
- Cruzamentos com a origem;
- Mudanças de sinal no declive;
- Comprimento de onda.

Finalmente, para a fase de classificação, foi implementada uma rede neuronal artificial de duas camadas, sendo que para a fase de treino da rede neuronal, foram utilizados dez conjuntos de dados.

#### *Sistemas modernos*

Após esta abordagem de Hudgins *et al.*, foram se observando muitas novas abordagens com diferentes algoritmos de *machine learning*, diferentes posições para os sensores, entre outros parâmetros variáveis. H. Wu *et al.* [41] apresenta um sistema baseado em Arduino, com uma amostragem a 500Hz em ambos os sensores que integram o sistema, e utilizando três algoritmos de classificação, nomeadamente controlo directo, controlo abstracto e análise linear discriminante. D. Espinoza *et al.* [42] faz um estudo interessante num sistema EMG de oito gestos, em que são comparadas várias variações de dois algoritmos de classificação, nomeadamente *support vector machines* (SVM) e árvores de decisão (resultados apresentados na Tabela 2.2). Adicionalmente, propõe o uso de uma ferramenta de análise de conjuntos de dados para algoritmos de classificação *Classification Learner*. Este estudo é particularmente interessante pois faz o uso de equipamento já escolhido numa primeira abordagem no plano inicial desta dissertação, nomeadamente o uso de um controlador Raspberry Pi 3, validando o seu uso para a implementação desta dissertação.

Tabela 2.2: Desempenho de diferentes algoritmos testados (adaptado de [42])

Algoritmo	Desempenho (%)
Árvore grande	49.2%
Árvore média	96.4%
Árvore pequena	99.1%
SVM Linear	99.8%
SVM Quadrática	99.8%
SVM Cúbica	99.9%
SVM Gaussiana grande	99.5%
SVM Gaussiana média	99.7%
SVM Gaussiana pequena	99.1%

Como foi referido, todos estes estudos seguem a mesma ordem de ideias. Acondicionamento de sinal, extração de características e classificação, havendo alguns casos que ainda implementem algum pós-processamento para correcção de erros [32]. Para simplificar, existem também vários estudos a fazerem um apanhado de vários sistemas EMG de reconhecimento de padrões [32], [43], [44]. Destes todos referidos, sugere-se que o leitor consulte [43], onde é feito um apanhado de mais de 30 estudos diferentes até ao não de 2020, bem como uma análise de relação de características do sistema para as suas respectivas necessidades e aplicações.

#### *Detecção sinais EMG no pulso ao invés do antebraço*

Algo importante neste tipo de aplicações é a rejeição de gestos não contidos no conjunto de gestos detectáveis. Ou seja, um gesto não intencionado ser detectado como um gesto dentro do conjunto detectável. Normalmente definem-se um conjunto, e o sistema será projectado de forma a detectar gestos dentro deste conjunto. Infelizmente, a detecção de movimentos não intencionados pode, em algumas aplicações, representar um risco de segurança para o utilizador. Para esta secção a aplicação de foco serão exoesqueletos com o objectivo de fornecer maior força e estabilidade ao seu utilizador. Existem vários exemplos de aplicações de exoesqueletos [45]–[52], sendo que nem todos utilizam a mesma forma de implementação, ou seja sensores EMG. Dito isto, estas luvas exoesqueleto, quando usadas num contexto de força como manuseamento de ferramentas de potência ou levantamento de grandes cargas, devem garantir a

maior robustez relativamente a estabilidade de interpretação de comandos do utilizador [53]. Dito isto, nesta secção será focado um estudo que aborda exactamente este tema especificamente em força de aperto ao nível da mão.

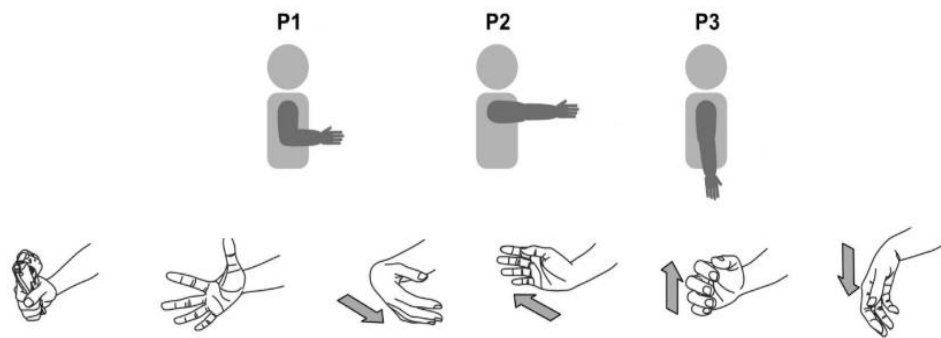
Um problema ao interpretar sinais EMG, é que alguns músculos encontram-se sobrepostos uns com os outros. Por consequente, geram-se muitos problemas interferência de sinais causados pela interacção sinérgica entre diferentes músculos que interagem uns com os outros para um determinado movimento. Adicionalmente, diferenças fisiológicas entre pessoas acabam por fazer diferenciar os locais óptimos de medição. Isto cria a necessidade de testar de pessoa para pessoa de forma a encontrar os pontos ideais. [53]

Em resposta a estes problemas, Cheon *et al.* [53] desenvolveram um sistema que faz as suas medições com apenas um sensor na zona das junções musculotendinosas do flexor superficial dos dedos (mesmo antes do pulso). Desta forma, possibilitando medições estáveis, tendo por base a identificação de "intenções de pega".



Figura 2.18: Luva exoesqueleto de um só sensor (adaptado de [53])

A Figura 2.18 apresenta a luva desenvolvida, onde se pode observar que existe apenas um sensor perto do pulso. Esta escolha é interessante neste contexto precisamente pela aplicação para a qual esta foi desenvolvida. A zona do pulso é conhecida por não ter uma qualidade de sinal muito boa. No entanto, é a zona onde se detecta uma grande amplitude do sinal EMG quando se dá um aperto ao nível da mão. Concluindo, de forma a concluir sobre validade desta propriedade, Cheon *et al.* comparam medições obtidas com um sistema de seis sensores ao nível do antebraço. Para isto, fez-se um estudo dividido em três fases, nomeadamente recolha de dados, treino de classificadores, e avaliação.



**Figura 2.19: Seis gestos diferenciados e as três posições de braço (adaptado de [53])**

Para a recolha de dados foram feitos testes em oito pessoas em que foram aplicados a cada um ambos os sistemas (um sensor no pulso e seis sensores no antebraço). Cada participante executou os seis movimentos apresentados na Figura 2.19. Foi usado um peso de mão para executar o gesto de aperto. Nas outras posições os participantes soltaram o pulso o máximo possível para isolar esses gestos do efectivo gesto de aperto, de forma a não interferir com os dados.

Os sinais EMG foram amostrados a frequência de 1000Hz, filtrados por um filtro de Butterworth de 4ª ordem entre 20 e 450Hz de forma a remover artefactos e ruído nas altas frequências. Para identificar um gesto como intencionado, foi definido que este começaria quando o sinal EMG estivesse acima de duas vezes

o desvio de sinal de posição de repouso. Desse sinal foram retiradas as seguintes características numa janela de 250ms e um step time de 50ms:

- *Mean absolute values* (MAV);
- *Zero crossings* (ZC);
- *Waveform lengths* (WL);
- *Slope sign changes* (SSC).

Recolheram-se vários conjuntos de dados para três posições de braço diferentes nomeadas de P1, P2 e P3:

- P1 - Quatro conjuntos de treino e dois de teste
- P2 - Dois conjuntos de teste
- P3 - Dois conjuntos de teste

Para a fase treino de classificadores, definiram-se dois classificadores de SVM. Enquanto um foi treinado com o apenas os dados do sensor próximo do pulso, o outro foi treinado também com os dados dos sensores do antebraço. Adicionalmente, adicionou-se um terceiro classificador. Para treinar os classificadores, utilizou-se o Scikitlearn, que é uma biblioteca de Python para *machine learning* e análise de dados. Mais especificamente utilizou-se o comando `svm.SVC` para fazer a classificação de intenção de apertar e não apertar. Finalmente, foram usados os restantes conjuntos de dados como conjuntos de teste para determinar a performance de cada classificador. Para avaliar a performance foram escolhidos os seguintes factores segundo:

- Sensibilidade - Definido por um *true positive rate for grasping* (TPRG), que determina a frequência que o classificador determina acertadamente a intenção de apertar.
- Especificidade - Definido por um *true negative rate for grasping* (TNRG), que determina o quão bem os outros gestos são rejeitados.

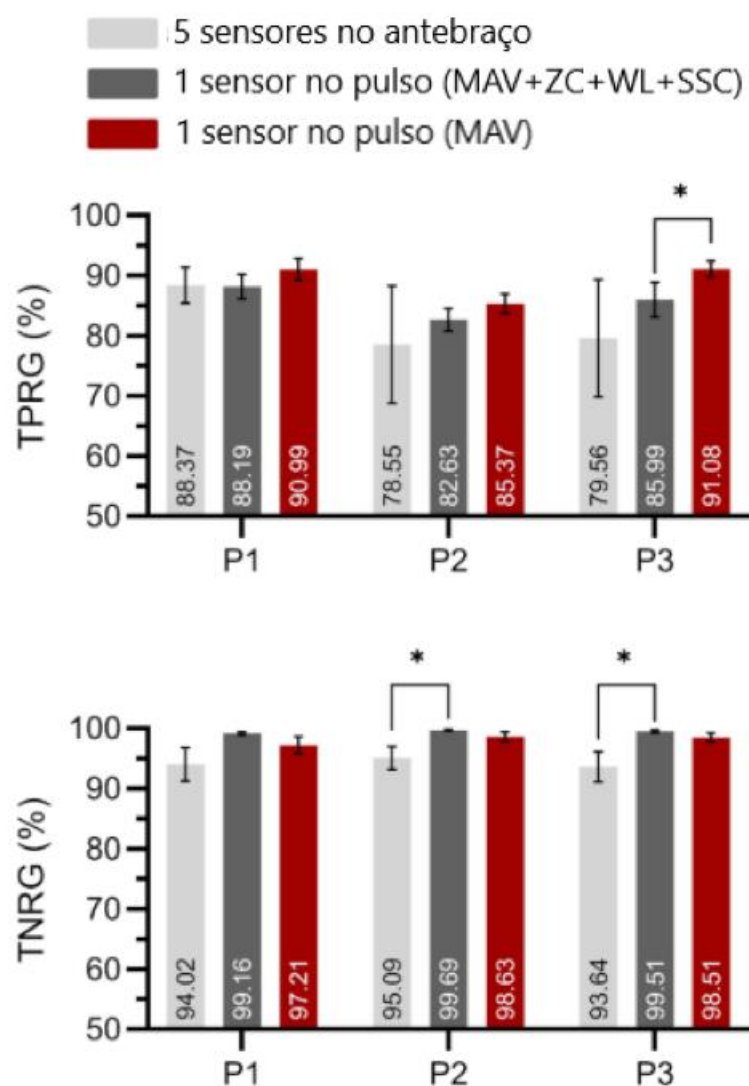


Figura 2.20: Resultados da fase de avaliação (adaptado de [53])

Após a fase de testes, obtiveram-se os resultados apresentados na Figura 2.20. Os resultados demonstraram que não existe uma diferença muito significativa em sensibilidade entre o pulso e o antebraço. Por outro lado, observou-se uma maior especificidade para detectar gestos de aperto utilizando apenas um sensor no pulso quando comparado com cinco sensores no antebraço.

