

**UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS**

Daniel Spiewak Finimundi

**Análise de viabilidade de algoritmo para avaliação
cinemática 3D de pronação e supinação do antebraço**

São Carlos

2016

Daniel Spiewak Finimundi

**Análise de viabilidade de algoritmo para avaliação
cinemática 3D de pronação e supinação do antebraço**

Trabalho de Conclusão de Curso apresentado
à Escola de Engenharia de São Carlos da
Universidade de São Paulo.

Curso de Engenharia da Computação

Orientador: Prof. Dr. Alberto Cliquet Junior

**São Carlos
2016**

Autorizo a reprodução total ou parcial deste trabalho,
por qualquer meio convencional ou eletrônico, para fins
de estudo e pesquisa, desde que citada a fonte.

S498a Spiewak Finimundi, Daniel
 Análise de viabilidade de algoritmo para avaliação
 cinemática 3D de pronação e supinação do antebraço /
 Daniel Spiewak Finimundi; orientador Alberto Cliquet
 Junior. São Carlos, 2016.

 Monografia (Graduação em Engenharia de Computação)
-- Escola de Engenharia de São Carlos da Universidade
de São Paulo, 2016.

 1. Engenharia Biomecânica. 2. Engenharia de
Reabilitação. 3. Instrumentação. 4. Sistema de Captura
de Movimento. 5. Python. I. Título.

FOLHA DE APROVAÇÃO

Nome: Daniel Spiewak Finimundi

Título: “Análise de viabilidade de algoritmo para avaliação cinemática 3D de pronação e supinação do antebraço”

Trabalho de Conclusão de Curso defendido em 16 / 11 / 2016.

Comissão Julgadora:

Prof. Titular Alberto Cliquet Júnior
(Orientador) - SEL/EESC/USP

Profa. Dra. Eliza Regina Ferreira Braga Machado
de Azevedo
UFSCar

Dr. Renato Varoto
Pós-doutorado/UNICAMP

Resultado:

Aprovado

Aprovado

Aprovado

Coordenador do Curso Interunidades Engenharia de Computação pela EESC:

Prof. Dr. Maximilian Luppe

*Este trabalho é dedicado à todos aqueles que acreditam
que o mundo ainda pode ser muito melhor.*

AGRADECIMENTOS

Agradeço à Universidade de São Paulo (USP) e os professores e funcionários com os quais tive contato na graduação, pela oportunidade de me desenvolver pessoalmente e profissionalmente dentro desse espaço de pesquisa.

Agradeço ao meu orientador Alberto Cliquet Junior e ao meu co-orientador Renato Varoto por todo o suporte e guia, ao mesmo tempo que promoviam a autonomia de pensamento.

Agradeço à professora Dra. Paula Rezende Camargo da Universidade Federal de São Carlos (UFSCar) por disponibilizar a sala de avaliação cinemática do departamento de fisioterapia para realização dos experimentos desse trabalho.

Agradeço à fisioterapeuta M.^a Lianna Rosa da UFSCar, por toda a ajuda na utilização da sala de avaliação cinemática, além do auxílio com o voluntário.

Agradeço à Dra. Karina Cristina Alonso e à Sabrina Ribeiro de Almeida da Universidade Estadual de Campinas (UNICAMP) pela ajuda inicial no sistema óptico de captura de movimentos do Laboratório de Biomecânica e Reabilitação do Aparelho Locomotor do Hospital das Clínicas da UNICAMP.

Agradeço ao meu amigo Eduardo Aramizu que foi o voluntário para os experimentos desse trabalho.

Agradeço aos meus amigos e colegas, por fazerem as noites de trabalhos e estudos serem lembradas com risadas de uma época boa que pudemos dividir e crescer juntos. E também aos amigos do colegial, que mesmo de longe, ainda influenciam nos meus resultados e conquistas.

Agradeço minha noiva, também fisioterapeuta, por me introduzir à terminologia de anatomia, por me ajudar na correção desse texto, por me ouvir, me aconselhar, me estimular, e principalmente pelo apoio e companhia incondicionais. Estando sempre comigo, mesmo quando estava longe.

Agradeço meus pais, irmã, avós e toda a família, por sempre acreditarem em mim, me apoiarem e me darem força para ser o meu melhor.

“Entendo todas as expressões das minhas possibilidades como minhas e pago, por isso, o preço estranho de não estar propriamente em nenhuma delas.”

Marcelo Camelo

RESUMO

FINIMUNDI, D. S. **Análise de viabilidade de algoritmo para avaliação cinemática 3D de pronação e supinação do antebraço.** 2016. 74p. Trabalho de Conclusão de Curso - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2016.

O objetivo deste trabalho é prover o estudo de um método alternativo *open source* para medição de rotação axial do antebraço, já que atuais métodos são, ou rudimentares, como o goniômetro, ou muito caros e complexos, como softwares de processamento de imagens. A proposta é a implementação de um algoritmo de código aberto para processamento dos dados de um sistema óptico de captura de movimento, com base na literatura e padrões já estabelecidos. Para testar esse algoritmo, foi utilizada uma sala de avaliação cinemática para realizar experimentos em um voluntário, com ajuda de uma fisioterapeuta. Os resultados obtidos tiveram um alto grau de consistência, com diferenças de no máximo 2° entre os testes. Porém, houve um erro maior em relação à medição do goniômetro, até 7° de diferença. Com isso, mostramos que existe uma viabilidade em se realizar medições com esse sistema, porém mais testes são necessários para corrigir os erros e validar o algoritmo.

Palavras-chave: Engenharia Biomecânica. Engenharia de Reabilitação. Instrumentação. Sistema de Captação de Movimento. Python.

ABSTRACT

FINIMUNDI, D. S. **Feasibility analysis of algorithm for 3D kinematic evaluation of elbow pronation and supination.** 2016. 74p. Trabalho de Conclusão de Curso - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2016.

The main goal for this work is to provide a study for an open source alternative method for measuring the forearm's axial rotation, since current methods are either primitives, like the universal goniometer, or too expensive and complex, like imaging processing software. The proposal is the implementation of an open source algorithm for data processing of an optic motion capture system, based on current literature and established patterns. In order to test this algorithm, a kinematics evaluation room was used to perform experiments in a volunteer, with help from a physiotherapist. The results had a high level of consistency, with maximum variation of 2° between the tests. However, the errors in comparison with a goniometer were as high as 7° . Thereby, it was shown that it is feasible to perform measurements with this system, although more test are needed in order to correct these errors and validate the algorithm.

Keywords: Biomechanical Engineering. Rehabilitation Engineering. Instrumentation. Motion Capture System. Python.

LISTA DE ILUSTRAÇÕES

Figura 1 – Imagem da órtese híbrida em funcionamento	23
Figura 2 – Imagens indicando como são realizados os movimentos de pronação e supinação do antebraço.	28
Figura 3 – Localização dos pontos anatômicos.	33
Figura 4 – Sistema de coordenadas locais do antebraço.	34
Figura 5 – Sistema de coordenadas locais do úmero.	34
Figura 6 – Foto da câmera Oqus 300 utilizada no experimento.	37
Figura 7 – Representação do posicionamento das câmeras (retângulos azuis). O retângulo vermelho representa a posição do volume de captura.	37
Figura 8 – Imagem gerada pelo software QTM indicando o cone de visão de cada uma das sete câmeras.	38
Figura 9 – Foto de um dos marcadores utilizados na captura.	39
Figura 10 – Varinha utilizada na calibração (750 mm).	43
Figura 11 – Suporte de referência (compatível com 750 mm).	43
Figura 12 – Marcadores posicionados, imagem lateral direita.	45
Figura 13 – Marcadores posicionados, imagem lateral esquerda.	45
Figura 14 – Marcadores posicionados, imagem frontal.	45
Figura 15 – Marcadores posicionados, imagem traseira.	45
Figura 16 – Imagem do software QTM mostrando os marcadores (círculos verdes) dentro do volume de calibração. A origem do eixo de coordenadas global é indicada pelas setas na imagem.	46
Figura 17 – Ilustração da sequência de movimentos a ser realizada.	46
Figura 18 – Exemplo de um arquivo TSV visualizado em um editor de planilhas.	47
Figura 19 – Gráfico da distância entre os dois marcadores em função do tempo	52
Figura 20 – Imagem demonstrando a utilização da classe ProSupADM	53
Figura 21 – Tela inicial do site para utilização do algoritmo desenvolvido	54
Figura 22 – Tela de resultados do site	55
Figura 23 – Gráfico da primeira captura do ângulo de rotação axial do antebraço	56
Figura 24 – Gráfico da segunda captura do ângulo de rotação axial do antebraço	57
Figura 25 – Gráfico da terceira captura do ângulo de rotação axial do antebraço	57
Figura 26 – Medição com o goniômetro universal. Posição neutra de rotação.	58
Figura 27 – Medição com o goniômetro universal. Movimento de supinação.	58
Figura 28 – Gráfico do ângulo de rotação axial por tempo nos 3 segundos iniciais da coleta, momento em que o antebraço está parado em posição neutra.	59
Figura 29 – Gráfico da densidade de potência por frequência.	60

LISTA DE TABELAS

Tabela 1 – Resultado da ADM para cada coleta e média geral	58
--	----

LISTA DE ABREVIATURAS E SIGLAS

MMSS	Membros superiores
AVD	Atividades de vida diária
ADM	Amplitude de movimento
2D	Duas dimensões
3D	Três dimensões
EENM	Estimulação Elétrica Neuromuscular
STC	<i>Standardization and Terminology Committee</i> , ou Comitê de Normalização e Terminologia, em português
ISB	<i>International Society of Biomechanics</i> , ou Sociedade Internacional de Biomecânica, em português
JCS	<i>Joint Coordinate System</i> , ou Sistema de Coordenadas de Articulações, em português
QTM	Software para visualização, processamento e análise dos dados obtidos na captura
TCP/IP	Protocolo de rede utilizado como padrão em uma rede <i>Ethernet</i>
IP	Endereço de um dispositivo em uma rede TCP/IP
DHCP	Protocolo cliente/servidor para atribuir endereços IP automaticamente
QDS	Software da <i>Qualisys</i> para serviço de DHCP
LED	Diodo emissor de luz
TSV	<i>Tab separated values</i> . Define um arquivo com valores separados por <i>tab</i>
GH	Centro de rotação glenoumeral
AC	Ponto mais dorsal na articulação acromioclavicular
EL	Ponto mais caudal no epicôndilo lateral
EM	Ponto mais caudal no epicôndilo medial
RS	Ponto mais caudal-lateral no processo estilóide do rádio
US	Ponto mais caudal-medial no processo estilóide da ulna

SUMÁRIO

1	INTRODUÇÃO	23
1.1	Contextualização e Motivações	23
1.2	Objetivos	24
2	EMBASAMENTO TEÓRICO	25
2.1	Órtese Híbrida	25
2.2	Biomecânica e Avaliação Cinemática das Articulações	25
2.3	Pronação/supinação do antebraço	26
2.4	Análise Cinemática	27
2.4.1	Padronização da representação de corpos rígidos em 3D	29
2.4.2	<i>Joint Coordinate System (JCS)</i>	30
2.4.3	JCS do cotovelo	32
2.4.4	Sistemas de coordenadas local para a articulação do cotovelo	32
2.5	Álgebra vetorial	34
3	DESENVOLVIMENTO	37
3.1	Sistema de captura de movimento	37
3.1.1	Marcadores	39
3.1.2	Quantidade de câmeras	39
3.1.3	Comunicação do sistema	40
3.1.4	Linearização das câmeras	40
3.1.5	Captura das imagens	40
3.1.6	Calibração	42
3.1.7	Processamento dos dados	43
3.2	Processo de coleta dos dados	44
3.3	Algoritmo para cálculo da ADM	47
4	RESULTADOS E DISCUSSÃO	51
4.1	Validação do sistema de captura de movimento	51
4.2	Software em Python	51
4.3	Experimento com voluntário	56
4.4	Análise de ruído	58
4.5	Discussão	59
4.5.1	Ruído	61
5	CONCLUSÃO	63

REFERÊNCIAS	65
APÊNDICES	67
APÊNDICE A – IMPLEMENTAÇÃO DO ALGORITMO EM PYTHON	69

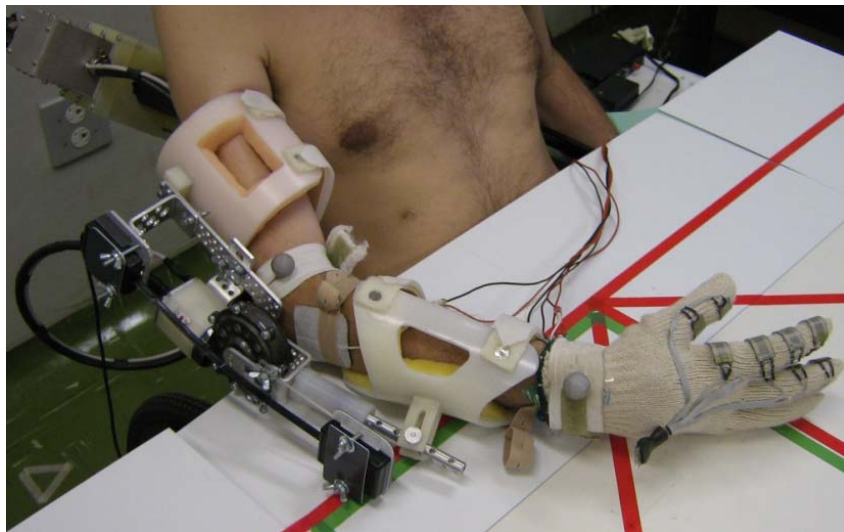
1 INTRODUÇÃO

1.1 Contextualização e Motivações

Indivíduos com disfunções motoras adquiridas ou idiopáticas que afetam os membros superiores (MMSS) podem apresentar diversas dificuldades nas realização de movimentos que afetam suas atividades de vida diária (AVDs). Isso por limitação de força, disparo nervoso ou amplitude de movimento (ADM). Uma das limitações que mais afeta a qualidade de vida é a dificuldade em realizar o movimento de alcance e preensão de objetos.

Pensando na melhora da qualidade de vida destes indivíduos, um dos possíveis auxílios ao tratamento é a utilização do equipamento órtese híbrida proposta por [Varoto \(2010\)](#) que pode ser vista na [Figura 1](#).

Figura 1 – Imagem da órtese híbrida em funcionamento



Fonte: Retirado de [Varoto \(2010\)](#)

Porém, uma limitação dessa órtese é a ausência dos movimentos no plano transversal de pronação (movimento de rotação medial do punho em relação ao cotovelo) e supinação (movimento de rotação lateral do punho em relação ao cotovelo) do antebraço, o qual permite que seja realizada a preensão de objetos em diferentes planos. Para que a órtese possa realizar esses movimentos, é necessário levar em consideração qual o grau de ADM passiva de cada indivíduo de forma que o equipamento não exceda esses limites, gerando lesões. Essa limitação é medida pelos graus de movimento.

Existem diversos métodos para realizar esta medição, desde equipamentos simples como o goniômetro universal ([MARQUES, 2003](#)), utilizado mais na prática clínica por seu

baixo custo e praticidade, porém é pouco preciso; até sistemas computacionais complexos ([ARMSTRONG et al., 1998](#)) que apresentam leituras mais precisas, porém possuem alto custo e são menos intuitivos.

1.2 Objetivos

O objetivo desse trabalho é desenvolver um software *open source* para realizar a medida da ADM dos movimentos de pronação e supinação do antebraço com base nos dados coletados de um sistema de avaliação cinemática composto por câmeras infravermelhas que identificam marcadores reflexivos posicionados no membro superior do indivíduo.

2 EMBASAMENTO TEÓRICO

2.1 Órtese Híbrida

Na reabilitação de indivíduos com disfunções motoras, um dos possíveis auxílios ao tratamento é a utilização de órteses, tanto para membros inferiores como superiores. Além das órteses, outras técnicas também são utilizadas, como estimulação elétrica neuromuscular (EENM). Ambas as técnicas auxiliam por meio da realização de tarefas repetitivas automatizadas. As órteses híbridas promovem a integração entre esses dois métodos: são compostos por uma parte mecânica e outra de estimulação elétrica. Com isso em mente, [Varoto \(2010\)](#) desenvolveu um sistema híbrido para o membro superior direito, [Figura 1](#).

Esse sistema é constituído de uma órtese dinâmica no cotovelo que realiza o movimento de flexão e extensão, com suporte ao antebraço e fixação do punho, em conjunto com EENM para o movimento de preensão de objetos e uma luva sensorial para fornecer *feedback* da força que está sendo utilizada.