Com isto, concluiu-se que, para detectar o acto de fazer força com a mão ou aperto, apenas um sensor EMG posicionado no pulso tem um melhor desempenho que seis sensores EMG no antebraço.



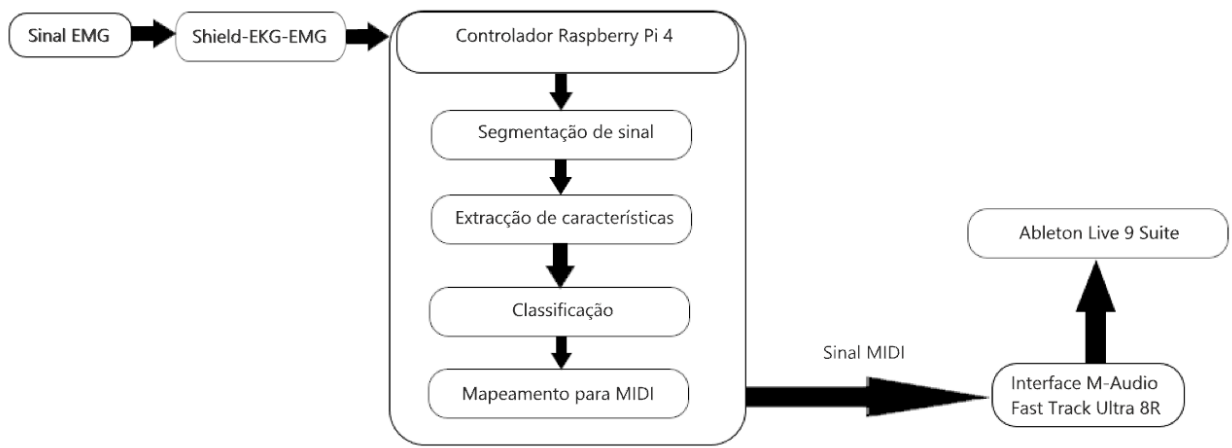
## 3. Proposta

Nesta secção são apresentadas várias abordagens ao problema que é a captação de movimentos da mão. Apesar de um foco desta dissertação ser o uso de sensores EMG, decidiu-se incluir ainda outras soluções que tomassem uso de outros sensores que não EMG. Com isto, podem-se concluir alguns aspectos desta dissertação.

Como foi visto em [42], o uso de SVM como método de classificação é bastante aceitável relativamente ao reconhecimento de postura da mão para oito posições diferentes. Adicionalmente, com este estudo é comprovada a viabilidade de um controlador Raspberry Pi para este tipo de aplicação. Foi também vista uma ferramenta de Matlab, *Classification Learner*, para estudar a viabilidade de determinados métodos de classificação para determinados conjuntos de treino e de teste. Isto facilita bastante a tarefa de ter de avaliar diferentes estudos de forma a encontrar aquele que tiver melhor desempenho. Finalmente, em [53] foi discutido o posicionamento ideal dos sensores, bem como uma comparação de precisão conforme o que é pretendido como gestos a detectar. Este estudo é no entanto, discutível visto que existem outros estudos que acrescentam que o pulso consegue ser ainda mais preciso até em aplicações que necessitem de detecção da postura da mão (sem contar com a intenção de aperto) [54]. Com isto concluem-se vários aspectos definitivos em relação á implementação desta dissertação.

Serão utilizados sensores EMG da Olimex para a captação de sinais EMG, bem como a placa Shield-EKG-EMG apresentada na Figura 2.10 para acondicionamento de sinal e discretização. A fase de processamento de sinal, ou seja, segmentação de transientes, extracção de características e classificação de gestos será feita com um controlador Raspberry Pi 4. Tendo em conta a sua linguagem de programação, este controlador é ideal para este tipo de aplicação tendo em conta as suas bibliotecas já existentes para implementações de algoritmos de análise de dados e *machine learning*. Relativamente às características a extrair, numa primeira abordagem, serão utilizadas as mesmas

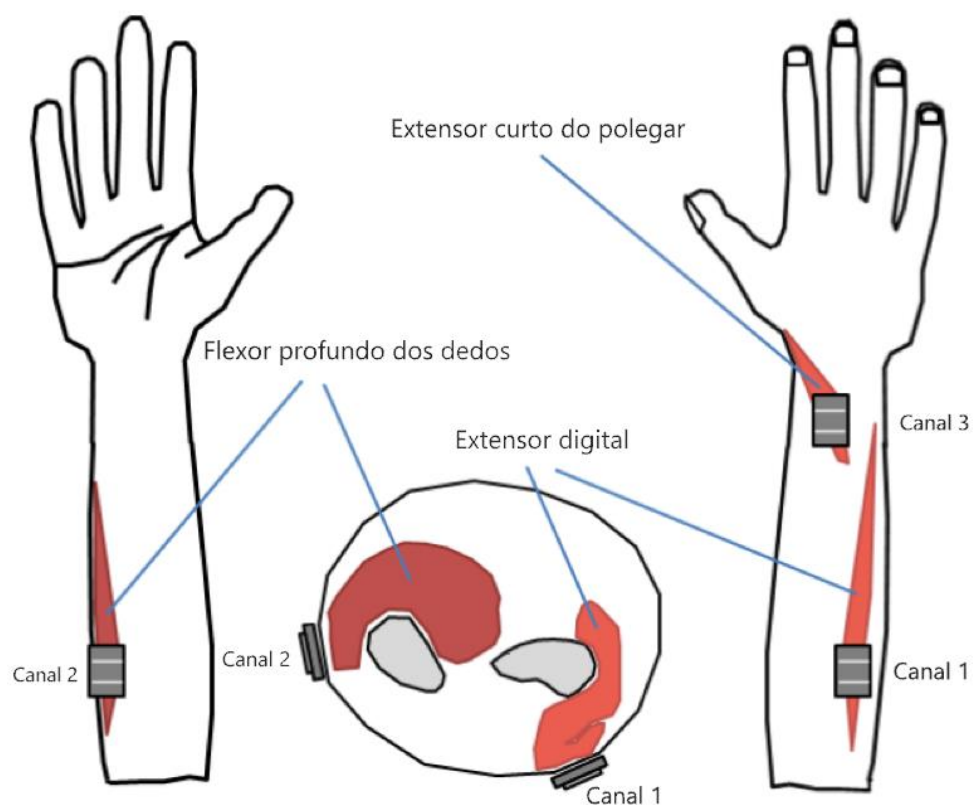
que em [42], com a forte possibilidade de experiência para obtenção de possíveis melhores resultados. Tendo em conta os resultados obtidos na sua performance em [42], não será implementado, numa primeira abordagem, nenhum algoritmo de pós-processamento de forma a corrigir classificações erradas. Finalmente, dos gestos seleccionados, cada um destes será mapeado para uma mensagem MIDI NOTE\_ON, ou seja, uma nota musical. Disto tudo, resulta então o diagrama de blocos apresentado na Figura 3.1.



**Figura 3.1: Diagrama de blocos da montagem final proposta**

### 3.1. Posição dos sensores

Serão aplicados três canais ao nível do antebraço direito, o que implica dois pólos por canal nos pontos de medição do braço direito, e três referências (uma por cada canal) no pulso esquerdo. Com este efeito, serão posicionados apanhando três pontos distintos do antebraço como é apresentado na Figura 3.2.



**Figura 3.2:** Esquema da posição dos sensores no braço direito de forma a capturar actividade de três músculos distintos do braço (adaptado de [55])

### 3.2. Leitura de sinal

Devido a questões de compatibilidade, segundo a documentação dos sensores [9], será necessário fazer toda a leitura de sinal EMG através de um controlador Arduino UNO que estará a enviar cada amostra para o controlador Raspberry PI 4.

Para enviar a informação do Arduino para o Raspberry PI serão consideradas duas opções, nomeadamente UART e *Serial Peripheral Interface* (SPI). Sendo que em Arduino, o protocolo SPI permite maior controlo sobre as mensagens que se transmitem, e tendo em conta que não existe limitação a conjuntos de 8 caracteres, decidiu-se que se ia usar SPI, com uma resolução de 8 bits por amostra, devido a necessidade de ler 3 valores por cada leitura dos sensores.

### 3.3. Acondicionamento de sinal e detecção de actividade gestual

Para acondicionar o sinal recebido é passado o sinal para uma janela de leitura de 50 amostras e onde o sinal é passado por um filtro de média com janela de 5 amostras de forma a reduzir ruídos. Para a detecção de gestos, são extraídas cinco características, nomeadamente:

- Valor Root Mean Square (RMS)
- Módulo do valor de pico
- Módulo do desvio médio
- Módulo do valor médio
- Cruzamentos com a origem

Para a análise de gestos definiu-se uma janela de 100 amostras para análise após a janela de detecção. Tal como na detecção, são extraídas as mesmas 5 características referidas na detecção e alimentadas ao classificador.

### 3.4. Treino de classificadores

Para definir implementar um classificador serão necessários um conjunto de treino e um conjunto de teste. Foram definidos quatro gestos sendo que, para cada um, registaram-se 60 medições diferentes. Com estas medições irão treinar-se três classificadores diferentes:

- Random Forest (RF)
- Support vector machines (SVM)
- Rede neuronal (RN)

Após treinar cada um destes classificadores, irá definir-se o mais adequado com base nos resultados obtidos. Tendo em conta bibliografia revista, é esperado um melhor desempenho do classificador SVM .

### 3.5. Conversão de gestos para mensagens MIDI

Existem várias bibliotecas para Arduino capazes de gerir envio e receção de mensagens MIDI. No entanto, implementar a gestão de MIDI do lado do Arduino implicaria mais um canal de comunicação entre o Arduino e o Raspberry PI, o que implicaria mais latência. Dito isto, todo o processamento MIDI será feito do

lado do Raspberry PI. Felizmente, existem também bibliotecas de Python para MIDI como o py-midi, que será usada para enviar mensagens MIDI em tempo real, por cabo MIDI. Na eventualidade de haver limitações a nível de hardware, ao invés de se usarem saídas MIDI, as mensagens poderão também ser enviadas internamente e lidas directamente pelo software de produção musical, instalado no Raspberry PI.



## 4. Implementação

Após a definição dos requisitos, implementou-se o projecto. Nesta secção será apresentada em maior detalhe a implementação de cada uma das fases referidas anteriormente, desde a captação do sinal EMG até a performance musical.

### 4.1. Arquitectura

#### 4.1.1. Hardware

A cadeia é iniciada pelos sensores, cuja alimentação e leitura é feita pelo Arduino. A comunicação entre o Arduino e o Raspberry PI é garantida através de protocolo SPI com uma frequência máxima de *clock* de 40kHz. Dito isto, o Arduino está ligado ao computador por USB, e ligado ao Raspberry PI através das respectivas saídas e entradas *Master In Slave Out* (MISO), *Master Out Slave In* (MOSI), e *Serial Clock* (SCLK) tal como está apresentado na Figura 4.1.

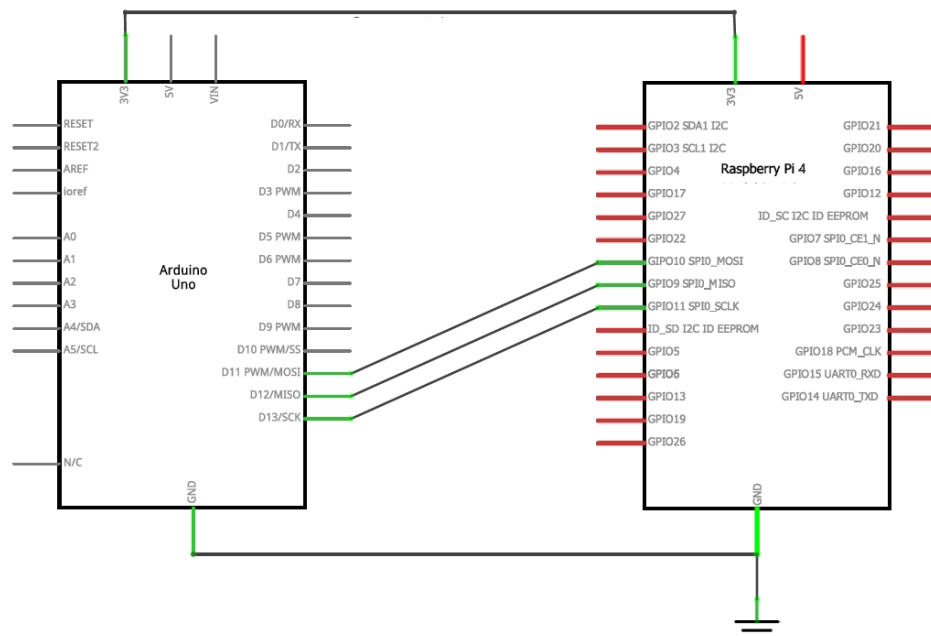


Figura 4.1: Ligação entre Arduino UNO e Raspberry PI

Enquanto se previa que se iria enviar mensagens MIDI internamente no Raspberry PI, devido a questões de compatibilidade com o sistema operativo, tal não foi possível. Em alternativa, estabeleceu-se que o Arduino UNO irá enviar mensagens MIDI para o computador através de UART. Concluindo, o circuito final é apresentado na Figura 4.2.

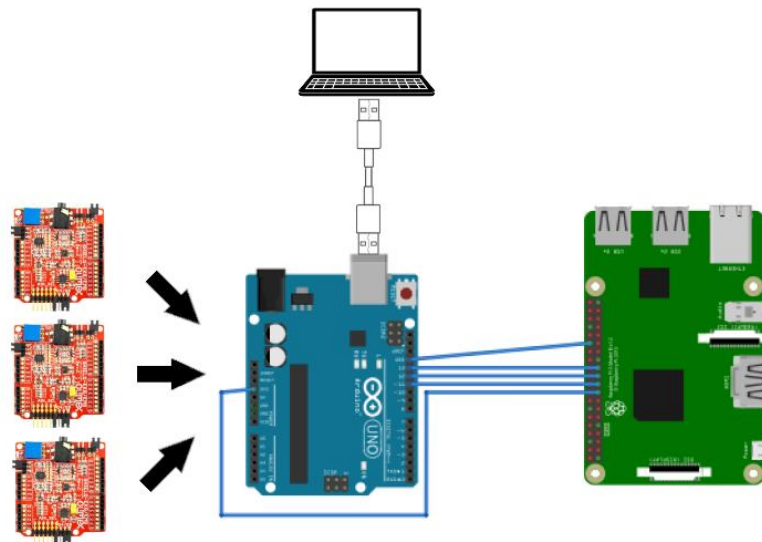


Figura 4.2: Arquitectura do circuito final

#### 4.1.2. Software

De um ponto de vista funcional, o Arduino e o Raspberry PI têm funções bastante separadas. Enquanto o Arduino está encarregue de leitura dos sensores e do envio de mensagens MIDI para fora, o Raspberry PI está encarregue de todo o processamento de sinal desde a limpeza do sinal até a classificação dos gestos. Dito isto a comunicação segue da seguinte forma:

- Raspberry PI faz um pedido de leitura ao Arduino por SPI
- Arduino envia, por SPI, uma leitura dos três sensores para o Raspberry PI
- Acondicionamento de sinal
- Extração de características
- Detecção de gestos
- Classificação de gestos
- Raspberry PI envia um sinal, por SPI, com o código do gesto detectado para o Arduino
- Arduino envia, por UART, a mensagem correspondente ao gesto executado

Esta lógica, é então apresentada de uma forma geral na Figura 4.3.

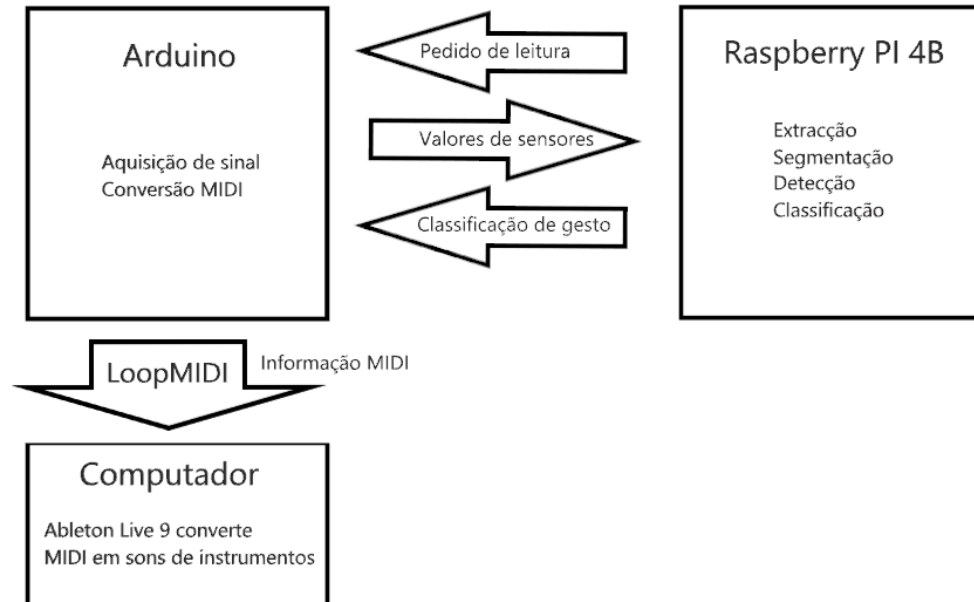


Figura 4.3: Esquema funcional da lógica do controlador (adaptado de [56])

O software encontra-se separado em dois módulos, nomeadamente `emg_slave.ino`, integrado no Arduino UNO, e `emg_read.py` integrado no Raspberry PI.

O módulo `EMG_Slave.ino` é um módulo que está integrado no Arduino UNO e está encarregue de garantir a leitura dos sensores, envio das leituras para o Raspberry PI, e envio de sinais MIDI para o computador. Para maximizar a rapidez de leitura ao nível do Arduino e garantir sincronismo entre os dois componentes, em vez de recorrer ao ciclo convencional, implementou-se a leitura por *interrupts*, que é disparada apenas por pedido do Raspberry PI. De forma, a garantir que o Arduino não envia informação mais lentamente que esta é pedida, não se implementou nenhum processamento ao nível do Arduino. O módulo `emg_read.py` está integrado no Raspberry PI e é responsável por obter as leituras do Arduino, detectar e classificar gestos, e envio dos resultados para o Arduino. Por fim, a informação MIDI é enviada por UART para um porto virtual configurado por um software a parte chamado LoopMIDI.

Tendo em conta a lógica referida anteriormente, é necessário, de forma aos dados dos sensores EMG cheguem ao Raspberry PI, estes dois componentes comunicarem um com o outro. Esta comunicação é garantida através de protocolo SPI, de forma a minimizar e estabilizar o tamanho dos pacotes a transmitir, garantindo uma frequência de amostragem estável. Com isto, o software segue a lógica apresentada no esquema apresentado na Figura 4.4.