A órtese é conectada a um computador que contém um software para configurar os movimentos que deverão ser realizados. Com esses movimentos definidos, o computador é responsável por enviar os sinais de controle para a órtese. Com isso, o sistema realiza esses movimentos repetidamente.

2.2 Biomecânica e Avaliação Cinemática das Articulações

Nos últimos anos, periódicos como *Journal of Biomechanics* da *Elsevier* tem publicado inúmeros artigos que contribuem para melhor compreensão da biomecânica músculo-esquelética humana.

A análise biomecânica passou por uma grande evolução de seus métodos chegando hoje à sua precisão por meio da tecnologia. Apesar de ainda serem utilizados equipamentos mecânicos simples pela sua praticidade e baixo custo, outros equipamentos computadorizados foram desenvolvidos, desde raio-x e eletromiografia, até métodos ópticos com a utilização de câmeras ([NIGG; HERZOG, 2007](#)).

Dentre esses métodos, o que possui melhor relação entre precisão e versatilidade é o óptico. Nele podem ser utilizadas câmeras de vídeo ou infravermelho ([BARROS et al., 1999](#)).

No caso de se utilizar câmeras de vídeo, a vantagem é a versatilidade de não precisar de marcadores posicionados no corpo a ser avaliado. Porém os algoritmos para processar a imagem são mais complexos e produzem resultados menos precisos, com mais ruídos. O

ambiente para a utilização desse tipo de medição possui restrições para garantir a boa qualidade da imagem. Um exemplo de sistema frequentemente utilizado no Brasil que utiliza essa tecnologia é o "Dvideow", desenvolvido por [Barros et al. \(1999\)](#) na Universidade Estadual de Campinas (UNICAMP).

No caso das câmeras infravermelhas, são utilizados marcadores passivos ou ativos. Esses marcadores são esferas posicionadas em locais estratégicos no objeto a ser identificado. No caso dos marcadores passivos, as câmeras ou uma fonte externa emite a luz infravermelha e as esferas são cobertas por um material reflexivo, que irá refletir essa luz que será captada pelas câmeras. No caso dos marcadores ativos, as esferas possuem um diodo emissor de luz (LED) infravermelho, facilitando sua identificação, porém com a desvantagem de precisarem de bateria ou cabos para serem ativadas. ([BARROS et al., 1999](#))

Como essas câmeras infravermelhas só identificam luz infravermelha, elas tem mais facilidade de encontrar o ponto desejado (marcadores) do que uma câmera de vídeo. Com isso os resultados tem melhor precisão. Porém como o Sol emite luz infravermelha, modelos de câmeras mais simples não podem realizar a coleta em ambientes externos, especialmente se forem utilizados marcadores passivos. Apesar disso, câmeras mais modernas já conseguem realizar medições em ambientes externos e até submersas em água. ([BARROS et al., 1999](#))

Um dos sistemas mais utilizados de câmeras infravermelhas é o da empresa sueca [Qualisys AB \(1989\)](#). Esta empresa possui soluções desde as câmeras especializadas até sistemas de processamento dos dados adquiridos para facilitar sua interpretação. Outros sistemas de destaque são o Vicon e o Optotrak Certus.

2.3 Pronação/supinação do antebraço

De acordo com [Kapandji \(1987\)](#), a pronação-supinação é o movimento de rotação do antebraço ao redor do seu eixo longitudinal. Graças a este movimento, a mão pode segurar uma bandeja ou um objeto, em supinação, ou comprimir um objeto para baixo e inclusive se apoiar em pronação. Também permite que se realize um movimento de rotação nas preensões centradas e rotativas, como no caso em que utilizamos uma chave de fenda na qual o eixo do utensílio coincide com o eixo de pronação/supinação.

Este movimento precisa da intervenção de duas articulações mecanicamente unidas, a articulação rádio-ulnar superior, que pertence anatomicamente ao cotovelo e a articulação rádio-ulnar inferior (que é diferente anatomicamente da articulação rádio-carpal).

As duas articulações rádio-ulnares são coaxiais: o seu funcionamento normal necessita de que o eixo de uma seja o prolongamento do eixo da outra sobre uma mesma reta que constitui o eixo de rotação da pronação-supinação e passa pelo centro das cabeças ulnar distalmente e radial proximalmente.

Durante esse movimento de rotação axial do antebraço, a ulna fica praticamente

estacionária, enquanto o rádio gira em torno dela.

Só é possível analisar a pronação/supinação com o cotovelo flexionado a 90° e encostado no corpo. De fato, se o cotovelo está estendido, o antebraço se encontra no prolongamento do braço e na rotação longitudinal do antebraço se acrescenta a rotação do braço ao redor do seu eixo longitudinal, graças aos movimentos de rotação externa e interna do ombro.

Com o cotovelo em flexão de 90° :

- a posição de supinação se realiza quando a palma da mão se dirige para cima com o polegar para fora. Nesta situação a ulna e o rádio estão um ao lado do outro, a ulna por dentro e o rádio por fora. Os seus eixos longitudinais são paralelos;
- a posição de pronação se realiza quando a palma da mão se dirige para baixo e o polegar para dentro. A ulna e o rádio estão cruzados: o rádio está por cima, externo com relação à ulna na articulação rádio-ulnar proximal, e interno com relação à ulna na articulação rádio-ulnar distal;
- a posição intermédia (neutra) é determinada pela direção do polegar para cima e da palma para dentro, ou seja, nem pronação, nem supinação. As amplitudes dos movimentos de pronação-supinação se medem a partir desta posição intermédia (posição zero).

Para a realização da medição devemos observar o antebraço e a mão alinhados e de frente, isto é, no prolongamento do eixo longitudinal, de acordo com a [Figura 2](#) e seguindo:

- a mão em posição intermédia (neutra) se situa no plano vertical, paralela ao plano sagital, plano de simetria do corpo;
- a mão em posição de supinação se situa no plano horizontal; assim sendo, a amplitude de movimento de supinação é de 90° .
- a mão em posição de pronação só se aproxima do plano horizontal no outro sentido; assim, a amplitude de pronação é de 85° .

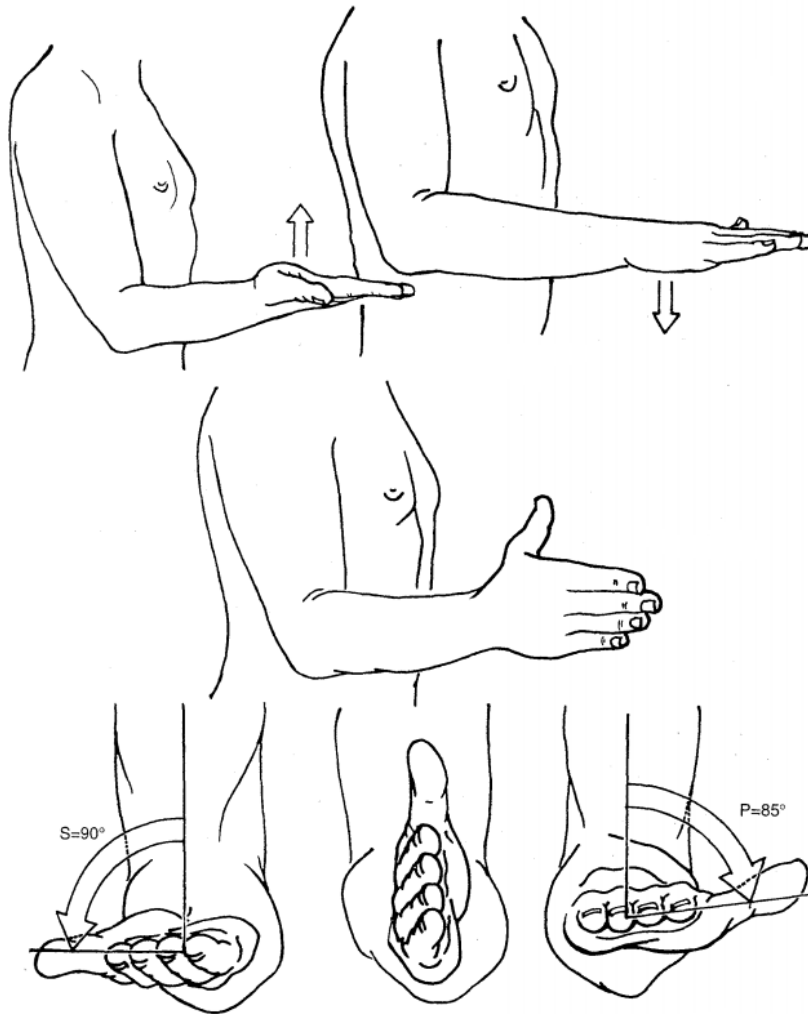
Em resumo, a amplitude total da verdadeira pronação-supinação, isto é, quando unicamente participa a rotação axial do antebraço, é de aproximadamente 180° .

2.4 Análise Cinemática

Para realizar análises de movimentos de membros e articulações, é necessário representar os segmentos corporais. Essa representação pode seguir vários padrões. Os

mais utilizados recentemente são os propostos por [Cole et al. \(1993\)](#) e descritos do livro *Biomechanics of the Musculo-skeletal System* de [Nigg e Herzog \(2007\)](#).

Figura 2 – Imagens indicando como são realizados os movimentos de pronação e supinação do antebraço.



Fonte: Retirado de [Kapandji \(1987\)](#)

Seguindo esse padrão, os segmentos corporais são considerados como corpos rígidos, cada um dos segmentos relevantes para a análise em questão devem ser representados por um sistema Cartesiano de coordenadas local. Esse sistema de coordenadas é definido por uma matriz de rotação em relação à um sistema Cartesiano de coordenadas global (de referência). Com essa matriz de rotação é possível transformar um vetor de um sistema de coordenadas para o outro, e vice-versa.

Existem várias formas de se calcular essa matriz de rotação. A mais simples é a concatenação dos vetores normalizados que representam os eixos do sistema de coordenadas local ([KWON, 1998](#)). Outro método para o cálculo da matriz de rotação que fornece

melhores resultados é o métodos dos mínimos quadrados, desenvolvido por [Veldpaus, Woltring e Dortmans \(1988\)](#).

Para interpretar os dados desses sistemas de coordenadas na análise biomecânica, são utilizados representações paramétricas da matriz de rotação. As representações mais utilizadas são a de ângulos de *Euler/Cardan*, *Joint Coordinate System* (JCS, ou sistema de coordenadas articulares, em português) e ângulos helicoidais ([NIGG; HERZOG, 2007](#)).

Alguns softwares comerciais possuem módulos que realizam esses cálculos, como o Visual3D da [C-Motion \(1997\)](#) e até pacotes da [Qualisys AB \(1989\)](#). Porém, o custo de softwares como esses são muitas vezes proibitivos para seu uso.

2.4.1 Padronização da representação de corpos rígidos em 3D

Na análise 3D de articulações existem várias formas de se realizar o estudo. Os modelos citados na seção 2.4 acima são um grande avanço no caminho para a padronização, porém, cada uma das representações utilizadas ainda possui muitas variabilidades.

No caso dos ângulos de *Euler/Cardan*, por exemplo, existem doze modos diferentes de se representar as rotações do sistema de coordenadas. E a comparação entre análises realizadas com sequências de rotação distintas não são válidas. De forma similar, o sistema do JCS pode ser definido de inúmeras formas.

Numa tentativa de facilitar a comunicação entre pesquisadores e a comparação de resultados de artigos, o Comitê de Normalização e Terminologia (STC) da *International Society of Biomechanics* (ISB) propôs em [Wu et al. \(2005\)](#) padrões para a definição de JCS dos membros superiores¹. Até o momento da escrita desse trabalho, esse artigo já foi citado 1673 vezes, de acordo com informações do [Google Scholar \(2016\)](#) de 22 de Outubro de 2016 (aumento de 15 citações somente no último mês), o que indica uma alta adesão às recomendações feitas.

Essa proposta de normalização do estudo de movimentos articulares define pontos anatômicos e sistemas de coordenadas locais, além dos eixos para o cálculo do JCS e também as rotações que devem ser seguidas em cada articulação para definir os ângulos de *Euler/Cardan* equivalentes ao JCS.

Esses eixos e rotações foram escolhidos para que os resultados dessa análise sejam similares às definições clínicas. Apesar disso, diferenças são inevitáveis, já que às definições clínicas são feitas em duas dimensões e não são consistentes com as definições em 3D.

Ambas as representações por *Euler/Cardan* e pelo JCS possuem vantagens e desvantagens semelhantes ([NIGG; HERZOG, 2007](#)). Ambas conseguem representar os ângulos de maneira similar à definição clínica, ambas possuem várias formas de serem definidos

¹ Outro artigo foi publicado em 2002 e trata de membros inferiores, porém esses não são cobertos neste trabalho de conclusão de curso.

e necessitam de uma padronização e ambas sofrem de *Gimbal lock*, uma singularidade matemática que impossibilita o cálculo dos ângulos de rotação. Porém, não precisamos nos preocupar com o *Gimbal lock* neste caso, pois o único movimento será o de rotação do antebraço, que é realizado em torno do eixo longitudinal, que é o último eixo de rotação (o mais interno), portanto não há como ocorrer essa singularidade nessas condições.

Apesar de todas as semelhanças, o JCS possui a vantagem de ser mais intuitivo, pois conceitualmente, é uma representação mais natural do movimento das articulações, já que a sequência de rotações dos ângulos de *Euler/Cardan* sugere que o movimento deve ocorrer da posição inicial neutra, o que não é sempre o caso. Considerando isso, o trabalho será desenvolvido com o sistema JCS.

2.4.2 Joint Coordinate System (JCS)

Para o cálculo do JCS, foi utilizado o algoritmo de [Cole et al. \(1993\)](#). Ele é definido como rotações sobre os três eixos unitários do JCS:

- e1** Eixo fixo médio-lateral do sistema de coordenadas proximal. Esse é o eixo de rotação para flexão/extensão da articulação analisada;
- e3** Eixo fixo longitudinal do segmento distal. A rotação axial da articulação é medida em relação a esse eixo;
- e2** Eixo flutuante definido como produto vetorial entre os eixos **e3** e **e1**. Ele define o eixo de rotação para adução/abdução da articulação.

Esse algoritmo é uma forma generalizada para o cálculo em qualquer articulação do corpo. Ele foi primeiramente desenvolvido por [Grood e Suntay \(1983\)](#) para o caso específico da articulação do joelho. Para definir a versão genérica, são definidos novos marcadores para os eixos de cada sistema de coordenadas local:

- Eixo F** Eixo unitário de flexão/extensão, denominado **f**. Escolhido como o eixo do segmento local que orientado predominantemente na direção médio-lateral.
- Eixo L** Eixo unitário longitudinal, denominado **l**. Escolhido como o eixo do segmento orientado predominantemente na direção longitudinal do segmento.
- Eixo T** O terceiro eixo, denominado **t**, definido como o produto vetorial entre os outros dois, $\mathbf{t} = \mathbf{l} \times \mathbf{f}$.

Os três vetores são obtidos pela matriz de rotação. Se o método utilizado para seu cálculo for o [Kwon \(1998\)](#), basta atribuir dois dos vetores **x**, **y** e **z** aos vetores **f** e **l** correspondentes dessa nova nomenclatura. O eixo **t** pode então ser calculado.

Se o método escolhido for o [Veldpaus, Woltring e Dortmans \(1988\)](#), os vetores \mathbf{f} e \mathbf{l} são definidos como as linhas ou colunas da matriz de rotação que definem os eixos \mathbf{x} , \mathbf{y} ou \mathbf{z} , ou seja, \mathbf{f} e \mathbf{l} serão definidos como \mathbf{x} , \mathbf{y} ou \mathbf{z} . E então o eixo \mathbf{t} também pode ser calculado.