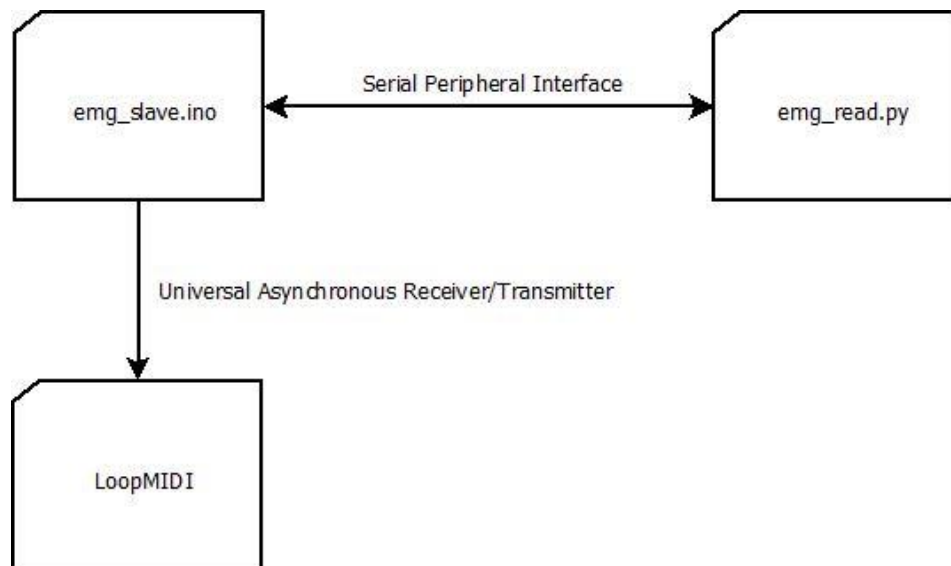


Figura 4.4: Esquema de comunicação entre componentes

## 4.2. Captação EMG

Três pontos importantes foram necessários definir para a captação do nosso sinal EMG. Em primeiro lugar, foi necessário escolher os pontos mais afinados para a aplicação pretendida. Por fim, de forma a afectar o mínimo possível a frequência de amostragem, determinou-se também uma forma eficiente de transmitir os dados para o Raspberry PI.

### 4.2.1. Leitura dos sensores no Arduino

Para a captação do sinal EMG, determinou-se que se iria utilizar três pontos de medição ao nível do antebraço, de forma a captar três músculos distintos, nomeadamente:

- Extensor digital
- Flexor profundo dos dedos
- Extensor curto do polegar

A medição destes três pontos é necessária, pois dentro dos três movimentos escolhidos, deve ser garantida a activação de pelo menos um destes três pontos por movimento. Como já foi referido em 3.2, serão utilizados sensores Shield EKG-EMG da Olimex. Cada sensor envolve dois pólos e um neutro. Cada conjunto de pólos será colocado directamente sobre a zona com mais fibra muscular de cada músculo do braço direito, e cada neutro no pulso esquerdo. Dito isto, respeitando os posicionamentos estabelecidos na Figura 3.2, posicionaram-se os sensores como apresentado na Figura 4.5.



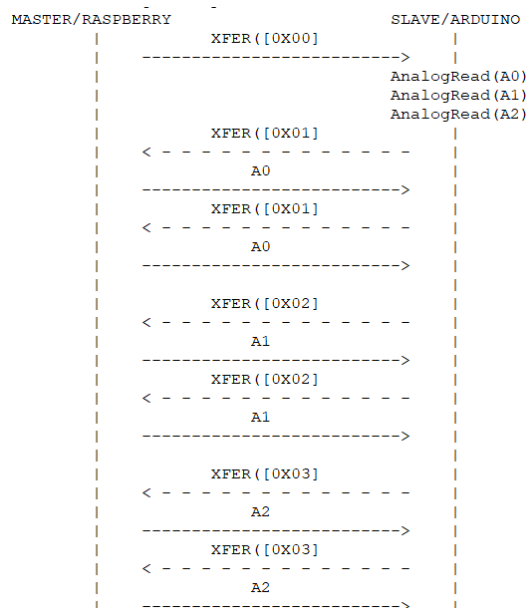
**Figura 4.5: Posição dos sensores no braço direito de forma a capturar actividade de três músculos distintos do braço (adaptado de [56])**

Com isto, o uso destes sensores exige que a sua placa seja alimentada por uma placa Arduino. Como tal, os sensores Shield EKG-EMG foram alimentados em série por um Arduino UNO, sendo a sua leitura garantida pelo mesmo, seguindo o código apresentado em anexo. Disto resulta uma amostragem inicial sensivelmente 10kHz. Infelizmente, devida a transmissão de dados entre Arduino e o Raspberry PI, é esperado que esta diminua significativamente.

#### 4.2.2. Transmissão para o Raspberry PI

Como o Arduino não possui capacidade de processamento para lidar com a detecção e classificação de gestos, é necessário fazer esse processamento ao nível da placa Raspberry PI. Como tal, é necessário encontrar uma forma eficiente e robusta de transmitir a informação lida pelos sensores. Com isto, determinaram-se duas opções, nomeadamente UART e SPI.

Por um lado, transmissão UART apresentou-se como uma forma simples de implementar, garantindo uma leitura simultânea de todos os sensores. Infelizmente, este método tinha uma largura de banda muito maior do que necessário para esta aplicação. Isto era causado devido ao facto de serem enviados *strings* inteiras com os valores. Tendo em conta que cada caracter representa um byte, além de cada pacote de informação ser muito maior do que necessário, dependendo dos valores, estas *strings* enviadas por UART tinham valores variáveis, causando uma frequência de amostragem instável ao nível do Raspberry PI. Por consequente, mostrou ser causa de latências muito significativas, afectando negativamente a frequência de amostragem ao nível do Raspberry PI. Felizmente, a transmissão por SPI revelou-se ser exactamente o que se estava à procura. A sua forma mais cirúrgica de lidar com informação facilitou na redução e estabilização do tamanho dos pacotes enviados para o Raspberry PI, passando a ser necessário apenas um byte por valor transmitido.



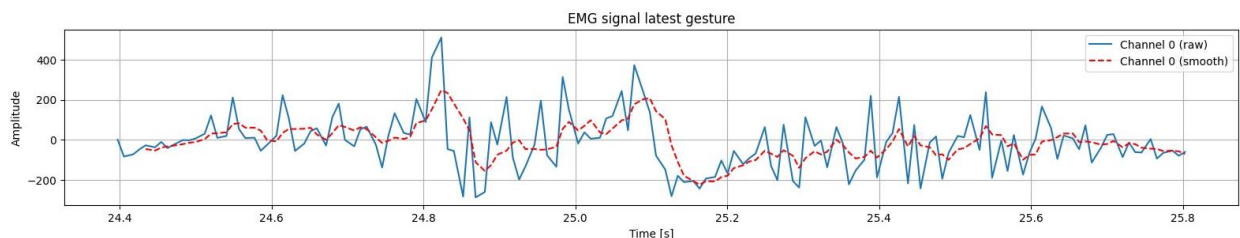
**Figura 4.6: Protocolo de comunicação para um ciclo de leitura dos três valores EMG (adaptado de [56])**

Com isto, estabeleceu-se um protocolo a seguir para uma devida leitura dos sensores, apresentado na Figura 4.6. Cada ciclo é iniciado com o envio de um byte a 0 para o Arduino. Após receber um byte nulo, o Arduino segue para uma

leitura de todos os registos, guardando-os em cache. Em seguida, o Raspberry PI irá enviar mensagens com os números dos sensores que pretende ler, indo de 1 a 3, recebendo por sua vez um byte por cada sensor.

#### 4.3. Processamento de sinal

De forma a ler devidamente o sinal EMG sem interferências, é necessário limpar de componentes indesejadas. Um problema muito comum com sinais EMG, é que costumam vir acompanhados de interferência ECG (batimento cardíaco). Isto representa um problema para a detecção de gestos na mão, visto que a actividade ECG poderá ser interpretada como um gesto. Felizmente, a gama de frequências para o sinal ECG está fora da gama utilizável EMG. Enquanto a banda utilizável da componente ECG vai apenas até aos 150Hz, a gama de frequências do sinal EMG, embora não dominante acima dos 150Hz, consegue ir até aos 500Hz [57]-[58]. Com isto, resolveu-se implementar um filtro passa-alto de 3ª ordem com frequência de corte nos 150Hz como primeira abordagem, cortando o máximo de componente ECG possível, afectando o mínimo possível o sinal EMG.



**Figura 4.7: Sinais EMG antes (azul) e depois (vermelho) do filtro de média**

Por fim, apresentado na Figura 4.7, aplicou-se também um filtro de média de 3 amostras para estabilizar o sinal sem perder picos mais essenciais.

#### 4.4. Extracção de características e detecção

Para descrever gestos, e até mesmo detectá-los, é necessário saber o que os descreve. Foi então definido um conjunto de 5 atributos de forma a ser possível descrever o sinal em tempo real:

- *Valor RMS* =  $\sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}$
- *Módulo do valor de pico*
- *Módulo do desvio médio* =  $\frac{\sum_{i=1}^N |x_i - \bar{x}|}{N}$
- *Módulo do valor médio* =  $\frac{\sum_{i=1}^N |x_i|}{N}$
- *Número de cruzamentos com a origem*

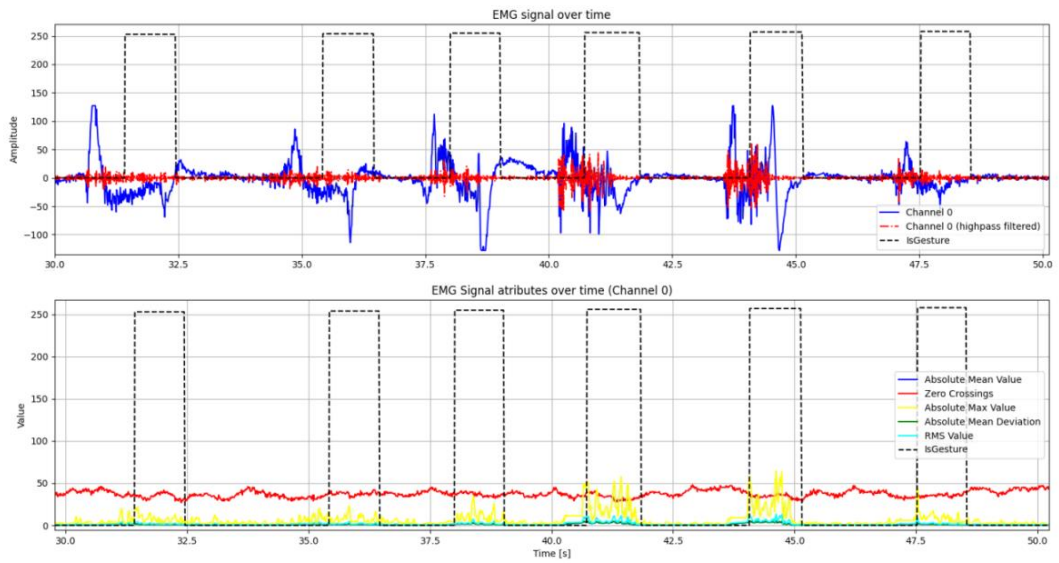
Estes valores são sempre calculados com base numa janela de 50 amostras na qual estes atributos vão ser conforme o seu valor de detecção estipulado.

$$x_{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}$$

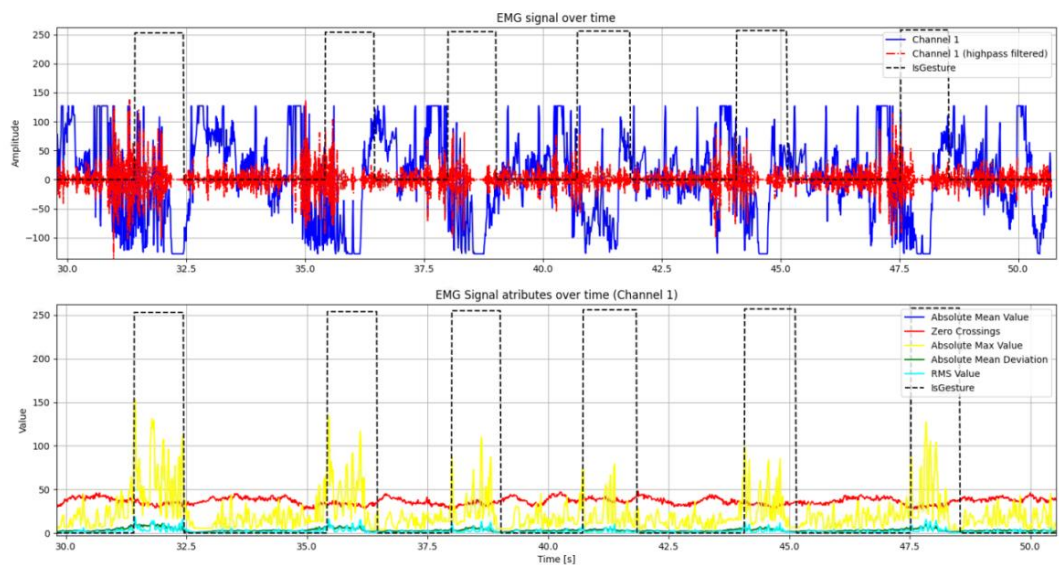
$$DM = \frac{\sum_{i=1}^N |x_i - \bar{x}|}{N}$$

$$\bar{x} = \frac{\sum_{i=1}^N |x_i|}{N}$$

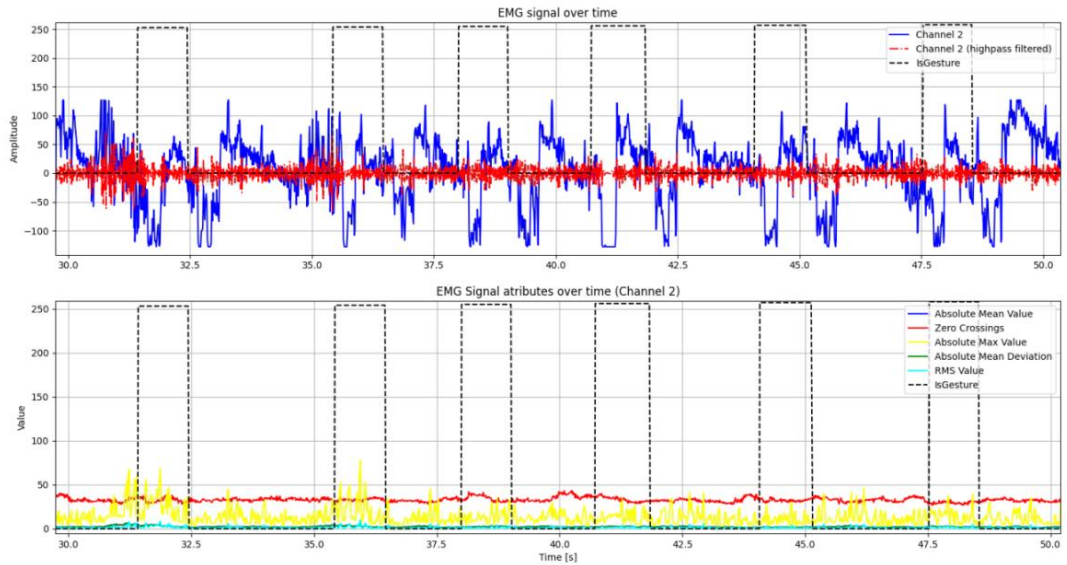
Para determinar esse valor de detecção, foi necessário fazer ensaios e determinar o atributo de maior distinção para detectar o início de cada gesto. Com isto, medindo a actividade muscular bem como as quatro características em cada canal, obtém-se os três diagramas descritos nas figuras Figura 4.8, Figura 4.9 e Figura 4.10.



**Figura 4.8: Captação de quatro movimentos distintos com o respectivo comportamento das características (canal 0).**



**Figura 4.9: Captação de quatro movimentos distintos com o respectivo comportamento das características (canal 1).**



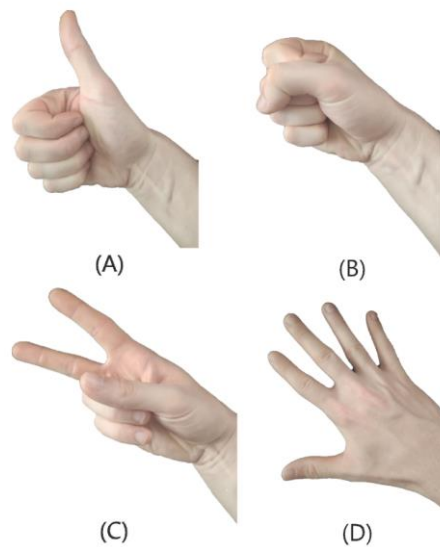
**Figura 4.10: Captação de quatro movimentos distintos com o respectivo comportamento das características (canal 2).**

É de notar que os canais 1 e 2 tinham alguns problemas de ruídos derivados de maus contactos ou até mesmo da qualidade do cabo. Após considerar o comportamento do sinal face a estas quatro características, definiu-se que o módulo do valor de pico seria o melhor para detecção devido a sua maior discrepância nos momentos de actividade muscular para disparar o processo de análise de gestos.

Após a detecção, são analisadas as 100 amostras seguintes, de forma a apanhar valores mais consistentes. Para ilustrar este evento, foi adicionado aos gráficos, a tracejado preto, as zonas de análise. Aqui, é possível apontar dois aspectos. Em primeiro, observa-se então o comportamento do valor de pico a destacar-se, ilustrando o que já foi concluído. Em segundo, observa-se que este não é disparado em todos os movimentos de igual forma pelos sensores. O primeiro movimento disparado, por exemplo, não afectou o sensor 0, e os três últimos movimentos quase não afectaram o sensor 2. No entanto, todos eles afectaram o sensor 1. Isto é esperado, visto que o sensor 1 estava localizado exactamente no meio do braço numa zona de maior actividade.

#### 4.5. Classificação de gestos

Antes de classificar gestos, em primeiro lugar determinou-se efectivamente que gestos iriam ser reconhecidos, apresentados na Figura 4.11. Com isto, para cada um foi necessário recolher amostras para treino, ou seja, retirar várias amostras de "imagens" do sinal EMG para cada um deles. Para cada gesto foram recolhidas 60 amostras, resultando num total de 1200 amostras com 15 atributos cada uma. Simplificando o problema, as amostras foram retiradas sempre com o braço relaxado para haver o mínimo de actividade EMG em simultâneo com cada gesto. Com isto, já se pode passar a implementação de classificadores.



**Figura 4.11: Quatro gestos escolhidos para detecção e classificação (adaptado de [56])**

Com base na bibliografia revista, escolheram-se três classificadores distintos para se concluir qual deles o mais adequado. Escolheu-se um classificador *Random Forest* (RF) de 500 estimadores, um de *Support Vector Machines* (SVM) e uma rede neuronal (RN) de três camadas com 15 perceptrões cada.

Para comparar os classificadores estabelecidos, fez-se uso da biblioteca Python *sklearn* para chegar a resultados de teste. Os resultados de precisão estão apresentados na Tabela 4.1.

Tabela 4.1: Resultados de precisão dos classificadores escolhidos

Gesto	Random Forest	Support Vector Machines	Rede Neuronal
A	96%	100%	88%
B	56%	68%	55%
C	58%	58%	58%
D	81%	86%	85%
<b>Todos</b>	<b>73%</b>	<b>78%</b>	<b>72%</b>

Analisando os resultados obtidos, concluiu-se que o melhor classificador era o de *Support Vector Machines*, finalizando então, a componente de análise de gestos. Disto resulta então, o fluxograma apresentado na Figura 4.12.

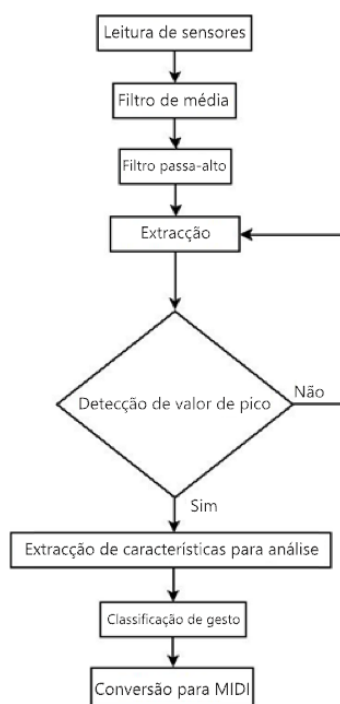


Figura 4.12: Fluxograma do processo executado pelo controlador MIDI implementado (adaptado de [56])

#### 4.6. Processamento MIDI

Nesta altura da cadeia, já estão a ser lidos, detectados e classificados os gestos da mão. Dito isto, sobra apenas converter esses gestos em mensagens MIDI para serem utilizados por um instrumento virtual. A conversão do output

dos classificadores para MIDI é feita ao nível do Arduino, e converte cada gesto numa de quatro notas musicais para ser convertida em som por um sequenciador. No caso desta dissertação, será utilizado o Ableton Live 9, um sequenciador usado para produção musical com instrumentos virtuais internos.