Considerando dois sistemas de coordenadas locais, um de referência, i , e outro como alvo, j , com os vetores unitários definidos em \mathbf{f} , \mathbf{l} e \mathbf{t} temos que as equações do JCS:

$$e_1 = f_i \quad (2.1)$$

$$e_3 = l_j \quad (2.2)$$

$$e_2 = \left(\frac{e_3 \times e_1}{|e_3 \cdot e_1|} \right) \cdot A \quad (2.3)$$

Onde:

$$A = \begin{cases} -1 & \text{se } (e_3 \times e_1) \cdot t_j < 0 \text{ e } ((e_3 \times e_1) \times e_3) \cdot f_j > 0 \\ +1 & \text{caso contrário} \end{cases}$$

Com essas definições é possível calcular os ângulos de rotação em cada um dos eixos para representar o segmento alvo em relação ao de referência. O ângulo ϕ_1 , [Equação 2.4](#), é o de rotação no eixo $\mathbf{e1}$ (ângulo de flexão/extensão), ele é definido como o ângulo entre T_i e o plano $T_j F_j$, medido no plano $L_i T_i$. O ângulo ϕ_2 , [Equação 2.5](#), é o de rotação no eixo $\mathbf{e2}$ (ângulo de adução/abdução), ele é definido como o ângulo entre \mathbf{r} e L_j , medido no plano $F_i L_j$. O ângulo ϕ_3 , [Equação 2.6](#), é o de rotação no eixo $\mathbf{e3}$ (ângulo de rotação axial), ele é definido como o ângulo entre T_j e o plano $T_i F_i$ medido no plano $F_j T_j$.

$$\phi_1 = \arccos(e_2 \cdot t_i) \cdot \text{sign}(e_2 \cdot l_i) \quad (2.4)$$

$$\phi_2 = \arccos(r \cdot l_j) \cdot \text{sign}(e_1 \cdot e_3) \quad (2.5)$$

$$\phi_3 = \arccos(e_2 \cdot t_j) \cdot \text{sign}(e_2 \cdot f_j) \quad (2.6)$$

Onde:

$$r = \left(\frac{e_1 \times e_2}{|e_1 \cdot e_2|} \right)$$

$$\text{sign}(x) = \begin{cases} +1 & \text{se } x \geq 0 \\ -1 & \text{se } x < 0 \end{cases}$$

Em relação ao cálculo com ângulos de *Cardan*, $\mathbf{e1}$ representa o primeiro eixo de rotação de *Cardan*, $\mathbf{e2}$ representa o segundo e $\mathbf{e3}$ o terceiro. Apesar dos ângulos serem calculados de formas diferentes, os princípios são os mesmos e o resultado é idêntico.

2.4.3 JCS do cotovelo

Segundo [Wu et al. \(2005\)](#), para que a descrição cinemática da articulação do cotovelo seja útil e prática, é necessário realizar algumas aproximações: a articulação glenoumeral é uma articulação esférica, as articulações úmero-ulnar e a radio-ulnar (proximal e distal) são articulações em dobradiça (gínglimo).

Outra aproximação feita é a de que os eixos do sistema de coordenadas coincidem com os eixos de rotação da articulação do cotovelo e radio-ulnar. Se houver interesse em se desenvolver um estudo detalhado dessas articulações, é necessário determinar a orientação desse eixo de rotação real. No caso desse trabalho, essa simplificação é permitida e será utilizada.

Para as definições à seguir, foi considerado o cálculo no membro direito. Se houver interesse em realizar estudo do membro esquerdo é necessário espelhar as coordenadas no eixo sagital ($z = -z$).

2.4.4 Sistemas de coordenadas local para a articulação do cotovelo

Para calcular o JCS da articulação do cotovelo, é preciso definir dois sistemas de coordenadas locais: um do úmero e outro do antebraço.

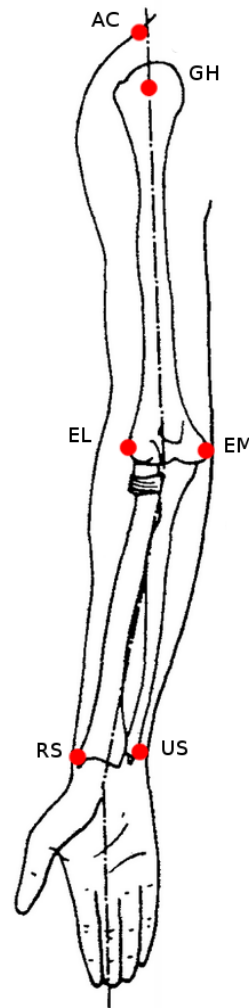
O sistema de coordenadas do úmero pode ser definido de duas formas diferentes. A primeira delas define o úmero somente com pontos anatômicos dele próprio, já a segunda utiliza os pontos do antebraço na definição. Essa segunda forma é preferível quando os pontos do antebraço estão disponíveis, pois ela é menos sensível a pequenos erros na posição dos marcadores ([WU et al., 2005](#), p. 984). Porém, é importante notar que para essa segunda definição ser precisa, o antebraço deve estar a 90° em relação ao úmero e completamente pronado. Como nesse trabalho o objetivo é o estudo da rotação do antebraço, a segunda definição será utilizada.

Para definir os dois sistemas de coordenadas, é necessário posicionar os marcadores em pontos anatômicos do corpo conforme descrito em [Wu et al. \(2005, p. 983-985\)](#). Esses marcos anatômicos estão representados na [Figura 3](#) e descritos abaixo.

Pontos anatômicos:

GH	Centro de rotação glenoumeral, estimado por regressão linear ou por registro de movimentos.
AC	Ponto mais dorsal na articulação acromioclavicular.
EL	Ponto mais caudal no epicôndilo lateral.
EM	Ponto mais caudal no epicôndilo medial.
RS	Ponto mais caudal-lateral no processo estilóide do rádio.
US	Ponto mais caudal-medial no processo estilóide da ulna.

Figura 3 – Localização dos pontos anatômicos.



Fonte: Retirado de Wu et al. (2005)

Assim, os eixos do sistema de coordenadas do antebraço é representado graficamente na Figura 4 e definido como (WU et al., 2005, p. 984-985):

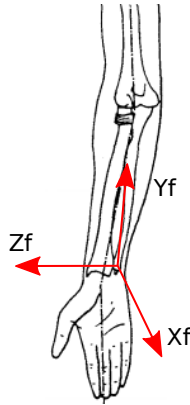
- O_f : Origem coincidente com US.
- Y_f : A linha conectando US e o ponto médio entre EL e EM, apontando proximalmente.
- X_f : A linha perpendicular ao plano formado pelos pontos US, RS e o ponto médio entre EL e EM, apontando para frente.
- Z_f : A linha perpendicular a X_f e Y_f , apontando para a direita.

E os eixos do sistema de coordenadas do úmero é representado graficamente na Figura 5 e definido como (WU et al., 2005, p. 984):

- O_h : Origem coincidente com GH.

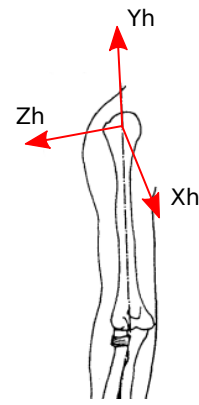
- Y_h : A linha conectando GH e o ponto médio entre EL e EM, apontando para GH.
 Z_h : A linha perpendicular ao plano formado por Y_h e Y_f , apontando para a direita.
 X_h : A linha perpendicular a Z_h e Y_h , apontando para frente.

Figura 4 – Sistema de coordenadas locais do antebraço.



Fonte: Imagem de autoria própria

Figura 5 – Sistema de coordenadas locais do úmero.



Fonte: Imagem de autoria própria

Com esses sistemas de coordenadas locais, é possível definir os eixos do JCS para o cotovelo ($e1$, $e2$ e $e3$) de acordo com Wu et al. (2005, p. 987). O movimento de interesse é o do antebraço em relação ao úmero e a ordem de rotações dos ângulos de *Cardan* correspondente é Z-X-Y.

- e1:** O eixo fixo no segmento proximal e coincidente com o eixo Z_h do sistema de coordenadas do úmero.
Rotação: flexão (positivo) e hiperextensão (negativo).
- e3:** O eixo fixo no segmento distal e coincidente com o eixo Y_f do sistema de coordenadas do antebraço.
Rotação: rotação axial do antebraço, pronação (positivo) e supinação (negativo).
- e2:** O eixo flutuante, perpendicular a $e1$ e $e3$. Ele é o eixo X_f rotacionado.
Rotação: ângulo valgo (*carrying angle*), esse é o ângulo entre o eixo longitudinal do antebraço e o plano perpendicular ao eixo de flexão/extensão.

2.5 Álgebra vetorial

A álgebra vetorial é a área da matemática que realiza operações com vetores. Nesse trabalho, todas as operações realizadas serão no espaço tri-dimensional (X, Y, Z). As definições se baseiam em Venturi (1949).

Um vetor saindo de um ponto, i , e apontando para outro, j , é definido como a diferença entre o destino e a origem: $\mathbf{u} = (x_j - x_i, y_j - y_i, z_j - z_i)$.

Pontos em um sistema de coordenadas podem ser pensados como um caso específico dessa diferença entre pontos, onde o ponto i é a origem do sistema. Assim, eles são definidos como um vetor apontando da origem para a posição do ponto e definido como a tripla $\mathbf{v} = (x, y, z)$.

Definimos a operação $\mathbf{v} \cdot \mathbf{w}$ como produto escalar entre dois vetores. Esse produto resulta em um número escalar tal que:

$$\mathbf{v} \cdot \mathbf{w} = |\mathbf{v}||\mathbf{w}| \cos \theta$$

Onde θ é a medida do ângulo entre os vetores, com $\theta \in [0^\circ, 180^\circ]$. Assim, se os vetores forem unitários, isso é, $|\mathbf{v}| = |\mathbf{w}| = 1$, podemos calcular θ com:

$$\theta = \arccos(\mathbf{v} \cdot \mathbf{w})$$

Para transformar um vetor qualquer em um vetor unitário, devemos normalizá-lo. O módulo de um vetor \mathbf{k} é definido como $|\mathbf{k}| = \sqrt{x^2 + y^2 + z^2}$. A transformação para vetor unitário é feita com:

$$\mathbf{v} = \frac{\mathbf{k}}{|\mathbf{k}|}$$

onde \mathbf{v} é o vetor unitário.

Definimos a operação $\mathbf{v} \times \mathbf{w}$ como produto vetorial entre dois vetores. Esse produto resulta em um outro vetor tal que a direção desse vetor é perpendicular a \mathbf{v} e \mathbf{w} , com o sentido em que \mathbf{v} , \mathbf{w} e $\mathbf{v} \times \mathbf{w}$, nessa ordem, formam um triedro positivo. O módulo é definido por:

$$|\mathbf{v} \times \mathbf{w}| = |\mathbf{v}||\mathbf{w}| \sin \theta$$

Essas definições são utilizadas nos cálculos da [subseção 2.4.2](#) e na implementação do algoritmo da [seção 3.3](#).

3 DESENVOLVIMENTO

3.1 Sistema de captura de movimento

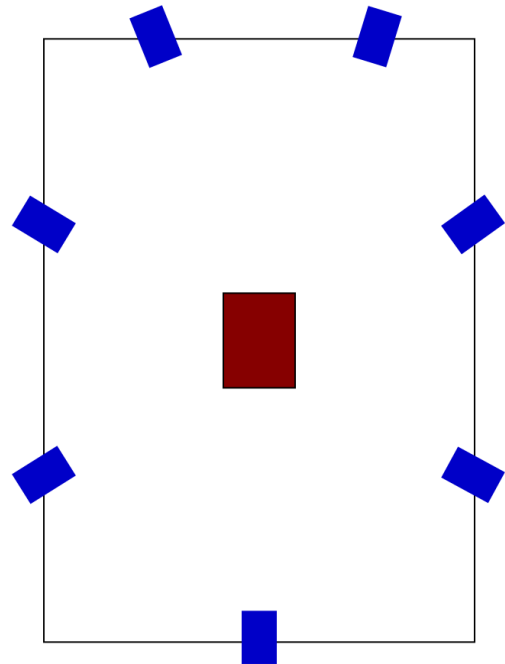
Para realizar a medida dos ângulos de pronação e supinação, foi utilizado o sistema óptico de captura de movimento da [Qualisys AB \(1989\)](#), composto por sete câmeras Oqus 300 ([Figura 6](#)), o software *Qualisys Track Manager* (QTM) e cinco marcadores passivos.

Figura 6 – Foto da câmera Oqus 300 utilizada no experimento.



Fonte: Imagem de autoria própria

Figura 7 – Representação do posicionamento das câmeras (retângulos azuis). O retângulo vermelho representa a posição do volume de captura.

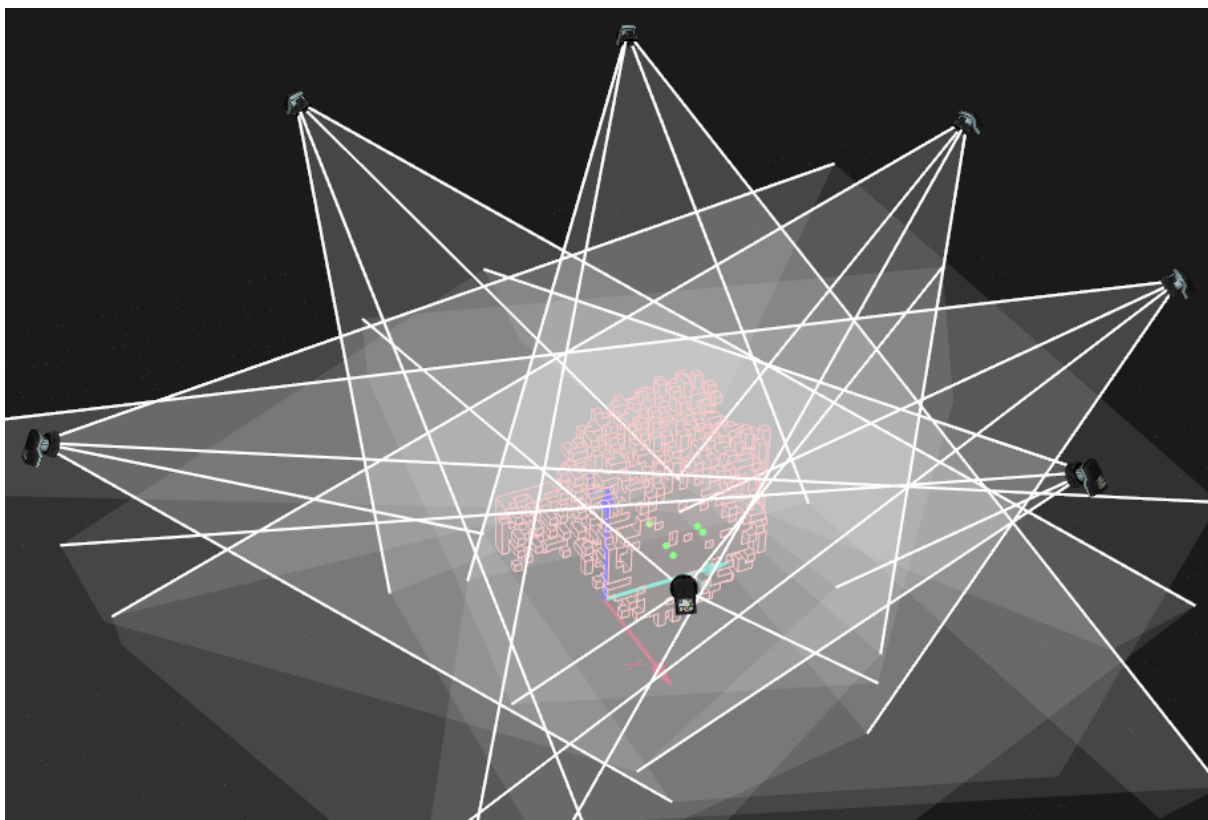


Fonte: Imagem de autoria própria

As câmeras estavam instaladas no laboratório de avaliação cinemática do Departamento de Fisioterapia da Universidade Federal de São Carlos (UFSCar). Elas foram posicionadas igualmente espaçadas uma da outra em um entorno de 360° (aproximadamente 50° entre elas), de acordo com as instruções do manual: o ângulo entre cada duas câmeras deve ser maior ou igual a 30°, porém ainda é menor do que o ângulo ideal de 60°. Um esquemático representando a sala pode ser visto na [Figura 7](#).

A [Figura 8](#) mostra a interpretação do software QTM sobre o posicionamento das câmeras e seus ângulos de visão.

Figura 8 – Imagem gerada pelo software QTM indicando o cone de visão de cada uma das sete câmeras.



Fonte: Imagem retirada do software QTM

Esse ângulo é importante para realizar o cálculo da posição 3D de marcadores, já que é necessário pelo menos duas câmeras para estimar essa terceira dimensão. Se o ângulo entre elas for muito pequeno, essa estimativa perde precisão. Como o ângulo entre elas é menor do que o de 60° recomendado, o ideal é que pelo menos 3 câmeras vejam cada marcador. Como foram utilizadas 7 câmeras, os marcadores tiveram uma cobertura suficiente.

Além disso, as câmeras foram instaladas no alto da sala para terem melhor visualização dos marcadores e para que tenham um ângulo de pelo menos 20° com o plano horizontal. Isso se faz necessário para que uma câmera não enxergue a luz infravermelha emitida pela outra e confunda isso com um marcador refletindo a luz.

Também é importante evitar reflexões indesejadas, como de superfícies metálicas ou vidros, por exemplo. Para isso, o laboratório possui chão e paredes pintados de preto e o sujeito sendo analisado deve retirar qualquer anel, corrente, relógio, brinco ou outro objeto do corpo que possa refletir luz.

3.1.1 Marcadores

Entre as opções de marcadores (ativo ou passivo), foram utilizados os passivos para a captura, pela facilidade de manuseio e pelo experimento ser realizado em uma sala especialmente preparada para a coleta, como dito anteriormente. Cada marcador tinha um diâmetro de 16 milímetros e foram posicionados de acordo com a [seção 3.2](#). Um exemplo de marcador utilizado pode ser visto na [Figura 9](#).

Figura 9 – Foto de um dos marcadores utilizados na captura.



Fonte: Imagem de autoria própria

As câmeras possuem LEDs (*Light-Emitting Diode*, em inglês) infravermelhos para emitir a luz que será refletida pelos marcadores. Com essa luz estroboscópica, os marcadores podem ficar posicionados até 25 metros de distância das câmeras. Na sala utilizada, a maior distância entre os marcadores e qualquer uma das câmeras é de 5 metros.

3.1.2 Quantidade de câmeras

Como a análise a ser feita só envolve cinco marcadores posicionados no membro superior direito, em teoria não seriam necessárias as sete câmeras que foram utilizadas. A princípio, quatro câmeras seriam suficientes para que cada marcador fosse visto por pelo menos duas câmeras com um ângulo de mais de 60° entre elas.

Já que o laboratório utilizado estava configurado com as sete câmeras, todas elas foram utilizadas, o que garante um resultado melhor ou igual ao que seria obtido com uma menor quantidade de equipamentos.

3.1.3 Comunicação do sistema

As câmeras são conectadas por cabos que transmitem dados e energia e são ligadas a um computador central com um conector do tipo RJ-45. Esse sistema forma uma topologia de rede em linha, onde cada câmera é conectada a uma anterior e uma próxima, com exceção da primeira e última câmera do barramento, que só são conectadas ou à próxima ou à anterior.

A comunicação é realizada através do protocolo TCP/IP. Para configurar o sistema, o computador possui o software QDS (*Qualisys DHCP Server*), que é responsável por criar um servidor de DHCP (*Dynamic Host Configuration Protocol*, em inglês) que irá atribuir os endereços IPs (*Internet Protocol*, em inglês) a cada uma das câmeras. Isso permite a troca de mensagens entre o *host* (computador) e os *peers* (câmeras).

Nessa arquitetura, o *host* envia comandos para os *peers* controlando o status, modo de operação e configuração de parâmetros, entre outros. Em contrapartida, as câmeras respondem às informações requisitadas e enviam os quadros captados ou os marcadores processados, dependendo do modo em que se encontram.

3.1.4 Linearização das câmeras

Toda câmera possui distorções na imagem que precisam ser compensadas para possibilitar o cálculo preciso de coordenadas 2D de um objeto em seu campo de visão. No caso do sistema da *Qualisys*, os arquivos com os parâmetros de linearização de cada câmera são fornecidos pela fabricante para facilitar a utilização pelo usuário final, já que, uma vez calculado, esses parâmetros não devem mudar para um determinado dispositivo. Assim, só é necessário carregar essas linearizações no QTM. O parâmetro principal desse arquivo é a distância focal de cada câmera.

3.1.5 Captura das imagens

As câmeras de captura de movimento da série Oqus 300 possuem três modos de captura de imagens: marcador, intensidade do marcador e vídeo. Dentre eles, a maior diferença está entre os dois primeiros e o terceiro: no terceiro a imagem captada pela câmera não é processada, e nos outros dois a câmera processa a imagem para identificar os marcadores e armazena ou envia esses dados dos marcadores ao invés da imagem.

O primeiro modo é o *Immediate*, quando ele está selecionado as câmeras capturam a imagem, realizam o processamento e armazenam esses dados em seu *buffer* interno ao mesmo tempo que em enviam eles para o sistema QTM.

O segundo modo é o *Frame buffering*, neste modo as câmeras também capturam a imagem, processam e armazenam em seu *buffer* interno, porém elas não transmitem os dados para o QTM de imediato. A transferência só é realizada quando a captura completa

termina. Neste momento ela irá transferir os dados de todos os quadros da captura de uma só vez.

No segundo modo, as câmeras possuem mais tempo para processar os dados, o que pode melhorar a qualidade do processamento (somente em câmeras *ProReflex*), já que elas não gastam tempo enviando os dados. Porém, a captura tem um limite de tempo restrito devido ao tamanho do *buffer* interno. Esse tempo irá depender da resolução das câmeras ou da quantidade de marcadores, dependendo se foram configuradas no modo *marcador* ou *vídeo*, respectivamente.

Neste trabalho, foi utilizada a captura no modo *Immediate*, que é o padrão do sistema, com imagens em 4.0 mega-pixeis de resolução (1280x1024 pixeis) e frequência de aquisição de 120 Hz.

Alguns outros parâmetros são importantes para a identificação dos marcadores e devem ser ajustados de acordo com o ambiente em que a captura está sendo realizada. Os principais parâmetros são: tempo de exposição da câmera, limiar dos marcadores, filtro de circularidade dos marcadores e limites dos marcadores.

O tempo de exposição define por quanto tempo a câmera irá capturar cada imagem, isso irá interferir na quantidade de luz que será absorvida pelas lentes e, portanto, quanto maior a exposição maior será a intensidade dos marcadores visíveis. Porém, uma exposição muito alta pode fazer com que marcadores muito próximos sejam percebidos como um só marcador. O ideal é que cada marcador seja identificado com um tamanho de pelo menos 200 subpixeis¹. Esse valor pode ser observado em tempo real pela interface do QTM, na visualização 2D dos marcadores. O tempo de exposição padrão é 400 μ s e foi utilizado o valor de 300 μ s para a captura.

O limiar dos marcadores define qual o mínimo valor de intensidade de um pixel para que ele seja reconhecido como um marcador. Um valor muito baixo tem o problema de identificar pequenos reflexos indesejados como marcadores, e um valor muito alto tem o problema de ignorar marcadores que não tinham uma intensidade tão forte. O valor padrão para o limiar é 17% e foi utilizado o valor de 20% nas capturas.

O filtro de circularidade dos marcadores define o quão circular um marcador identificado deve ser (relação entre o tamanho no eixo x e y) para ser considerado válido. Se a diferença entre as dimensões x e y forem maiores do que as especificadas nesse filtro, o marcador será ignorado. Aumentar o nível de circularidade ajuda o sistema a considerar dados de maior qualidade, porém um nível muito alto fará com que muitos marcadores sejam desconsiderados e, assim, haverão buracos na trajetória dos marcadores. O padrão é

¹ Subpixel é a unidade de medida do tamanho de um objeto como visto pela câmera. Ela não representa o tamanho real do objeto em milímetros, mas sim o tamanho do objeto relativo a distância dele até a câmera. Para converter de pixel para subpixel, é feita a multiplicação por 64.

não realizar essa filtragem.

Os limites dos marcadores definem o menor e maior tamanho de um marcador, em subpixels, para que seja considerado válido, além do número máximo de marcadores que podem ser identificados. O principal objetivo dessa configuração é remover pequenas reflexões que estavam sendo vistas como marcadores. Além disso, se o número máximo de marcadores for menor do que o número presente, os marcadores da parte debaixo da imagem serão descartados, porque o processamento deles é feito de cima para baixo. O padrão desses limites é desativado, considerando todos os marcadores válidos.

3.1.6 Calibração

Após posicionar as câmeras, conectar o computador central e configurar todos os parâmetros, é necessário realizar uma calibração do sistema. Essa calibração é necessária para fornecer a posição e orientação de cada câmera para o software QTM. Com o posicionamento de cada câmera definido, o QTM pode realizar os cálculos com os dados 2D e estimar os dados em 3D.

Antes de realizar a calibração, é necessário definir qual será o volume de medição, que é o volume no qual o movimento a ser capturado ocorrerá. Como neste caso o sujeito estará sentado e o único movimento será a rotação axial do antebraço, o volume foi definido como uma caixa que engloba uma cadeira no centro da sala, com 2 metros de altura, 1 metro de largura e 1 metro de comprimento.

Existem três métodos possíveis para realizar a calibração, porém só iremos discutir um deles, que é a calibração por varinha. Com essa calibração, duas estruturas são utilizadas: uma varinha com formato de T e um suporte de referência em formato de L .

A varinha possui dois marcadores posicionados em cada uma das extremidades superiores, conforme [Figura 10](#), e é utilizada para realizar os movimentos de calibração. O suporte possui três marcadores posicionados conforme a [Figura 11](#).

A posição do suporte define o sistema de coordenadas global para a captura, com a haste maior representando a direção do eixo x e a menor a direção do eixo y . Assim, é importante posicionar a estrutura de acordo com o movimento que deseja ser analisado para facilitar o processamento dos dados.

Com a estrutura de referência posicionada dentro do volume de medição, deve-se realizar movimentos com a varinha de forma que as câmeras enxerguem os dois marcadores no maior número possível de orientações diferentes. Um dos modos de realizar esse movimento é posicionar a estrutura em T na horizontal e girá-la enquanto a move por todo o volume, garantindo que o movimento é realizado no maior número de direções diferentes possível. Para que nenhuma câmera seja obstruída durante toda a calibração pelo corpo da pessoa realizando o movimento, o operador deve mover-se rodeando o volume de calibração

enquanto realiza o movimento com a varinha. Dessa forma, cada câmera só terá sua visão obstruída por um curto intervalo de tempo.

Figura 10 – Varinha utilizada na calibração (750 mm).



Fonte: Imagem de autoria própria

Figura 11 – Suporte de referência (compatível com 750 mm).



Fonte: Imagem de autoria própria

Para mais informações sobre como realizar a calibração, consultar o manual do QTM ([QUALISYS AB, 2013](#)).

3.1.7 Processamento dos dados

Com os marcadores capturados pelas câmeras, em 2D, e os dados adquiridos pela calibração, o software QTM irá processar o conjunto de dados e estimar a posição e a trajetória dos marcadores em 3D. Para auxiliar os cálculos, ele armazena um pequeno *buffer* (de 4 a 10 quadros) de posições anteriores para tentar prever qual será a próxima posição do marcador. Esse algoritmo utiliza três parâmetros para ajustarmos os cálculos: erro de predição, residual máximo e tamanho mínimo da trajetória.

O erro de predição define a maior distância (em milímetros) permitida entre a posição prevista pelo algoritmo e a posição calculada por aquele instante da captura. Se o erro for maior, o ponto em questão não será considerado como parte da trajetória. Para esse parâmetro foi utilizado o valor padrão de 30 mm.

O residual máximo auxilia determinar quais marcadores 2D entre as diferentes câmeras representam o mesmo marcador 3D. Quando a posição 3D é calculada, o residual máximo define a maior distância (em milímetros) que um raio do marcador 2D será considerado como pertencente a esse ponto 3D. Em outras palavras, ele define o quanto a intersecção dos marcadores 2D podem estar distantes uma da outra e ainda serem consideradas o mesmo ponto. O valor padrão para esse parâmetro é 10 mm, mas pode ser definido como o 2-5 vezes o valor médio do residual mostrado no resultado da calibração das câmeras.

O tamanho mínimo da trajetória define, como o próprio nome diz, a menor quantidade de quadros necessária para formar uma trajetória. O valor padrão é 2 quadros.

3.2 Processo de coleta dos dados

Para realizar a coleta dos dados, o sistema de captura de movimento foi configurado e calibrado de acordo com os parâmetros discutidos na [seção 3.1](#). Com o sistema pronto, uma fisioterapeuta foi responsável pelo posicionamento dos marcadores em um voluntário, que foi posicionado no centro da sala.

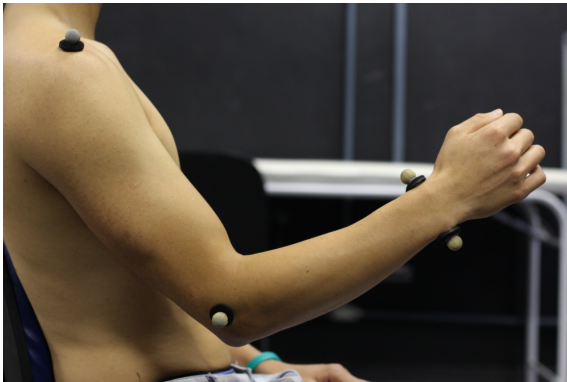
Como descrito na [subseção 2.4.3](#), as definições de [Wu et al. \(2005\)](#) são para o cotovelo direito, assim, os testes foram realizados somente neste lado. Como não foi realizada nenhuma coleta para o lado esquerdo, o algoritmo não pôde ser testado nesse caso, portanto só há suporte para a análise do lado direito nessa versão.

O voluntário foi posicionado de acordo com o procedimento descrito na [seção 2.3](#): sentado, com o cotovelo flexionado a 90° e rotação neutra. Alguns ajustes precisaram ser feitos a essa posição para que as câmeras visualizassem o marcador EM: o ombro precisou ser flexionado em 15°-20° para frente.

Os pontos para posicionar os marcadores foram definidos de acordo com a [seção 2.4](#). Porém, foi feita uma adaptação no ponto GH. Como ele não é um ponto anatômico e seria necessário realizar uma regressão linear ou utilizar o método IHA ([WU et al., 2005](#), p. 983) para definir sua posição, foi utilizado o acrômio (AC) como uma aproximação de GH. Essa aproximação é considerada válida para esse caso em específico, já que o ombro estará em posição de flexão, com rotação neutra e parado. Importante destacar que os pontos RS e US devem ser posicionados de forma que fiquem alinhados na vertical com o antebraço posicionado em rotação axial neutra. Os marcadores posicionados no voluntário podem ser vistos nas Figuras [12](#), [13](#), [14](#) e [15](#).

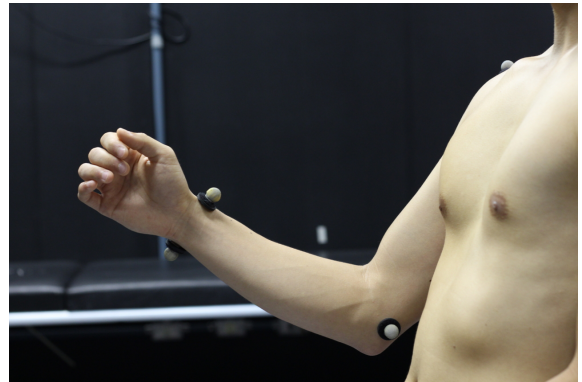
O voluntário deve, preferencialmente, estar sem camiseta, para que os marcadores fiquem bem fixados nos pontos anatômicos. Caso não seja possível, pelo menos o braço e ombro devem estar descobertos.

Figura 12 – Marcadores posicionados, imagem lateral direita.



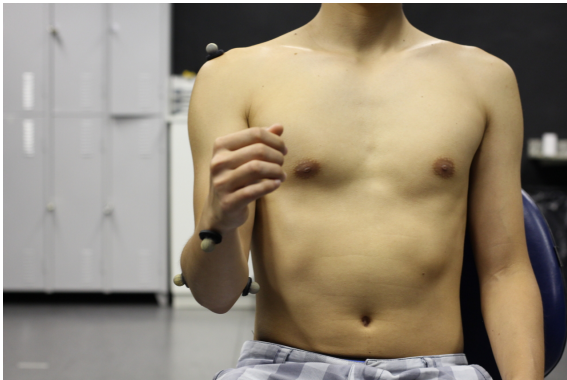
Fonte: Imagem de autoria própria

Figura 13 – Marcadores posicionados, imagem lateral esquerda.



Fonte: Imagem de autoria própria

Figura 14 – Marcadores posicionados, imagem frontal.



Fonte: Imagem de autoria própria

Figura 15 – Marcadores posicionados, imagem traseira.