Como foi visto anteriormente, existem mensagens específicas para notas musicais, chamadas de NOTE\_ON e NOTE\_OFF. Uma mensagem de NOTE\_ON representa um início de nota e um NOTE\_OFF um final de nota. Para revisão, a estrutura destas mensagens é apresentada na Figura 2.3 e na

Tabela 2.1. No entanto, para facilitar na implementação, em alternativa a mensagens de NOTE\_OFF, serão enviadas mensagens de NOTE\_ON com velocidade a 0. Com isto, tendo como referência a nomenclatura estabelecida para cada gesto na Figura 4.11, estabeleceu-se o seguinte mapeamento apresentado na Tabela 4.2.

**Tabela 4.2: Mapeamento de gestos para notas musicais**

<b>Gesto</b>	<b>Nota musical</b>	<b>Valor MIDI decimal</b>	<b>Valor MIDI binário</b>
A	Dó	60	00111100
B	Ré#	63	00111111
C	Fá	67	01000011
D	Lá	69	01000101

Visto que desta vez, os dados enviados têm sempre o mesmo tamanho, e não há forma de minimizar o tamanho de cada pacote para reduzir latência, nesta fase já faz sentido implementar a transmissão dos dados MIDI para o sequenciador por UART. Esta informação é enviada para um porto virtual criado com software LoopMIDI, que por sua vez envia esta informação de volta para o Ableton Live 9. Finalmente, enviando essa informação para um instrumento virtual dentro do sequenciador, é finalmente possível reproduzir sons musicais com os gestos da mão, apresentado na Figura 4.13.

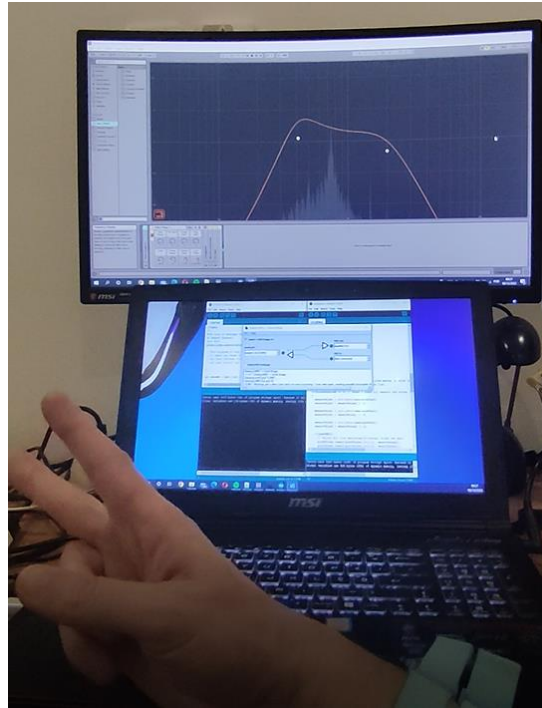


Figura 4.13: Performance musical em Ableton Live 9 (adaptado de [56])

## 5. Discussão de Resultados

Esta dissertação acabou por sofrer alguns desvios relativamente ao que foi planeado inicialmente. Seja por compatibilidade de bibliotecas ou alimentação dos sensores, foram necessárias algumas modificações que mostraram ter grandes impactos no resultado final. Nesta secção serão discutidos os resultados obtidos, as suas causas, bem como possíveis alterações para melhoria dos mesmos.

### 5.1. Leitura da informação EMG

Observando os resultados, sendo o primeiro elemento da cadeia, a leitura dos sensores comprometeu logo os restantes blocos. A necessidade dos sensores Olimex EMG-EKG terem de passar primeiro por um Arduino para implementar a sua leitura tornou o processo muito maior do que esperado, acrescentando um passo de transmissão de dados ao invés de uma leitura directa para o Raspberry PI.

Esta fase revelou-se ainda mais problemática ao implementar a abordagem de transmissão por UART, que numa leitura dos três sensores baixou a frequência de amostragem inicial de 10kHz para 40Hz. Esta amostragem tornava o uso de três sensores impossível, e por consequente, a classificação de gestos também. Felizmente, com a transmissão por SPI, foi possível minimizar e estabilizar o tamanho dos pacotes de transmissão. Por consequente, a amostragem de 40Hz aumentou para 140Hz. Apesar de ainda ser bem longe do ideal, esta frequência de amostragem já possibilitou a captação de um terceiro ponto no braço de uma forma funcional.

### 5.2. Processamento do sinal

Algo difícil de explicar nesta fase é o efeito que a amostragem tem nesta fase. Segundo os cálculos feitos internamente, a amostragem atingida foi de 140Hz, o que significa, segundo o teorema de Nyquist, uma frequência máxima

de 70Hz no sinal EMG captado. Por consequente, apesar de continuar a justificar a implementação do filtro passa-alto em teoria, torna o seu impacto muito menos significativo. Com isto, observando os gráficos das figuras Figura 4.8, Figura 4.9 e Figura 4.10, o impacto foi sem dúvida positivo, estabilizando o sinal para detecção posterior.

### 5.3. Detecção e classificação

Considerando os resultados das fases anteriores, os resultados acabaram por ser melhores do que esperados nesta fase. Tendo uma frequência de amostragem dez vezes abaixo da desejada, seria de esperar que os classificadores não fossem capazes de detectar os padrões desejados. No entanto, ainda assim atingiu-se uma percentagem de precisão de 78%, com dois movimentos (polegar para cima e mão aberta) com precisões acima dos 80% para todos os classificadores.

Tendo em conta os dois movimentos mais reconhecidos, consegue-se retirar alguma correlação nos seus resultados. Estes são os dois movimentos que apresentam completa extensão do polegar. Isto sugere uma melhor captação do extensor do polegar, e por consequente, talvez um posicionamento comprometedor dos sensores para os outros movimentos menos bem classificados.

### 5.4. Conversão para MIDI

Como foi definido em requisito, um dos objectivos era utilizar o mínimo de hardware possível. Como tal, em vez de enviar a informação por cabo MIDI, enviou-se a informação internamente por portos virtuais. Em questões de latência esta abordagem não teve qualquer efeito, no entanto, tem algum impacto na sua interoperabilidade pois passa a exigir mais componentes (LoopMIDI) que com a extracção do sinal MIDI por cabo não seriam necessárias.

## 5.5. Validação dos dados

Existem várias métricas para avaliar o desempenho de um determinado modelo. Para os efeitos desta dissertação, será feito um foco nos seguintes:

- Exatidão
- Precisão
- Sensibilidade
- Coeficiente F1

Dito isto, para chegar a estas métricas, deve-se começar por obter a matriz de confusão para cada um dos classificadores. Os resultados para as tabelas de confusão, apresentadas nas tabelas

**Tabela 5.1: Tabela de confusão para o classificador Random Forest**

<b>Gesto real</b>	Polegar para cima	Mão fechada	Mão aberta	Tesoura
<b>Gesto estimado</b>				
Polegar para cima	22	0	1	0
Mão fechada	6	19	5	4
Mão aberta	0	10	21	5
Tesoura	1	4	0	22

**Tabela 5.2: Tabela de confusão para o classificador de Support Vector Machines**

<b>Gesto real</b>	Polegar para cima	Mão fechada	Mão aberta	Tesoura
<b>Gesto estimado</b>				
Polegar para cima	24	0	0	0
Mão fechada	5	17	2	1
Mão aberta	0	12	25	6
Tesoura	0	4	0	24

**Tabela 5.3: Tabela de confusão para o classificador de rede neuronal**

<b>Gesto real</b>	Polegar para cima	Mão fechada	Mão aberta	Tesoura
<b>Gesto estimado</b>				
Polegar para cima	22	2	0	1
Mão fechada	5	18	6	4
Mão aberta	1	10	21	4
Tesoura	1	3	0	22

Uma matriz de confusão é uma matriz que determina os casos de verdadeiros e falsos positivos, e verdadeiros e falsos negativos, tal como apresentado nas Figura 5.1 e Figura 5.2

		Valores reais	
		Positivos	Negativos
Valores estimados	Positivos	TRUE POSITIVE (TP)	FALSE POSITIVE (FP)
	Negativos	FALSE NEGATIVE (FN)	TRUE NEGATIVE (TN)

Figura 5.1: Formato da matriz de confusão de um classificador

Polegar levantado		Valores reais	
		Positivos	Negativos
Valores estimados	Positivos	22	1
	Negativos	7	90

Figura 5.2: Exemplo de matriz de confusão para o gesto de polegar levantado do classificador Random Forest

As restantes matrizes encontram apresentadas no apêndice A para cada um dos classificadores. Tendo em conta os resultados, é possível então calcular as métricas finais com as seguintes fórmulas:

- $Exatidão = \frac{TP+TN}{Total}$
- $Precisão = \frac{TP}{TP+FP}$
- $Sensibilidade = \frac{TP}{TP+FN}$
- $Coeficiente\ F1 = \frac{2 \times Precisão \times Sensibilidade}{Precisão + Sensibilidade}$

Aplicando estes cálculos a todos os gestos dos três classificadores resultam as tabelas Tabela 5.4, Tabela 5.5 e Tabela 5.6.