Fonte: Imagem de autoria própria

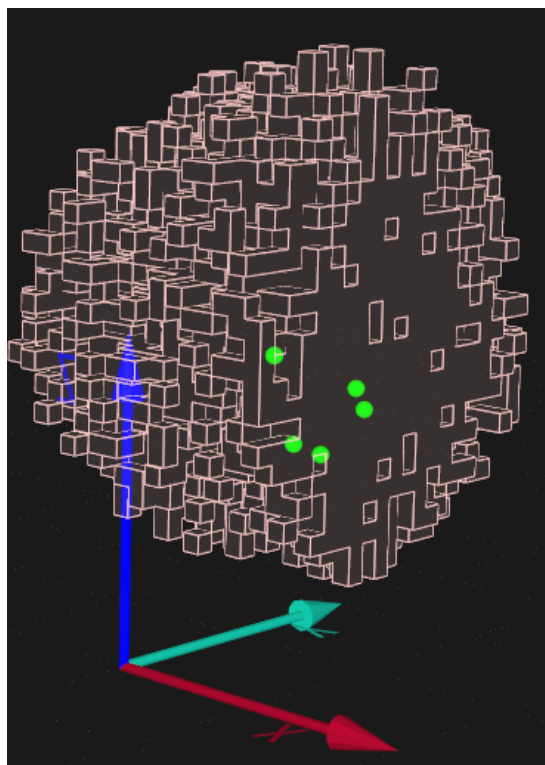
Para um posicionamento mais adequado, recomenda-se utilizar um apoio para o cotovelo do voluntário, o que reduz o esforço necessário para segurar o braço na posição ideal de medição e, com isso, evita o cansaço e eventual erro de medida.

Com o voluntário preparado e posicionado, é necessário verificar se todos os marcadores estão contidos no interior do volume calibrado, que pode ser verificado pelo software QTM na visualização 3D, como na [Figura 16](#). Após isso, a captura pode ser iniciada pelo próprio software.

A captura deve ser iniciada com o cotovelo em rotação neutra. O voluntário deve manter essa posição por cinco segundos para definir o ponto neutro. Após os cinco segundos, ele deve realizar o movimento de supinação seguido do de pronação e repetir esses movimentos em sequência mais duas vezes e por fim parar na posição neutra novamente. Ao finalizar a sequência de movimentos, a captura é encerrada no software. Cada movimento

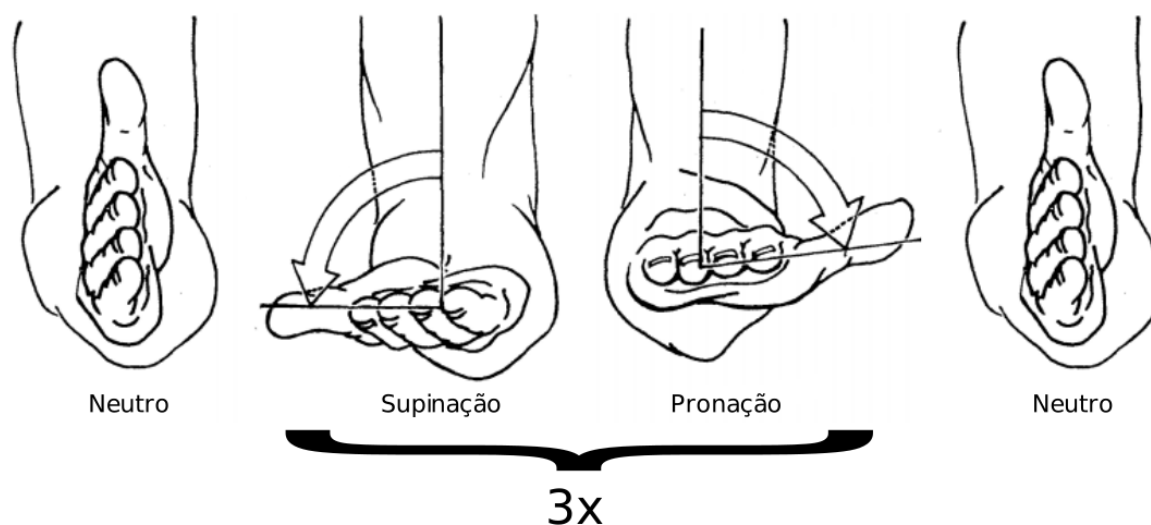
de supinação seguido de pronação deve ter uma duração de pelo menos 2 segundos. A sequência de movimentos é ilustrada na [Figura 17](#).

Figura 16 – Imagem do software QTM mostrando os marcadores (círculos verdes) dentro do volume de calibração. A origem do eixo de coordenadas global é indicada pelas setas na imagem.



Fonte: Imagem retirada do software QTM

Figura 17 – Ilustração da sequência de movimentos a ser realizada.



Fonte: Adaptação de imagem do [Kapandji \(1987\)](#)

Com a captura encerrada, é possível verificar o *replay* da coleta para garantir que os movimentos foram realizados de maneira correta. Isso pode ser feito na visualização 3D do QTM, clicando no botão de *play*. As coletas realizadas nesse trabalho tiveram uma duração entre 22-35 segundos, dependendo da velocidade com que o voluntário realizava os movimentos.

Após verificar que a coleta foi realizada com sucesso, os cinco marcadores devem ser nomeados para que possam ser exportados e processados pelo algoritmo posteriormente. O nome dos marcadores deve seguir a nomenclatura descrita na [seção 2.4](#), com o ponto GH trocado por AC.

Se algum outro marcador foi registrado pelo sistema de captura de movimentos, ele deve ser descartado no software QTM para não ser incluído no arquivo exportado. Também deve ser verificado o *Fill Level* de cada marcador na aba *Labeled trajectories*. Se algum deles possuir um valor inferior a 100%, recomenda-se repetir a coleta garantindo que pelo menos duas câmeras, posicionadas com um ângulo maior do que 30° entre elas, consigam ver cada marcador.

Com os marcadores nomeados, a captura pode ser exportada como um arquivo TSV. As configurações devem seguir as definições: exportar dados 3D, sem dados analógicos nem dados de força, incluindo o cabeçalho do TSV, excluindo trajetórias não identificadas e trajetórias vazias. Um exemplo do arquivo gerado pode ser visto na [Figura 18](#), as 10 primeiras linhas representam o cabeçalho e as três linhas seguintes são os três primeiros quadros da captura (dos 2806 presentes nessa captura) com o posicionamento 3D de cada marcador nesse instante de tempo.

Figura 18 – Exemplo de um arquivo TSV visualizado em um editor de planilhas.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	NO_OF_FRAMES	2808													
2	NO_OF_CAMERAS	8													
3	NO_OF_MARKERS	5													
4	FREQUENCY	120													
5	NO_OF_ANALOG	0													
6	ANALOG_FREQUENCY	0													
7	DESCRIPTION	--													
8	TIME_STAMP	2016-10-21, 14:49:28	921.3608												
9	DATA_INCLUDED	3D													
10	MARKER_NAMES	AC	EM	EL	US	RS									
11	501.896	49.824	992.076	434.282	222.532	706.104	529.289	226.735	691.373	489.94	465.522	781.539	495.599	418.396	849.914
12	501.884	49.805	992.115	434.112	222.603	706.203	529.268	226.749	691.427	489.916	465.483	781.534	495.556	418.384	849.936
13	501.881	49.811	992.122	434.196	222.59	706.211	529.242	226.795	691.416	489.879	465.516	781.586	495.517	418.39	849.953

Fonte: *Printscreen* de autoria própria

3.3 Algoritmo para cálculo da ADM

Para processar os dados do arquivo exportado do QTM e calcular a ADM, foi desenvolvido um algoritmo na linguagem de programação Python (versão 3.5). Ela é uma linguagem *script* com licença *open source* (código aberto), o que significa que pode ser utilizada por qualquer pessoa ou instituição, livre de custos ([PYTHON SOFTWARE](#)

FOUNDATION, 2016). Essa linguagem foi escolhida pela sua facilidade de manipulação de vetores e matrizes, graças ao pacote para computação científica NumPy (2016), que fornece funções similares as de softwares comerciais como o Matlab. A linguagem Python possui diversas implementações, a utilizada foi a CPython.

O desenvolvimento do algoritmo foi baseado em Cole et al. (1993), descrito na subseção 2.4.2. Sua implementação pode ser vista no Apêndice A e foi dividida em etapas:

1. Na primeira parte, o arquivo é carregado e o cabeçalho processado. Esse processamento irá identificar o número de marcadores presentes, a frequência utilizada na captura (Hz) e o nome e ordem dos marcadores. Caso haja alguma inconsistência entre os marcadores do arquivo e os definidos nesse trabalho, o algoritmo emitirá um erro informando o problema. Linhas 39-73 do código.
 - a) A ordem dos marcadores no arquivo não importa, pois o algoritmo reconhece a ordem automaticamente. Só é preciso garantir que existem cinco, com os nomes AC, EL, EM, RS e US.
 - b) A frequência da captura é utilizada para gerar o eixo do tempo no gráfico do resultado.
2. Na segunda parte, o algoritmo processa todo o restante do arquivo e armazena os valores da posição 3D de cada marcador (X, Y e Z) em memória. Cada tripla de coordenadas (X, Y, Z) é armazenada como um vetor no formato da biblioteca NumPy (NumPy array). Depois, cada vetor de posição desse é adicionado à uma lista contendo todos os vetores dos marcadores, onde cada posição na lista representa um instante de tempo. Linhas 75-89 do código.
3. Após ter as informações na memória, na terceira etapa são realizadas as operações vetoriais com as posições 3D de acordo com o algoritmo descrito na subseção 2.4.2 (todos os vetores no algoritmo são definidos como vetores unitários):
 - a) Primeiramente o ponto médio entre os marcadores EL e EM é calculado. Linha 104 do código.
 - b) Então os sistemas de coordenadas locais do antebraço e do úmero são definidos. Linhas 106-117 do código.
 - c) Em seguida são definidos os vetores \mathbf{f} e \mathbf{l} e calculado o vetor \mathbf{t} . Linhas 119-121 do código.
 - d) Com isso, os eixos $\mathbf{e1}$ e $\mathbf{e3}$ do JCS são definidos e o eixo $\mathbf{e2}$ é calculado. Linhas 123-125 do código.
 - e) A matriz \mathbf{A} utilizada na Equação 2.3 é implementada como uma condição *if* que troca o sinal de $\mathbf{e2}$ para o caso da condição testada em \mathbf{A} ser verdadeira. Linhas 127-129 do código.

- f) O ângulo de rotação axial já pode ser calculado seguindo a [Equação 2.6](#). Linhas [135-137](#) do código.
- g) Esse ângulo é adicionado em uma lista com os ângulos de rotação axial do cotovelo. Linha [140](#) do código.
4. Com a frequência de captura extraída do cabeçalho do arquivo, é gerada uma lista com os tempos de cada amostra. Essa lista definirá o eixo x do gráfico do resultado. Linhas [146-147](#) do código.
5. Com essa lista de ângulos, os picos de pronação (positiva) e supinação (negativa) são identificados com auxílio do pacote *PeakUtils* desenvolvido por [Negri \(2016\)](#). Então a média dos picos é calculada para gerar o valor da ADM de pronação e de supinação. Linhas [149-163](#) do código.
6. Por fim, para visualizar o movimento de rotação axial (ADM) realizado, é gerado um gráfico com a lista do tempo no eixo x e a lista dos ângulos no eixo y . Além de imprimir também o resultado da ADM de pronação, supinação e total. Esse gráfico é gerado com o pacote *matplotlib* do Python. Linhas [174-208](#) do código.

É importante destacar que foi realizada uma modificação nas definições dos vetores \mathbf{f} , \mathbf{l} e \mathbf{t} . O algoritmo foi definido de forma a se obter um ângulo de 0° na posição neutra do movimento. Porém, como a posição neutra do algoritmo genérico de [Cole et al. \(1993\)](#) é considerada como a posição anatômica, os ângulos da rotação axial variam de 0° (pronação) a -180° (supinação), o que não é desejado.

Para termos o ângulo de 0° na posição neutra de rotação axial, devemos considerar a definição do movimento de flexão². Na posição neutra para medição de ADM de pronação/supinação, o antebraço está alinhado com plano sagital, assim, o movimento que seria de adução do punho, se torna flexão nessa posição. Com isso, o eixo \mathbf{f}_j do antebraço será definido como o eixo X_f do antebraço (e não o padrão que seria defini-lo como Z_f). Em outras palavras, estamos considerando o eixo X_f como o eixo de flexão e o eixo Z_f como eixo de rotação.

Com isso, teremos 0° de rotação na posição neutra, ângulos positivos para pronação e ângulos negativos para supinação, como definido em [Wu et al. \(2005, p. 987\)](#).

Esse algoritmo possui a limitação de só poder ser aplicado para análises do cotovelo direito. Embora a adaptação para análise do membro esquerdo não seja difícil, é necessário realizar outras coletas com os marcadores posicionados no lado esquerdo para ser possível testar e validar o algoritmo.

² Flexão é o movimento que ocorre em um plano paralelo ao plano sagital.

4 RESULTADOS E DISCUSSÃO

4.1 Validação do sistema de captura de movimento

Antes de realizar as capturas, foram realizados dois testes para verificar a precisão do sistema de captura de movimento. Um teste estático e outro dinâmico:

1. O teste estático consistiu em posicionar um ponto fixo dentro do volume calibrado e medir sua posição 3D para analisar a variação e o ruído do sistema.
2. O teste dinâmico foi realizado com uma varinha de calibração, porém menor do que a utilizada na calibração do sistema, movendo-a dentro do volume de forma aleatória. O objetivo desse teste é medir a precisão (média e desvio padrão) da distância calculada entre os dois marcadores da varinha.

No primeiro teste, foi realizada a análise das posições 3D do marcador estático no Matlab. Essa análise mostrou que o desvio padrão nas coordenadas X, Y e Z foram 0.066 mm, 0.040 mm e 0.048 mm, respectivamente. Assim, observamos uma precisão melhor do que 1 mm, o que é suficiente para nosso experimento.

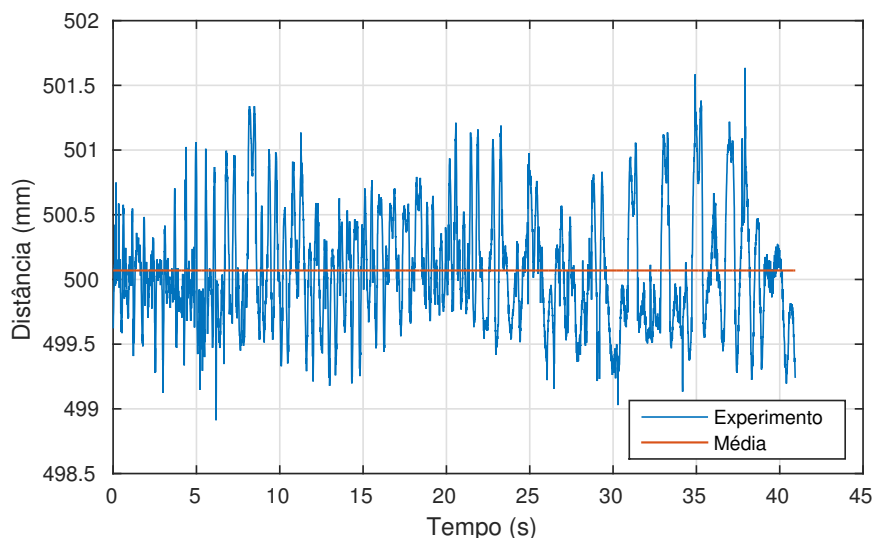
O segundo teste teve duração de aproximadamente 41 segundos, a distância entre os dois marcadores foi calculada pelo software QTM e esse valor exportado para o Matlab para análise, [Figura 19](#). A distância reportada entre os marcadores pela fabricante é de 500.6 mm e a média encontrada pelo cálculo no QTM foi 500.07 mm, com desvio padrão de 0.44 mm. A distância máxima calculada foi 501.63 mm e a mínima foi 498.91 mm. Com esses resultados, obtemos que os piores erros identificados foram menores do que 2 mm (menos de 0.4% de erro), e o desvio padrão é menor do que 0.5 mm.

Com os dois resultados anteriores temos segurança de realizar o experimento com esse sistema de captura de movimentos, já que erros nessa ordem de grandeza não devem invalidar os resultados obtidos.

4.2 Software em Python

Por ser desenvolvido em Python, o software pode ser utilizado em Windows, Mac OS ou Linux, além de ser livre de custos. Para executá-lo, é necessário ter o Python 3.3+ instalado, assim como os pacotes mencionados na [seção 3.3](#): NumPy, Matplotlib e PeakUtils. Uma sugestão para facilitar a instalação dos requisitos é instalar a distribuição de Python *Anaconda*, pois ela já possui o Python 3.5, NumPy e Matplotlib inclusos.

Figura 19 – Gráfico da distância entre os dois marcadores em função do tempo



Fonte: Análise própria no Matlab

O algoritmo foi implementado como uma classe, chamada *ProSupADM*, para facilitar sua utilização por outros softwares. Essa classe possui um método chamado *analyse* que realiza todo o algoritmo, desde processar o arquivo, calcular a ADM e mostrar o gráfico e os resultados. O único parâmetro que esse método recebe é o caminho do arquivo TSV a ser utilizado.

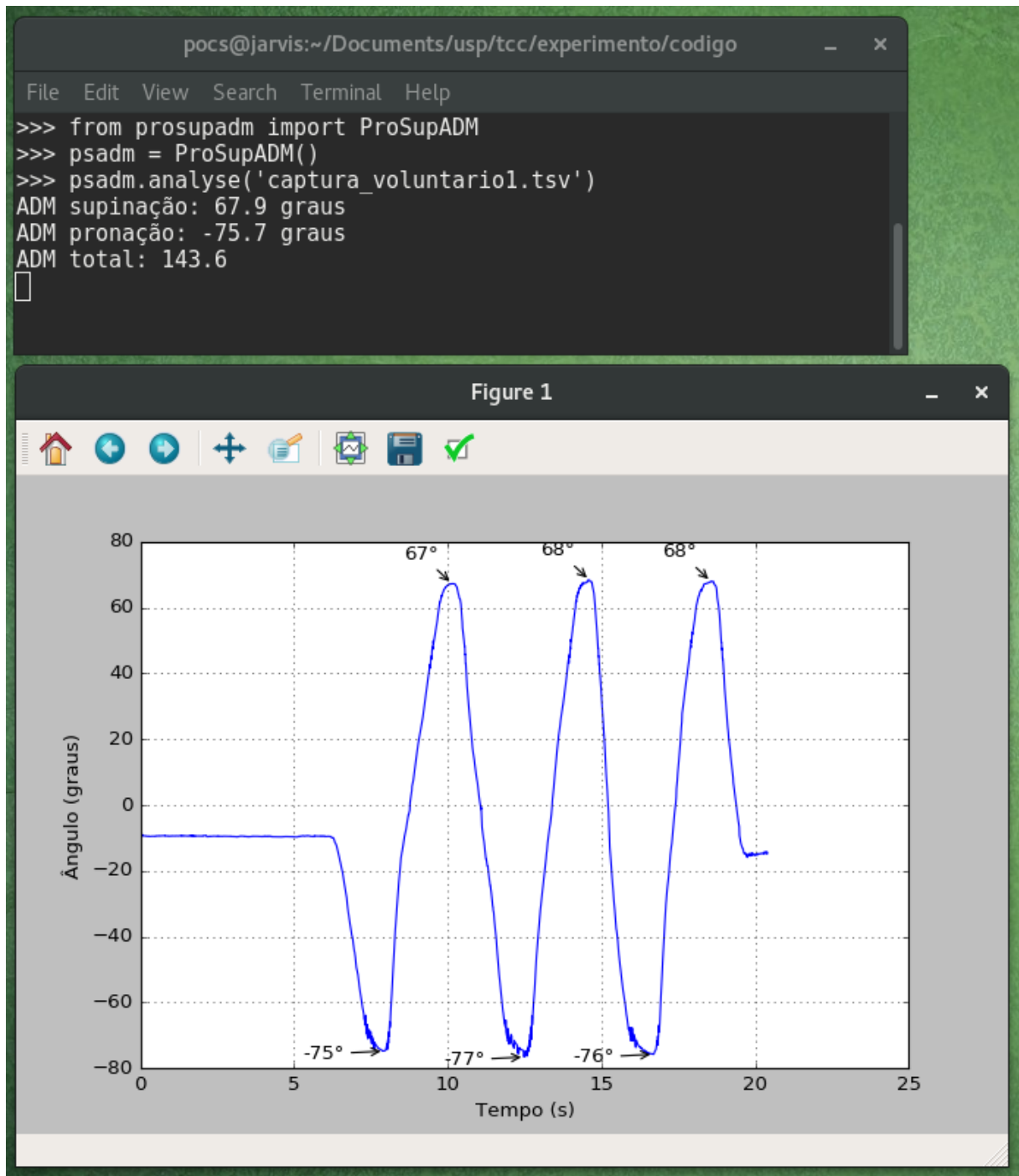
A [Figura 20](#) mostra um exemplo de utilização da classe pelo terminal interativo do Python. Esse terminal do Python pode ser acessado digitando *python3* na linha de comando do Windows (*cmd.exe*) e no Linux e Max OS no terminal padrão do sistema operacional.

O resultado da execução dos três comandos demonstrados na [Figura 20](#) é a exibição do gráfico e da impressão no terminal dos valores da ADM de pronação, supinação e total, como pode ser observado. Caso hajam erros, esses serão impressos na tela para que o usuário possa corrigi-los e tentar novamente.

Podem ser realizadas várias análises em sequência chamando o método *analyse* diversas vezes com arquivos diferentes como parâmetro. Porém, para que seja possível retornar à linha de comando, é necessário fechar a janela do gráfico gerado.

A janela do gráfico possui alguns recursos que podem ser utilizados, como: zoom, mover posição e salvar gráfico. É possível salvar o mesmo gráfico com diferentes níveis de zoom, por exemplo. Essa é uma boa maneira de analisar o resultado e gravá-lo no disco para referência futura.

Figura 20 – Imagem demonstrando a utilização da classe ProSupADM



Fonte: *Printscreen* de autoria própria

Além do método *analyse*, a classe *ProSupADM* possui outros três métodos de destaque: *processFile*, *calculateAngles* e *showPlot*. Esses métodos são utilizados pelo método principal, *analyse*, que fornece uma forma simples de executá-los todos em sequência. Porém, para usuários mais avançados, esses três métodos fornecem uma facilidade para realizar personalizações e extensões do algoritmo (como implementar um filtro passa-baixa

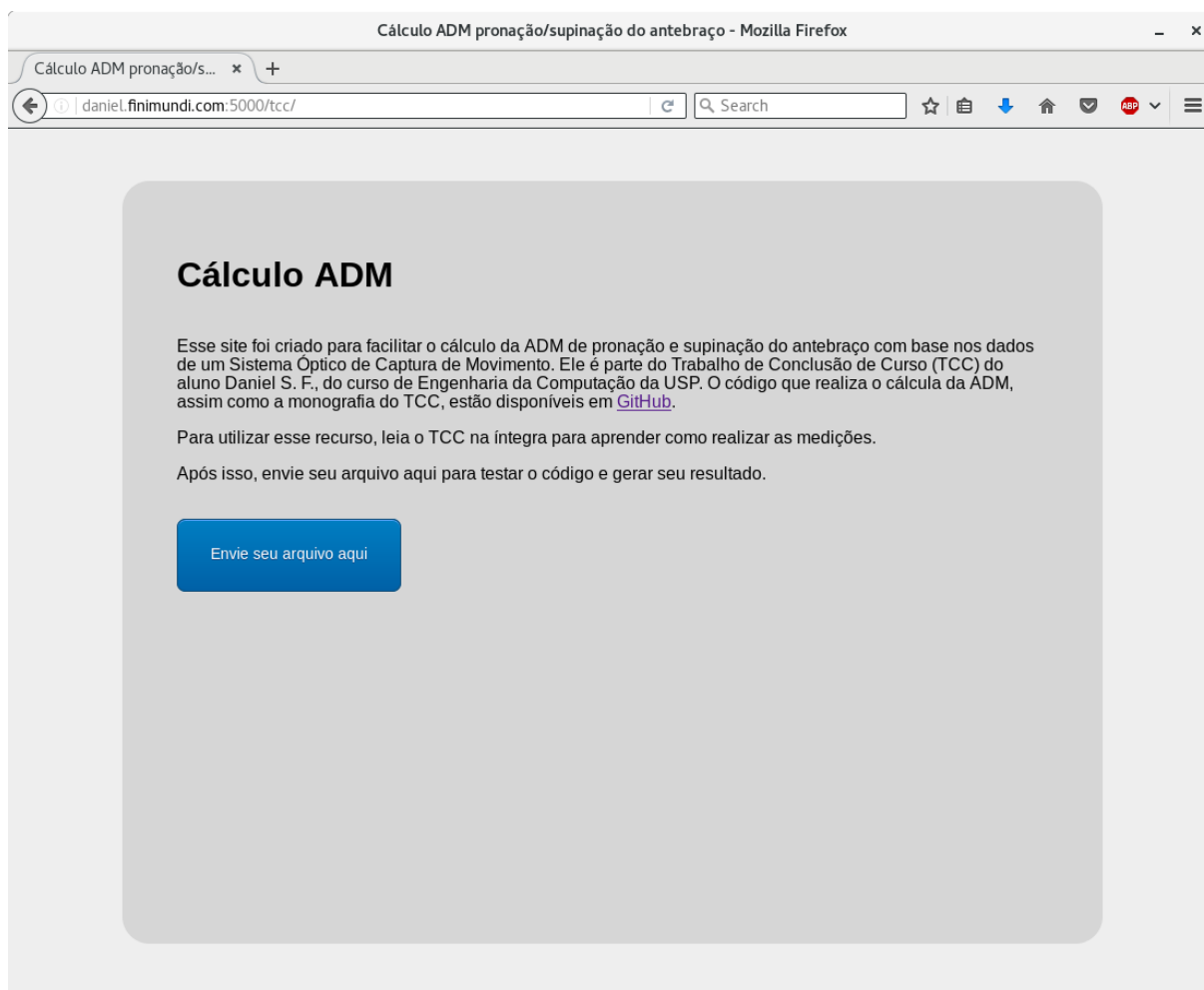
antes ou após o cálculo dos ângulos, ou mudar a forma como o gráfico é apresentado).

Nessa classe também existem atributos que facilitam a interação com usuários que desejam ir mais a fundo. Existe o atributo com a lista dos pontos 3D dos marcadores: *points*; a lista com os ângulos de rotação axial calculados: *angles*; e o resultados dos cálculos da ADM nos atributos *adm_pronation*, *adm_supination* e *adm_total*.

Utilizar o software pela linha de comando é muito versátil, porém requer certo conhecimento de programação para ser uma ferramenta útil. Como profissionais da saúde não costumam ter esses conhecimentos aprofundados em software, foi desenvolvido um site público para facilitar o acesso e utilização desse algoritmo.

Esse site está disponível temporariamente em <http://daniel.finimundi.com:5000/tcc/>. Seu código também é aberto sob licença MIT e está disponível em [Finimundi \(2016\)](#) para que qualquer um possa copiá-lo e hospedá-lo em um servidor de sua escolha.

Figura 21 – Tela inicial do site para utilização do algoritmo desenvolvido

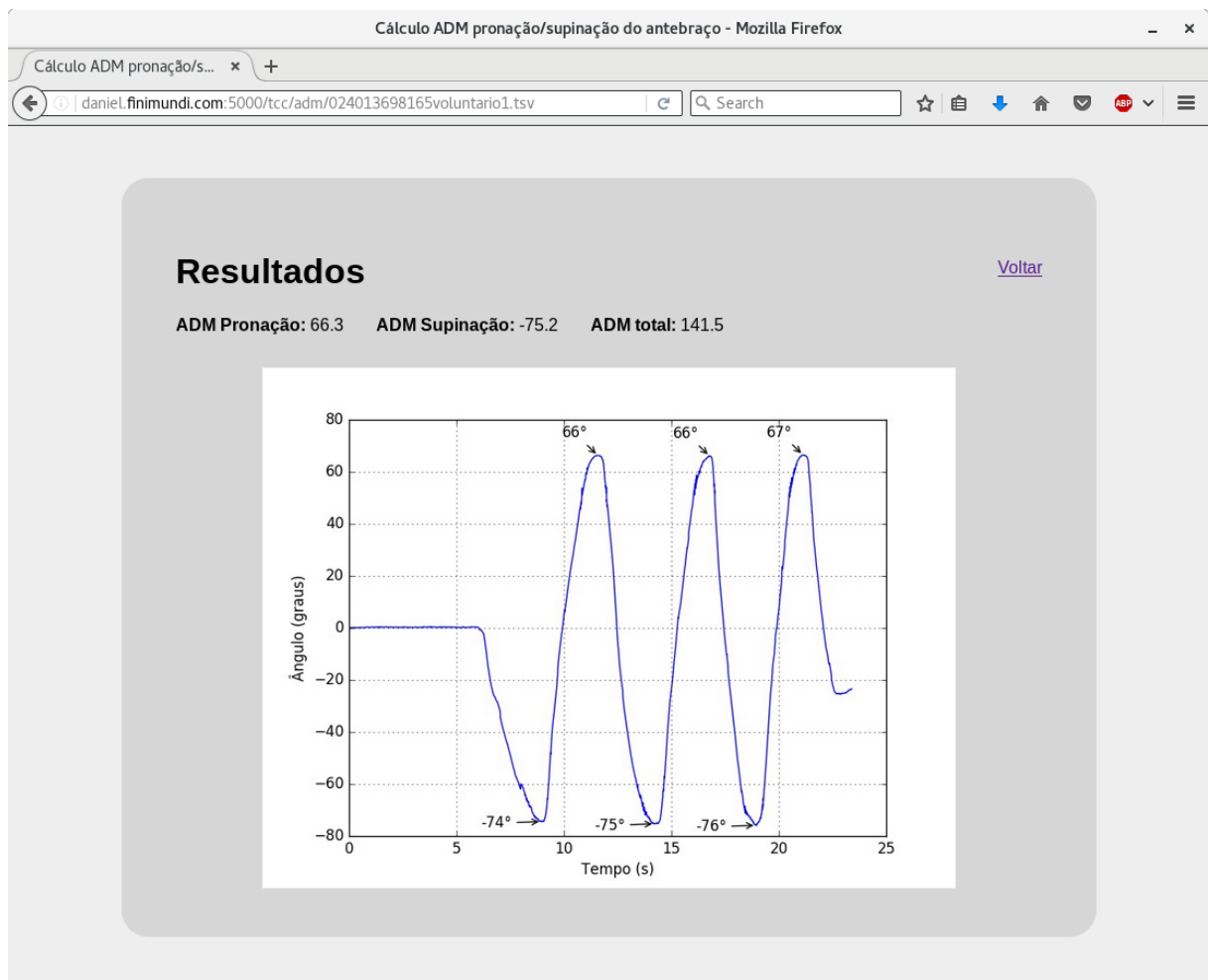


Fonte: *Printscreen* de autoria própria

O site também foi desenvolvido em Python, com o framework *Flask*. Sua utilização

é simples: na página inicial, [Figura 21](#), existe um botão para enviar o arquivo TSV. Enviando esse arquivo o site irá utilizar a classe *ProSupADM* para realizar os cálculos e irá carregar outra página com os resultados (gráfico e ADMs), [Figura 22](#). Eventuais erros serão mostrados com mensagens escritas em vermelho na tela. Para enviar outro arquivo, basta clicar no link *Voltar*, posicionado na extremidade superior direita do quadro cinza. Para salvar a imagem no computador, basta clicar com o botão direito do mouse sobre o gráfico e selecionar "*Salvar Imagem Como...*".

Figura 22 – Tela de resultados do site



Fonte: *Printscreen* de autoria própria

Eventuais problemas encontrados no software web podem ser reportados em [Finimundi \(2016\)](#) na aba *Issues*. Problemas reportados dessa forma serão endereçados no próprio sistema do GitHub e o código será atualizado para refletir a correção.

Assim, o algoritmo apresentado aqui no [Apêndice A](#) é somente uma versão inicial funcional e versões atualizadas serão disponibilizadas no site do GitHub, com a devida referência à esse trabalho original.