Tabela 5.4: Métricas de desempenho para Random Forest

Gesto	Exatidão	Precisão	Sensibilidade	F1-score
Polegar para cima	93.33333%	95.65217%	75.86207%	84.61538%
Mão fechada	75.83333%	55.88235%	57.57576%	56.71642%
Mão aberta	82.50000%	58.33333%	77.77778%	66.66667%
Tesoura	88.33333%	81.48148%	70.96774%	75.86207%
<b>Média</b>	<b>85.00000%</b>	<b>72.83734%</b>	<b>70.54584%</b>	<b>70.96513%</b>

Tabela 5.5: Métricas de desempenho para Support Vector Machines

Gesto	Exatidão	Precisão	Sensibilidade	F1-score
Polegar para cima	95.83333%	100.00000%	82.75862%	90.56604%
Mão fechada	80.00000%	68.00000%	51.51515%	58.62069%
Mão aberta	83.33333%	58.13953%	92.59259%	71.42857%
Tesoura	90.83333%	85.71429%	77.41935%	81.35593%
<b>Média</b>	<b>87.50000%</b>	<b>77.96346%</b>	<b>76.07143%</b>	<b>75.49281%</b>

Tabela 5.6: Métricas de desempenho para rede neuronal

Gesto	Exatidão	Precisão	Sensibilidade	F1-score
Polegar para cima	91.66667%	88.00000%	75.86207%	81.48148%
Mão fechada	75.00000%	54.54545%	54.54545%	54.54545%
Mão aberta	82.50000%	58.33333%	77.77778%	66.66667%
Tesoura	89.16667%	84.61538%	70.96774%	77.19298%
<b>Média</b>	<b>84.58333%</b>	<b>71.37354%</b>	<b>69.78826%</b>	<b>69.97165%</b>

Felizmente, a biblioteca *sklearn* possui já métodos que automatizam todo este processo, nomeadamente o método *sklearn.metrics.classification\_report*, que retorna um relatório completo de todos estes coeficientes de uma só vez, facilitando todo o processo de avaliação e de treino. A primeira conclusão que se pode retirar é, portanto, o óbvio melhor desempenho da parte do classificador SVM. É importante sublinhar que existem também uma diferença de quase 7% na sensibilidade (ou taxa de verdadeiros positivos) entre entre SVM para os outros classificadores. Quanto ao coeficiente F1, observa-se em todos um valor na ordem dos 70%, com SVM a ter o valor máximo com 75%. Enquanto todos estes valores F1 podem ser considerados algo satisfatórios tendo em conta os desvios em montagem tomados, ainda não são exactamente satisfatórios para a aplicação tomada. Dito isto, a conclusão é bastante evidente, sendo o classificador SVM o mais indicado para este sistema.



## 6. Conclusão e trabalho futuro

Apesar de ainda haver alguns melhoramentos a ajustar na captação de sinal, nomeadamente, optimização na transmissão de dados bem como acondicionamento de sinal de forma a limpar o ruído e sinal ECG não desejado, existem alguns pontos que se podem dar como necessários para esta aplicação.

O corte nos 150Hz é uma clara necessidade para limpar o sinal EMG da sua componente ECG, de forma a não interferir na análise dos gestos. Adicionalmente um filtro de média também é necessário de forma a suavizar a curva e estabilizar o sinal de ruídos mecânicos causados por maus contactos.

De forma a optimizar transmissão de dados entre Arduino UNO e Raspberry PI 4, conclui-se que, entre UART e SPI, o protocolo mais indicado foi SPI por permitir a transmissão de blocos de 8 bits de cada vez ao invés de *strings* inteiras, ou seja, grandes blocos vários conjuntos de oito bits. Por outro lado, UART revelou-se mais adequada para a transmissão de sinais MIDI, visto que o tamanho dos pacotes neste andar era constante e mantinha sempre uma estrutura de dados fixa.

Finalmente, é importante referir que esta dissertação teve a sua participação na conferência Young Engineers Forum in Electrical and Computer Engineering (YEF-ECE 2023), com um artigo publicado a descrever e discutir os resultados desta dissertação [56]

### 6.1. Trabalho futuro

Com estas conclusões, existem vários aspectos que podem ser melhorados em três frentes, nomeadamente, frequência de amostragem, tratamento do sinal EMG e facilidade de instalação do controlador.

Em primeiro lugar e mais importante, tendo sido um dos maiores problemas de todo o controlador, a amostragem terá que ser aumentada. Uma forma de fazer isto poderá ser eliminando o Arduino da arquitectura e converter o sinal analógico dos sensores com um conversor mais rápido que o conversor

interno do Arduino, e enviar directamente para o Raspberry PI. Desta forma, eliminar-se-ia um passo muito pesado em termos de processamento em toda a cadeia, aumentando significativamente a frequência de amostragem.

Em segundo lugar, e como consequência de um aumento na frequência, torna-se possível um tratamento mais eficiente do sinal EMG. Uma consequência do filtro implementado foi a perda de amplitude de sinal. Reduzir esta perda de sinal irá oferecer uma maior dinâmica que, por sua vez, facilitará a detecção e classificação de gestos. Adicionalmente, pode-se também experimentar outros filtros para esta fase como é apresentado em [43].

Finalmente, a adição de um módulo MIDI separado do Raspberry PI facilitaria imenso a configuração do controlador, possibilitando uma operação mais directa sem haver a necessidade de configurar portos virtuais.

## 7. Referências

- [1] B. Eren, "Use of Music in Special Education and Application Examples from Turkey," *Procedia - Soc. Behav. Sci.*, vol. 116, pp. 2593–2597, 2014.
- [2] S. M. Srinivasan and A. N. Bhat, "A review of 'music and movement' therapies for children with autism: Embodied interventions for multisystem development," *Front. Integr. Neurosci.*, vol. 7, no. MAR, pp. 1–15, 2013.
- [3] C. M. Zelazny, "Therapeutic instrumental music playing in hand rehabilitation for older adults with osteoarthritis: Four case studies," *J. Music Ther.*, vol. 38, no. 2, pp. 97–113, 2001.
- [4] N. Nikmaram *et al.*, "Musical Sonification of Arm Movements in Stroke Rehabilitation Yields Limited Benefits," *Front. Neurosci.*, vol. 13, no. December, pp. 1–14, 2019.
- [5] B. C. Florea, "MIDI-based controller of electrical drives," *Proc. 2014 6th Int. Conf. Electron. Comput. Artif. Intell. ECAI 2014*, pp. 27–30, 2015.
- [6] D. M. Huber, *The MIDI Manual: A Practical Guide to MIDI in the Project Studio*, 3rd Editio. Focal Press, 2007.
- [7] R. B. Reilly and T. C. Lee, "Electrograms (ECG, EEG, EMG, EOG)," *Technol. Heal. Care*, vol. 18, no. 6, pp. 443–458, 2010.
- [8] T. Dan Vacarus, C. Popescu, A. Moise, and G. Bucur, "Electromyography and applications based on the interpretation of the electrical activity associated with the depolarization-repolarization cycle of the muscle fiber membrane," *Rev. Chim.*, vol. 70, no. 9, pp. 325–336, 2020.
- [9] Olimex, "SHIELD-EKG-EMG bio-feedback shield USER'S MANUAL," *SHIELD-EKG-EMG*, 2014.
- [10] L. Dipietro, A. M. Sabatini, and P. Dario, "A survey of glove-based systems and their applications," *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 38, no. 4, pp. 461–482, 2008.
- [11] D. J. Sturman and D. Zeltzer, "A Survey of Glove-based Input," pp. 30–39, 1994.
- [12] D. J. Sandin and T. A. DeFanti, "Sayre Glove - US NEA R60-34-163 Final Project Report." pp. 1–30, 1977.
- [13] G. J. Grimes and P. E. J. Groody, "Digital Data Entry Glove Interface Device," 1983.

- [14] M. Velte, "Bachelor Thesis A MIDI Controller based on Human Motion Capture by Maurice Velte," 2014.
- [15] A. H. J. Moreira, S. Queirós, J. Fonseca, P. L. Rodrigues, N. F. Rodrigues, and J. L. Vilaça, "Real-time hand tracking for rehabilitation and character animation," *SeGAH 2014 - IEEE 3rd Int. Conf. Serious Games Appl. Heal. Books Proc.*, no. June 2015, 2014.
- [16] Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE Multimed.*, vol. 19, no. 2, pp. 4–10, 2012.
- [17] ST Microelectronics, "MEMS motion sensor: three-axis digital output gyroscope," *L3GD20*, 2010.
- [18] ST Microelectronics, "Ultra-compact high-performance eCompass module: 3D accelerometer and 3D magnetometer," *LSM303DLHC*, 2013.
- [19] ST-Microelectronics, "STM32F405xx STM32F407xx Rev4," no. June, pp. 1–185, 2013.
- [20] Texas Instruments, "TCA9548A Low-Voltage 8-Channel I2C Switch with Reset," *TCA9548A*, 2019.
- [21] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm ACT Profile Report: State. Graduating Class 2012. Montana," *IEEE Int. Conf. Rehabil. Robot. Rehab Week Zurich, ETH Zurich Sci. City, Switzerland*, pp. 1–7, 2011.
- [22] R. Mahony, T. Hamel, and J. M. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Trans. Automat. Contr.*, vol. 53, no. 5, pp. 1203–1218, 2008.
- [23] E. R. Bachmann, R. B. McGhee, X. Yun, and M. J. Zyda, "Inertial and magnetic posture tracking for inserting humans into networked virtual environments," *ACM Symp. Virtual Real. Softw. Technol. Proceedings, VRST*, pp. 9–16, 2001.
- [24] P. Martin and E. Salaün, "Design and implementation of a low-cost observer-based attitude and heading reference system," *Control Eng. Pract.*, vol. 18, no. 7, pp. 712–722, 2010.
- [25] J. L. Hernandez-Rebollar, N. Kyriakopoulos, and R. W. Lindeman, "A new instrumented approach for translating American Sign Language into sound and text," *Proc. - Sixth IEEE Int. Conf. Autom. Face Gesture Recognit.*, no. June 2014, pp. 547–552, 2004.
- [26] J. L. Hernandez-Rebollar, N. Kyriakopoulos, and R. W. Lindeman, "The AcceleGlove: A whole-hand input device for virtual reality," *ACM SIGGRAPH 2002 Conf. Abstr. Appl. SIGGRAPH 2002*, p. 259, 2002.
- [27] J. L. Hernandez-Rebollar, R. W. Lindeman, and N. Kyriakopoulos, "A