4.3 Experimento com voluntário

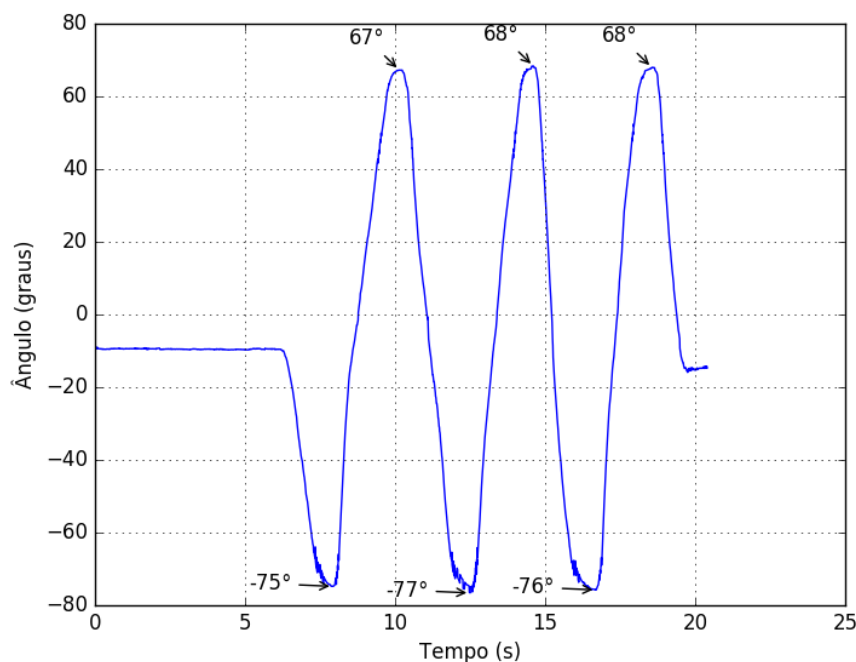
No experimento com o voluntário, os marcadores foram posicionados nos pontos anatômicos por uma fisioterapeuta, o voluntário foi colocado sentado na posição inicial e três capturas foram realizadas de acordo com a [seção 3.2](#), até a exportação dos dados para o arquivo TSV.

Os arquivos exportados das coletas foram utilizados no software da [seção 3.3](#) e [4.2](#). Os gráficos com a variação do ângulo de rotação axial (pronação positiva e supinação negativa) por tempo de cada uma das três coletas podem ser vistos nas Figuras [23](#), [24](#) e [25](#).

Para cada um dos gráficos, os três pontos de pico tanto de pronação, como de supinação, tiveram valores próximos, com diferenças de no máximo 2°.

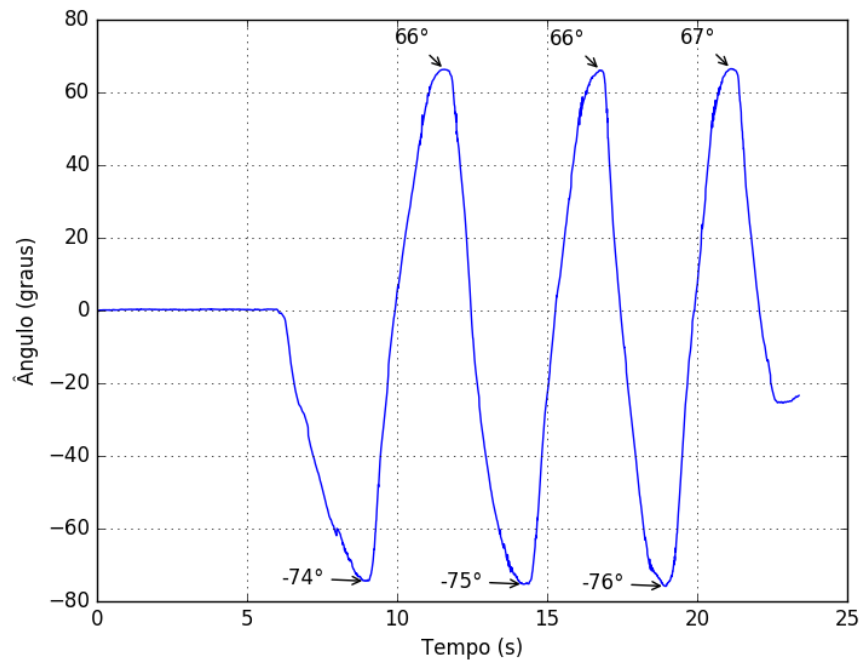
As informações das imagens estão resumidas na [Tabela 1](#). Para obter o valor de ADM, realizamos uma média com os três valores de pico para pronação e depois para supinação. Podemos observar que os valores resultantes de cada coleta foram consistentes, com uma variação máxima de 2°.

Figura 23 – Gráfico da primeira captura do ângulo de rotação axial do antebraço



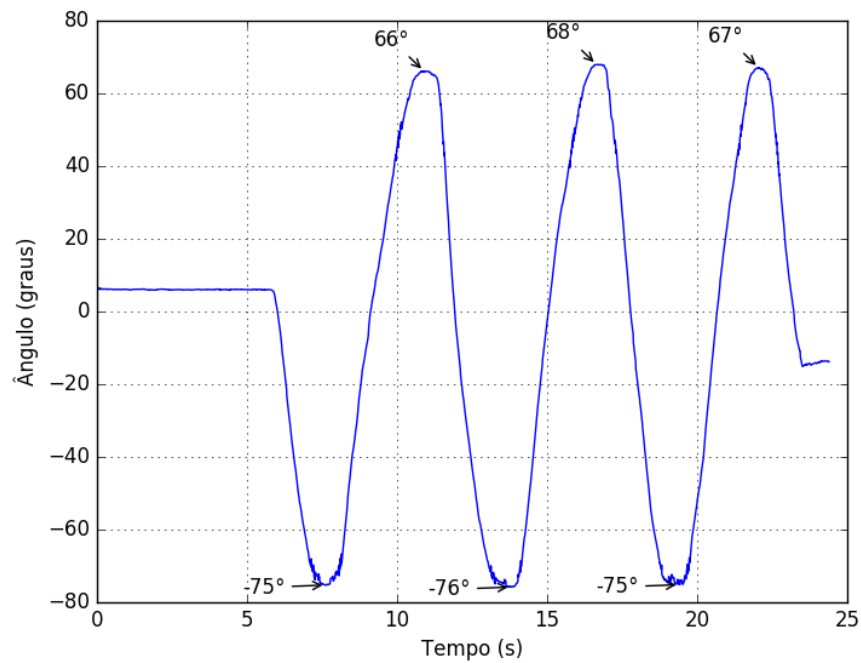
Fonte: Análise própria gerada pelo algoritmo da [seção 3.3](#)

Figura 24 – Gráfico da segunda captura do ângulo de rotação axial do antebraço



Fonte: Análise própria gerada pelo algoritmo da [seção 3.3](#)

Figura 25 – Gráfico da terceira captura do ângulo de rotação axial do antebraço



Fonte: Análise própria gerada pelo algoritmo da [seção 3.3](#)

Tabela 1 – Resultado da ADM para cada coleta e média geral

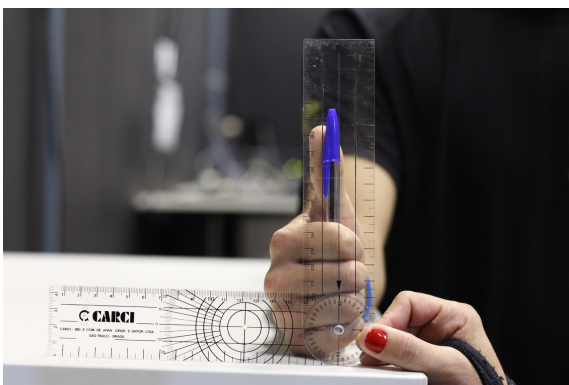
Coleta	Pronação (°)	Supinação (°)	Amplitude total (°)
Primeira	68	76	144
Segunda	66	75	141
Terceira	67	75	142
Média	67	75	142

Fonte: Produzido pelo autor.

Nota: A última linha, *Média*, representa a média dos valores das três coletas realizadas (sem precisão decimal.)

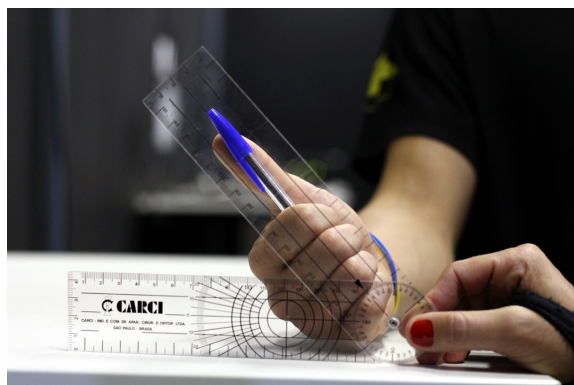
Após as três coletas, a fisioterapeuta também realizou a medida de pronação e supinação com um goniômetro universal, imagens mostrando a medida sendo realizada podem ser vistas nas Figuras 26 e 27. De acordo com essa medida, foi observado o ângulo de 70° para ambos os movimentos. Esse é um valor próximo do encontrado com o sistema de câmeras, porém, como o voluntário tem posições diferentes em cada medição, esse pode ser um fator de erro. Apesar disso, seriam necessários mais testes para identificar qual o motivo da discordância.

Figura 26 – Medição com o goniômetro universal. Posição neutra de rotação.



Fonte: Imagem de autoria própria

Figura 27 – Medição com o goniômetro universal. Movimento de supinação.



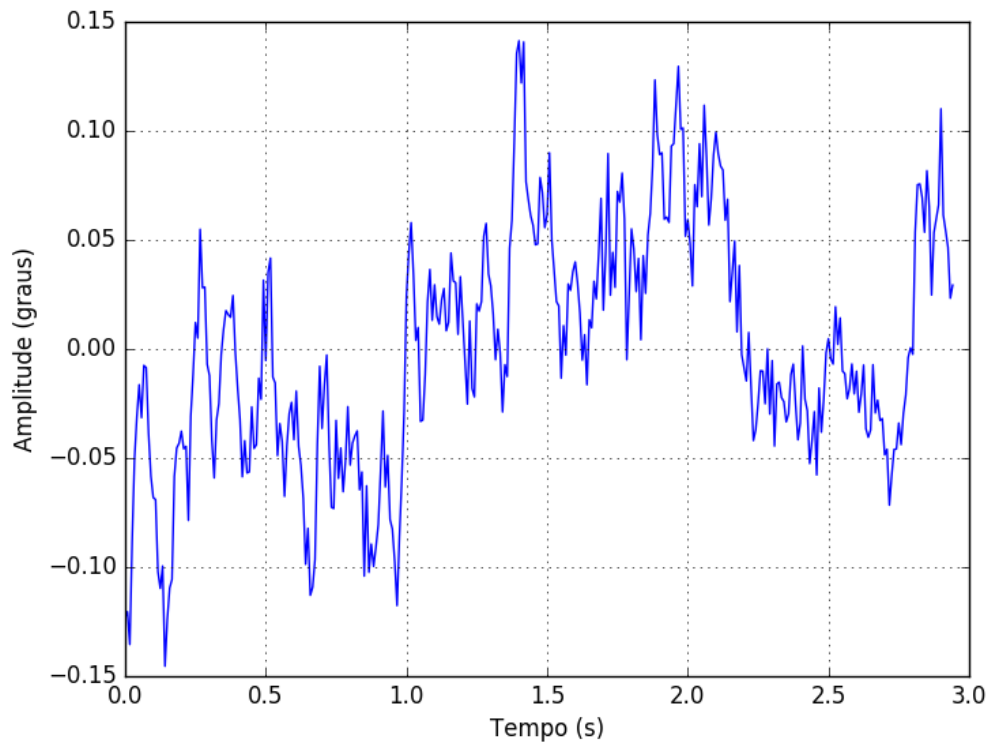
Fonte: Imagem de autoria própria

4.4 Análise de ruído

Outra análise realizada nas coletas foi a de ruído. O ruído nos marcadores individuais foi calculado no início desse capítulo, porém os cálculos realizados podem propagá-lo e amplificá-lo. Portanto devemos analisá-lo no resultado final do ângulo. A Figura 28 mostra o ângulo rotacional nos 3 primeiros segundos de uma coleta, onde o antebraço está parado em posição neutra. Podemos ver que existe uma instabilidade do sinal, porém a variação é

de menos de 0.15° .

Figura 28 – Gráfico do ângulo de rotação axial por tempo nos 3 segundos iniciais da coleta, momento em que o antebraço está parado em posição neutra.



Fonte: Análise própria em Python

Para encontrar a frequência predominante do ruído, foi realizada uma análise espectral utilizando o método de Welch. Essa função foi implementada no pacote *scipy.signal* no Python. Realizando essa análise no sinal do ângulo de rotação axial, obtemos o gráfico da Figura 29. É possível ver que a maior densidade de potência está em frequências muito próximas de zero, devido à baixa velocidade na realização do movimento. Porém, existe um pequeno pico da densidade de potência perto de 8 Hz, devido ao ruído.

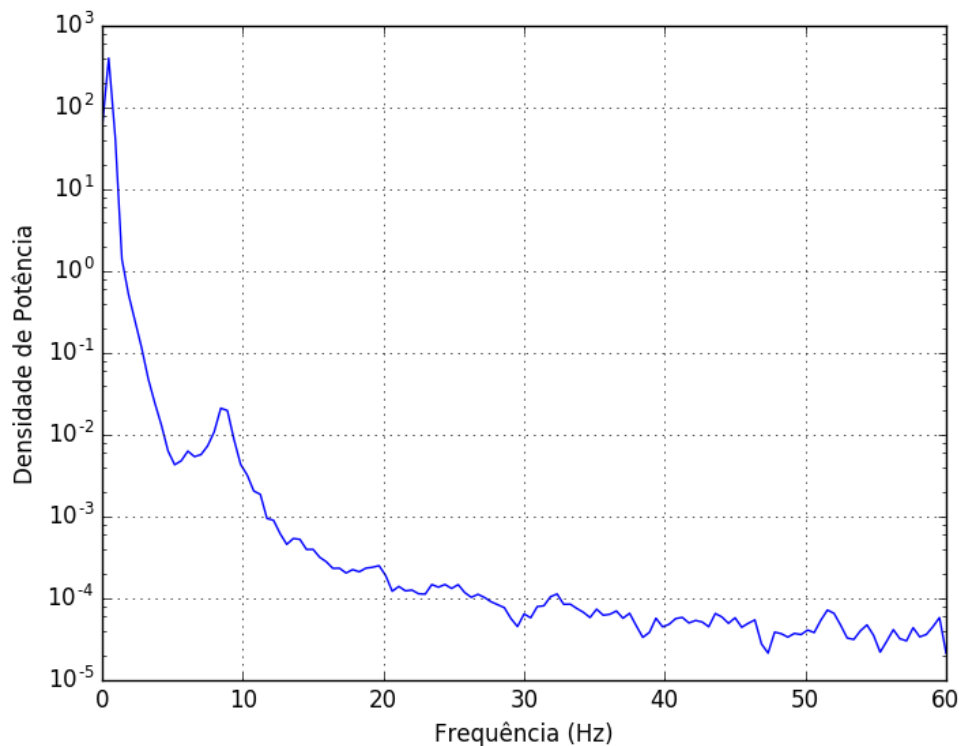
4.5 Discussão

Os gráficos obtidos nos resultados do trabalho foram bastante significativos: tanto os dados dentro de uma mesma coleta foram consistentes, como os resultados entre as três coletas também, conforme podemos ver pela Tabela 1.

Uma observação interessante é que podemos ver pelos gráficos que o voluntário iniciou cada coleta com uma posição neutra diferente. Um possível motivo dessa variação é a falta de um apoio para o cotovelo, fazendo com que o voluntário assuma posições iniciais diferentes e possivelmente interferindo nos resultados. Com isso, verificamos a importância

do posicionamento dos marcadores RS e US: se eles não estiverem exatamente alinhados, os ângulos de pronação e supinação serão alterados dependendo da posição inicial, apesar da amplitude total do movimento se manter a mesma.

Figura 29 – Gráfico da densidade de potência por frequência.



Fonte: Análise própria em Python

Essa pode ser uma razão pela qual os valores encontrados pelo algoritmo com o sistema de captura de movimento diferiu do valor medido pelo goniômetro universal. Um pequeno deslocamento entre os marcadores do punho (RS e US) pode ter alterado o *offset* da posição neutra.

Assim, considerando essa possibilidade, se houvesse um deslocamento da posição neutra (*offset*) de 4° em direção a pronação, os valores médios encontrados iriam de 67° de pronação e 75° de supinação para 71° em ambos, concordando com a medida de 70° encontrada com o goniômetro.

Apesar disso, não podemos garantir que esse problema realmente ocorreu, já que a literatura indica que normalmente o ângulo de supinação, em geral, é maior do que o de pronação: 85° de pronação e 90° de supinação de acordo com [Kapandji \(1987\)](#); e 70° de pronação e 85° de supinação de acordo com [Eaton \(2016\)](#). Assim, o algoritmo resultou em ângulos menores, porém aproximadamente correspondentes a esses valores da literatura.

O algoritmo em Python teve um tempo de execução entre 1 e 1.2 segundos, para um arquivo com 2449 a 3000 quadros (21 a 25 segundos de captura). Porém o algoritmo foi desenvolvido para facilidade de entendimento e facilidade de adaptação, como adicionar filtros ou realizar outras análises. Portanto, ainda seria possível otimizar o código para se obter uma performance melhor.

4.5.1 Ruído

Em alguns sistemas, o ruído pode degenerar o sinal a ponto dele não ser mais utilizável. Porém, pela [Figura 28](#), vemos que a interferência desse ruído é muito baixa em relação ao ângulo sendo medido, podendo ser desconsiderada nesse caso.

Apesar disso, análise de ruído da [Figura 29](#), mostra que a interferência ocorre aproximadamente de 6-8 Hz. Assim, se quiséssemos melhorar o sinal, poderia ser implementado um filtro digital passa-baixa com frequência de corte nessa faixa de 6 Hz a 8 Hz.

Na literatura ([GORDON; ROBERTSON, 2004](#)) ([FLEISIG; KWON, 2016](#)), o mais comum é ser implementado um filtro *Butterworth* passa-baixa de quarta ordem, com frequências de corte de 6 Hz a 12 Hz. Isso mostra que os resultados da análise do ruído foram condizentes.

Também é possível ver que ocorreu uma vibração involuntária no punho durante os movimentos de supinação, com amplitude bastante superior ao ruído. Essa vibração foi observada também durante a coleta e, de acordo com depoimento do voluntário, ocorreu devido à tensão dele próprio para realizar o movimento lentamente.

5 CONCLUSÃO

O objetivo desse trabalho era realizar uma análise da viabilidade de se desenvolver um software *open source* para medir a amplitude de movimento de pronação e supinação do antebraço com o auxílio de um sistema de captura de movimento.

De acordo com os resultados obtidos no [Capítulo 4](#), podemos concluir que os cálculos se mostram consistentes e promissores, apesar de ainda serem necessários mais experimentos para validar esse sistema.

Também chegamos a conclusão de que o posicionamento dos marcadores do punho é de extrema importância para o cálculo correto dos ângulos de pronação e supinação, apesar de não interferir com o valor total da amplitude de movimento. Seria interessante realizar testes retirando e colocando o marcador novamente para verificar a repetibilidade desse procedimento.

A princípio, o algoritmo se mostrou uma alternativa viável em relação a medição pelo goniômetro universal, com a possibilidade de se obter resultados mais precisos.

A implementação do algoritmo em Python se mostrou eficaz na facilidade de codificação, velocidade de processamento e versatilidade. E por ser uma linguagem de código aberto e poder ser utilizada sem custo, o código ficará disponível online, no site do GitHub, [Finimundi \(2016\)](#), sob licença MIT, que garante que qualquer um pode utilizar e alterar o código, desde que provida a referência ao trabalho original.

A interface web desenvolvida é de fácil utilização e possibilita que pessoas fora da área de tecnologia tenham mais facilidade em tirar proveito desse sistema.

Para os trabalhos futuros, é necessário realizar mais experimentos para validar o algoritmo. Esses experimentos devem ser realizados com diversos voluntários e alguns operadores diferentes para garantir a repetibilidade do cálculo. Além disso, também implementar a análise para o cotovelo esquerdo.

REFERÊNCIAS

- ABOUT Python. Python Software Foundation, 2016. Disponível em: <<https://www.python.org/about/>>. Acesso em: 24 out. 2016.
- ARMSTRONG, A. D. et al. Reliability of range-of-motion measurement in the elbow and forearm. **Journal of Shoulder and Elbow Surgery**, v. 7, n. 6, p. 573 – 580, 1998. ISSN 1058-2746. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1058274698900039>>.
- BARROS, R. et al. Desenvolvimento e avaliação de um sistema para análise cinemática tridimensional de movimentos humanos. **Revista Brasileira de Engenharia Biomédica**, v. 15, n. n. 1-2, p. p. 79–86, jan/ago 1999.
- C-MOTION. **Visual3D**. C-Motion, 1997. Disponível em: <<http://www2.c-motion.com/products/visual3d>>. Acesso em: 18 out. 2016.
- COLE, G. K. et al. Application of the joint coordinate system to three-dimensional joint attitude and movement representation: A standardization proposal. **Journal of Biomechanical Engineering**, v. 115, p. 344–349, 1993.
- EATON, C. **Normal Range of Motion Reference Values**. E-Hand.com The Electronic Textbook of Hand Surgery, 2016. Disponível em: <<http://www.eatonhand.com/nor/nor002.htm>>. Acesso em: 22 nov. 2016.
- FINIMUNDI, D. S. **Algoritmo para calculo da ADM de pronação e supinação do antebraço**. GitHub, 2016. Disponível em: <<https://github.com/danielmundi/PronoSupinoADM>>. Acesso em: 28 out. 2016.
- FLEISIG, G.; KWON, Y.-H. **The Biomechanics of Batting, Swinging, and Hitting**. [S.l.]: Routledge, 2016.
- GOOGLE Scholar. Google, 2016. Disponível em: <<https://scholar.google.com.br/scholar?q=isb+recommendations+joint+coordinate+systems>>. Acesso em: 22 out. 2016.
- GORDON, D.; ROBERTSON, E. **Research Methods in Biomechanics**. ilustrada. [S.l.]: Human Kinetics, 2004.
- GROOD, E. S.; SUNTAY, W. J. A joint coordinate system for the clinical description of three-dimensional motions: application to the knee. **Journal of biomechanical engineering**, American Society of Mechanical Engineers, v. 105, n. 2, p. 136–144, 1983.
- KAPANDJI, I. A. **Physiology of the Joints: Upper limb**. 6. ed. [S.l.]: Churchill Livingstone, 1987. v. 1.
- KWON, Y.-H. **Computation of the orientation angles**. Kwon3D, 1998. Disponível em: <<http://www.kwon3d.com/theory/euler/orient.html>>. Acesso em: 18 out. 2016.
- MARQUES, A. **Manual de Goniometria**. Editora Manole, 2003. ISBN 9788520440810. Disponível em: <<https://books.google.com.br/books?id=hV0iCgAAQBAJ>>. Acesso em: 02 out. 2016.

- NEGRI, L. H. **PeakUtils**. [s.n.], 2016. Versão 1.0.3. Disponível em: <<https://bitbucket.org/lucashnegri/peakutils>>. Acesso em: 26 out. 2016.
- NIGG, B. M.; HERZOG, W. **Biomechanics of the musculo-skeletal system**. [S.l.]: John Wiley & Sons, 2007.
- NUMPY. NumPy Developers, 2016. Disponível em: <<http://www.numpy.org/>>. Acesso em: 24 out. 2016.
- QUALISYS AB. Qualisys, 1989. Disponível em: <<http://www.qualisys.com/>>. Acesso em: 16 out. 2016.
- QUALISYS AB. **QTM - Qualisys Track Manager**. Packhusgatan 6, 411 13 Gothenburg, Sweden, 2013.
- VAROTO, R. **Desenvolvimento e avaliação de um protótipo de sistema híbrido para membro superior de tetraplégicos**. 2010. Tese (Doutorado) — Universidade de São Paulo, Escola de Engenharia de São Carlos, São Carlos, 2010.
- VELDPAUS, F.; WOLTRING, H.; DORTMANS, L. A least-squares algorithm for the equiform transformation from spatial marker co-ordinates. **Journal of Biomechanics**, v. 21, n. 1, p. 45 – 54, 1988. ISSN 0021-9290. Disponível em: <<http://www.sciencedirect.com/science/article/pii/002192908890190X>>.
- VENTURI, J. J. **Álgebra vetorial e geometria analítica**. 10. ed. [S.l.]: Livrarias Curitiba, 1949.
- WU, G. et al. Isb recommendation on definitions of joint coordinate systems of various joints for the reporting of human joint motion—part ii: shoulder, elbow, wrist and hand. **Journal of biomechanics**, Elsevier, v. 38, n. 5, p. 981–992, 2005.

Apêndices

APÊNDICE A – IMPLEMENTAÇÃO DO ALGORITMO EM PYTHON

Código da implementação do algoritmo para cálculo de ADM como a classe *ProSupADM* em Python.

```

1 import csv
2 import os.path
3 import math
4 import numpy as np
5 from numpy.linalg import norm
6 import matplotlib.pyplot as plt
7 import peakutils
8
9 class ProSupADM():
10     ''' Calculate forearm's axial rotation from optic
11     motion capture system's data '''
12
13     markers_order = ["AC", "EL", "EM", "RS", "US"]
14
15     def __init__(self, path=None):
16         self.path = path
17         self.initialize()
18
19     def analyse(self, path=None, imagePath=None):
20         if self.angles is not None:
21             self.initialize(path)
22
23         ok = self.processFile(path)
24         if not ok:
25             return False
26
27         ok = self.calculateAngles()
28         if not ok:
29             return False
30
31         ok = self.showPlot(imagePath)
32         if not ok:
33             return False
34
35     def processFile(self, path=None):

```

```

36     if not self.checkPath(path):
37         return False
38
39     with open(self.path, "r") as tsvfile:
40         tsvreader = csv.reader(tsvfile, delimiter="\t")
41
42     for line in tsvreader:
43         if line[0] == "NO_OF_FRAMES":
44             self.total_frames = int(line[1])
45             self.angles = np.empty(self.total_frames)
46         elif line[0] == "FREQUENCY":
47             self.frequency = int(line[1])
48         elif line[0] == "NO_OF_MARKERS":
49             if int(line[1]) != 5:
50                 print("Erro: Numero de marcadores " \
51                     "incorretos. 5 marcadores esperados, " \
52                     "{} encontrados no arquivo. Verifique " \
53                     "exportacao do software QIM.".format(
54                         int(line[1])))
55                 return False
56             elif line[0] == "MARKER_NAMES":
57                 if len(line) > 6:
58                     print("Erro: Cabecalho do arquivo TSV em " \
59                         "formato incorreto.")
60                     return False
61                 markers = line[1:]
62
63                 # Find the indices according to the expected
64                 # order. Considering the markers can be on
65                 # any order in the file
66                 indices = [markers.index(m) for m
67                           in self.markers_order]
68
69                 break
70
71     if len(markers) < 5:
72         print("Erro: Arquivo com menos marcadores do que " \
73             "o esperado.")
74         return False
75
76     for line in tsvreader:
77         if len(line) != len(markers)*3:
78             print("Erro: Incosistencia nos vetores de " \

```

```

78         "posicao dos marcadores.")
79         return False
80
81         line = list(map(float, line))
82         # Read the 3D position of each marker from
83         # the current line
84         AC, EL, EM, RS, US = (
85             np.asarray(line[3*indices[i]: 3*indices[i]+3])
86                         for i in range(5)
87         )
88
89         self.points.append((AC, EL, EM, RS, US))
90
91         return True
92
93     def calculateAngles(self):
94         if len(self.points) == 0:
95             print("Erro: Chame 'processFile()' antes de calcular " \
96                 "os angulos.")
97             return False
98         if self.frequency == 0:
99             self.frequency = 120
100             print("Atencao: Frequencia de captura nao definida no " \
101                 "cabecalho. Utilizando o padrao de 120 Hz.")
102
103         for i, (AC, EL, EM, RS, US) in enumerate(self.points):
104             epi_mean = np.asarray([(x+y)/2 for x,y in zip(EL, EM)])
105
106             # Defining forearm coordinate system
107             Of = US
108             Yf = self.defineVector(epi_medium, US)
109             Xf = self.crossNorm(self.defineVector(epi_mean, RS),
110                               self.defineVector(epi_mean, US))
111             Zf = self.crossNorm(Xf, Yf)
112
113             # Defining humerus coordinate system
114             Oh = AC
115             Yh = self.defineVector(AC, epi_mean)
116             Zh = self.crossNorm(Yf, Yh)
117             Xh = self.crossNorm(Yh, Zh)
118
119             fj = Xf

```

```

120         lj = Yf
121         tj = self.crossNorm(lj , fj)
122
123         e1 = Zh
124         e3 = Yf
125         e2 = self.crossNorm(e3 , e1)
126
127         if (np.dot(np.cross(e3 , e1) , tj) < 0 and
128             np.dot(np.cross(np.cross(e3 , e1) , e3) , fj)) > 0:
129             e2 = -e2
130
131         angle1 = math.copysign(
132             math.acos(np.dot(e2 , Xh)) ,
133             np.dot(e2 , Yh)
134         )
135         angle3 = math.copysign(
136             math.acos(np.dot(e2 , tj)) ,
137             np.dot(e2 , fj)
138         )
139
140         self.angles[i] = math.degrees(angle3)
141
142         self.angles = np.asarray(self.angles)
143         # Need to reverse angles to find supination peaks
144         angles_neg = -1*self.angles
145
146         self.x = np.arange(0 , len(self.angles)/120 , 1/120)
147         self.x = self.x[:len(self.angles)]
148
149         # Find pronation peaks
150         self.id_prono_peaks = peakutils.indexes(self.angles ,
151             thres=0.2 , min_dist=240)
152         # Find supination peaks
153         self.id_supino_peaks = peakutils.indexes(angles_neg ,
154             thres=0.3 , min_dist=240)
155
156         # Calculate mean for the pronation peaks
157         self.adm_pronation = sum(self.angles[i] for i
158             in self.id_prono_peaks)/len(self.id_prono_peaks)
159         # Calculate mean for the supination peaks
160         self.adm_supination = sum(self.angles[i] for i
161             in self.id_supino_peaks)/len(self.id_supino_peaks)

```

```

162
163     self.adm_total = self.adm_pronation-self.adm_supination
164
165     print("ADM supinacao: {:.1f} graus".format(
166         self.adm_pronation))
167     print("ADM pronacao: {:.1f} graus".format(
168         self.adm_supination))
169     print("ADM total: {:.1f}".format(
170         self.adm_total))
171
172     return True
173
174 def showPlot(self, imagePath=None):
175     if len(self.angles) == 0 or len(self.angles) != len(self.x):
176         print("Erro: Execute as funcoes 'processFile' e ' \
177             'calculateAngles' antes de imprimir o grafico.")
178         return False
179
180     fig = plt.figure()
181     sp = fig.add_subplot(111)
182
183     sp.plot(self.x, self.angles)
184
185     for peak in self.id_prono_peaks:
186         sp.annotate("{:.0f}".format(self.angles[peak]),
187             xy=(peak/120, self.angles[peak]), xycoords='data',
188             xytext=(-30, 15), textcoords='offset points',
189             arrowprops=dict(arrowstyle="->"))
190
191     for peak in self.id_supino_peaks:
192         sp.annotate("{:.0f}".format(self.angles[peak]),
193             xy=(peak/120, self.angles[peak]), xycoords='data',
194             xytext=(-50, -5), textcoords='offset points',
195             arrowprops=dict(arrowstyle="->"))
196
197     plt.xlabel("Tempo (s)")
198     plt.ylabel("Angulo (graus)")
199
200     sp.grid()
201
202     if imagePath is None:
203         plt.show()

```

```
204         plt.draw()
205     else:
206         plt.savefig(imagePath)
207
208     return True
209
210 def defineVector(self, end, begin):
211     return (end-begin)/norm(end-begin)
212
213 def crossNorm(self, vec1, vec2):
214     vecr = np.cross(vec1, vec2)
215     vecr = vecr/norm(vecr)
216     return vecr
217
218 def checkPath(self, path):
219     if path is None:
220         print("Erro: Nenhum arquivo selecionado")
221         return False
222     elif not os.path.exists(path):
223         print("Erro: Arquivo nao existente")
224         return False
225     elif path.rsplit('.', 1)[1] not in ['tsv', 'TSV']:
226         print("Erro: Extensao do arquivo deve ser '.tsv'")
227         return False
228     else:
229         self.path = path
230         return True
231
232 def initialize(self, path=None):
233     if path is not None:
234         self.path = path
235     self.total_frames = 0
236     self.frequency = 0
237     self.angles = None
238     self.x = None
239     self.id_prono_peaks = []
240     self.id_supino_peaks = []
241     self.points = []
242     self.adm_pronation = 0
243     self.adm_supination = 0
244     self.adm_total = 0
```