- multi-class pattern recognition system for practical finger spelling translation," *Proc. - 4th IEEE Int. Conf. Multimodal Interfaces, ICMI 2002*, pp. 185–190, 2002.
- [28] R. D. J. Friedman N., Chan V., Zondervan D., Bachman M., "MusicGlove: motivating and quantifying hand," *33rd Annu. Int. Conf. IEEE MBS*, pp. 2359–2363, 2011.
- [29] G. D. Kessler, L. F. Hodges, and N. Walker, "Evaluation of the CyberGlove as a Whole-Hand Input Device," *ACM Trans. Comput. Interact.*, vol. 2, no. 4, pp. 263–283, 1995.
- [30] H. H. S. Ip, K. C. K. Law, and B. Kwong, "Cyber composer: Hand gesture-driven intelligent music composition and generation," *Proc. 11th Int. Multimed. Model. Conf. MMM 2005*, pp. 46–52, 2005.
- [31] Polhemus, "Fastrack 3d digitizer & quad sensor motion tracker," pp. 1–2, 2017.
- [32] N. Parajuli *et al.*, "Real-time EMG based pattern recognition control for hand prostheses: A review on existing methods, challenges and future implementation," *MDPI*, vol. 19, no. 20, 2019.
- [33] B. Hudgins, P. Parker, and R. N. Scott, "A New Strategy for Multifunction Myoelectric Control," *IEEE Trans. Biomed. Eng.*, vol. 40, no. 1, pp. 82–94, 1993.
- [34] O. W. Samuel *et al.*, "Intelligent EMG pattern recognition control method for upper-limb multifunctional prostheses: Advances, current challenges, and future prospects," *IEEE Access*, vol. 7, no. c, pp. 10150–10165, 2019.
- [35] J. Falk-dahlin, "Evaluation of Post-Processing Strategies for Simultaneous Pattern Recognition Based Myoelectric Prosthetic Control," Chalmers University of Technology, 2013.
- [36] E. J. Scheme, K. B. Englehart, and B. S. Hudgins, "Selective classification for improved robustness of myoelectric control under nonideal conditions," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 6, pp. 1698–1705, 2011.
- [37] S. Amsuss, P. M. Goebel, N. Jiang, B. Graimann, L. Paredes, and D. Farina, "Self-correcting pattern recognition system of surface EMG signals for upper limb prosthesis control," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 4, pp. 1167–1176, 2014.
- [38] K. Englehart, B. Hudgins, and P. A. Parker, "A wavelet-based continuous classification scheme for multifunction myoelectric control," *IEEE Trans. Biomed. Eng.*, vol. 48, no. 3, pp. 302–311, 2001.
- [39] Y. Huang, K. B. Englehart, B. Hudgins, and A. D. C. Chan, "A Gaussian mixture model based classification scheme for myoelectric control of powered upper limb prostheses," *IEEE Trans. Biomed. Eng.*, vol. 52, no. 11,

- pp. 1801–1811, 2005.
- [40] A. M. Simon, L. J. Hargrove, B. A. Lock, and T. A. Kuiken, “A decision-based velocity ramp for minimizing the effect of misclassifications during real-time pattern recognition control,” *IEEE Trans. Biomed. Eng.*, vol. 58, no. 8, pp. 2360–2368, 2011.
  - [41] H. Wu, M. Dyson, and K. Nazarpour, “Arduino-based myoelectric control: Towards longitudinal study of prosthesis use,” *Sensors (Switzerland)*, vol. 21, no. 3, pp. 1–13, 2021.
  - [42] D. L. Espinoza and L. Eli Sanchez Velasco, “Comparison of EMG signal classification algorithms for the control of an upper limb prosthesis prototype,” *2020 17th Int. Conf. Electr. Eng. Comput. Sci. Autom. Control. CCE 2020*, 2020.
  - [43] B. Rodríguez-Tapia, I. Soto, D. M. Marínez, and N. C. Arballo, “Myoelectric Interfaces and Related Applications: Current State of EMG Signal Processing-A Systematic Review,” *IEEE Access*, vol. 8, pp. 7792–7805, 2020.
  - [44] M. Asghari Oskoei and H. Hu, “Myoelectric control systems-A survey,” *Biomed. Signal Process. Control*, vol. 2, no. 4, pp. 275–294, 2007.
  - [45] M. N. Castro, J. Rasmussen, M. S. Andersen, and S. Bai, “A compact 3-DOF shoulder mechanism constructed with scissors linkages for exoskeleton applications,” *Mech. Mach. Theory*, vol. 132, pp. 264–278, 2019.
  - [46] S. Crea *et al.*, “A novel shoulder-elbow exoskeleton with series elastic actuators,” *Proc. IEEE RAS EMBS Int. Conf. Biomed. Robot. Biomechatronics*, vol. 2016-July, pp. 1248–1253, 2016.
  - [47] H. C. Hsieh, D. F. Chen, L. Chien, and C. C. Lan, “Design of a Parallel Actuated Exoskeleton for Adaptive and Safe Robotic Shoulder Rehabilitation,” *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 5, pp. 2034–2045, 2017.
  - [48] Y. Mao and S. K. Agrawal, “Transition from mechanical arm to human arm with CAREX: A cable driven ARm EXoskeleton (CAREX) for neural rehabilitation,” *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 2457–2462, 2012.
  - [49] S. Lessard, P. Pansodtee, A. Robbins, J. M. Trombadore, S. Kurniawan, and M. Teodorescu, “A Soft Exosuit for Flexible Upper-Extremity Rehabilitation,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 8, pp. 1604–1617, 2018.
  - [50] A. Ebrahimi, D. Groninger, R. Singer, and U. Schneider, “Control parameter optimization of the actively powered upper body exoskeleton using subjective feedbacks,” *2017 3rd Int. Conf. Control. Autom. Robot. ICCAR 2017*, pp. 432–437, 2017.
  - [51] E. Vlachos, E. Jochum, and L. P. Demers, “HEAT: The Harmony

- Exoskeleton Self - Assessment Test," *RO-MAN 2018 - 27th IEEE Int. Symp. Robot Hum. Interact. Commun.*, pp. 577–582, 2018.
- [52] I. A. Ben, Y. Bouteraa, and C. Rekik, "Design and development of 3d printed myoelectric robotic exoskeleton for hand rehabilitation," *Int. J. Smart Sens. Intell. Syst.*, vol. 10, no. 2, pp. 341–366, 2017.
- [53] S. Cheon *et al.*, "Single EMG Sensor-Driven Robotic Glove Control for Reliable Augmentation of Power Grasping," *IEEE Trans. Med. Robot. Bionics*, vol. 3, no. 1, pp. 1–1, 2020.
- [54] F. Botros, A. Phinyomark, and E. Scheme, "EMG-Based Gesture Recognition: Is It Time to Change Focus from the Forearm to the Wrist?," *IEEE Trans. Ind. Informatics*, no. January 2021, 2020.
- [55] T. Tsujimura, S. Yamamoto, and K. Izumi, "Hand Sign Classification Employing Myoelectric Signals of Forearm," *Comput. Intell. Electromyogr. Anal. - A Perspect. Curr. Appl. Futur. Challenges*, 2012.
- [56] S. Pedrosa and A. Costa, "EMG Based MIDI Controller," *Proc. - 2023 Int. Young Eng. Forum Electr. Comput. Eng. YEF-ECE 2023*, pp. 106–111, 2023.
- [57] P. Konard, "The ABC of EMG," *Noraxon: Scottsdale*, no. April, pp. 1–60, 2012.
- [58] L. Shaw and S. Bagha, "Online Emg Signal Analysis for Diagnosis of Neuromuscular Diseases By Using Pca and Pnn," *Int. J. Eng. Sci.*, vol. 4, no. 10, pp. 4453–4459, 2012.



## A. Matrizes de confusão

### A.1. Random Forest

Tabela A.1: Matriz de confusão para o gesto de polegar levantado para o classificador Random Forest

		Valores reais	
		Positivos	Negativos
Valores estimados	Positivos	22	1
	Negativos	7	90

Tabela A.2: Matriz de confusão para o gesto de mão fechada para o classificador Random Forest

		Valores reais	
		Positivos	Negativos
Valores estimados	Positivos	19	15
	Negativos	14	72

Tabela A.3: Matriz de confusão para o gesto de mão aberta para o classificador Random Forest

		Valores reais	
		Positivos	Negativos
Valores estimados	Positivos	21	15
	Negativos	6	78

**Tabela A.4: Matriz de confusão para o gesto de tesoura para o classificador Random Forest**

		Valores reais	
		Positivos	Negativos
Valores estimados	Positivos	22	5
	Negativos	9	84

## A.2. Support Vector Machines

**Tabela A.5: Matriz de confusão para o gesto de polegar levantado para o classificador Support Vector Machines**

		Valores reais	
		Positivos	Negativos
Valores estimados	Positivos	24	0
	Negativos	5	91

**Tabela A.6: Matriz de confusão para o gesto de mão fechada para o classificador Support Vector Machines**

		Valores reais	
		Positivos	Negativos
Valores estimados	Positivos	17	8
	Negativos	16	79

**Tabela A.7: Matriz de confusão para o gesto de mão aberta para o classificador Support Vector Machines**

		Valores reais	
		Positivos	Negativos
Valores estimados	Positivos	25	18
	Negativos	2	75

**Tabela A.8: Matriz de confusão para o gesto de tesoura para o classificador Support Vector Machines**

		Valores reais	
		Positivos	Negativos
Valores estimados	Positivos	24	4
	Negativos	7	85

### A.3. Rede neuronal

**Tabela A.9: Matriz de confusão para o gesto de polegar levantado para o classificador de rede neuronal**

		Valores reais	
		Positivos	Negativos
Valores estimados	Positivos	22	3
	Negativos	7	88

**Tabela A.10: Matriz de confusão para o gesto de mão fechada para o classificador de rede neuronal**

		Valores reais	
		Positivos	Negativos
Valores estimados	Positivos	18	15
	Negativos	15	72

**Tabela A.11: Matriz de confusão para o gesto de mão aberta para o classificador de rede neuronal**

		Valores reais	
		Positivos	Negativos
Valores estimados	Positivos	21	15
	Negativos	6	78

**Tabela A.12: Matriz de confusão para o gesto de tesoura para o classificador de rede neuronal**

		Valores reais	
		Positivos	Negativos
Valores estimados	Positivos	22	4
	Negativos	9	85

