



UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Mecânica

ALYSSON FERNANDES MAZONI

**Utilização das ferramentas da Inteligência
Artificial em aplicações mecatrônicas – Estudo de
casos**

CAMPINAS

2021

ALYSSON FERNANDES MAZONI

Utilização das ferramentas da Inteligência Artificial em aplicações mecatrônicas – Estudo de casos

Tese apresentada à Faculdade de Engenharia Mecânica da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Doutor em Engenharia Mecânica, na Área de Mecatrônica.

Orientador: Prof. Dr. João Maurício Rosário

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA TESE DEFENDIDA PELO ALUNO ALYSSON FERNANDES MAZONI, E ORIENTADA PELO PROF. DR. JOÃO MAURÍCIO ROSÁRIO.

CAMPINAS

2021

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

M458u Mazoni, Alysson Fernandes, 1983-
Utilização das ferramentas da Inteligência Artificial em aplicações
mecatrônicas - estudo de casos / Alysson Fernandes Mazoni. – Campinas, SP :
[s.n.], 2021.

Orientador: João Maurício Rosário.
Tese (doutorado) – Universidade Estadual de Campinas, Faculdade de
Engenharia Mecânica.

1. Aprendizado de máquina. 2. Inteligência Artificial. 3. Redes Neurais
Convolucionais. 4. Identificação de sistemas. 5. Simulação computacional. I.
Rosário, João Maurício, 1959-. II. Universidade Estadual de Campinas.
Faculdade de Engenharia Mecânica. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Use of Artificial Intelligence in mechatronic applications - Case studies

Palavras-chave em inglês:

Machine learning

Artificial intelligence

Convolutional neural networks

Systems identification

Computational simulation

Área de concentração: Mecatrônica

Títuloção: Doutor em Engenharia Mecânica

Banca examinadora:

João Maurício Rosário [Orientador]

Ely Carneiro de Paiva

Francisco Carlos Parquet Bizarria

Oscar Fernando Avilés Sánchez

Daniel Augusto Pereira

Data de defesa: 15-03-2021

Programa de Pós-Graduação: Engenharia Mecânica

Identificação e informações acadêmicas do(a) aluno(a)

- ORCID do autor: <https://orcid.org/0000-0001-5265-689>

- Currículo Lattes do autor: <http://lattes.cnpq.br/0347768173274366>

**UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA**

TESE DE DOUTORADO ACADÊMICO

Utilização das ferramentas da Inteligência Artificial em aplicações mecatrônicas – Estudo de casos

Autor: Alysson Fernandes Mazoni

Orientador: Prof. Dr. João Maurício Rosário

A Banca Examinadora composta pelos membros abaixo aprovou esta Tese:

Prof. Dr. João Maurício Rosário

Universidade Estadual de Campinas - FEM/UNICAMP

Prof. Dr. Ely Carneiro de Paiva

Universidade Estadual de Campinas - FEM/UNICAMP

Prof. Dr. Francisco Carlos Parquet Bizarria

Universidade de Taubaté

Prof. Dr. Oscar Fernando Avilés Sánchez

Universidad Militar Nueva Granada

Prof. Dr. Daniel Augusto Pereira

Universidade Federal de Lavras

A Ata de Defesa com as respectivas assinaturas dos membros encontra-se no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 15 de Março de 2021.

AGRADECIMENTOS

Este trabalho só foi possível graças à contribuição de pessoas que trabalham com pesquisa, tecnologia e ciência no Brasil. Cito os mais os mais próximos: Carolina Zabini, Daniela Virginia Vaz, Valéria Andrade Pinto, Carolina Marques Andrade, Renan Fernandes Kozan, Carlos Caetano de Almeida, Lucas Goulart Vazquez, Pedro Henrique Correa Kim, e o orientador deste trabalho, João Maurício Rosário.

Agradecemos à FAPESP pelo suporte financeiro ao projeto 2017/10956-5 que fez parte desta tese.

Agradecemos à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), código de financiamento 001, pelo suporte aos projetos desenvolvidos no Programa de Pós Graduação em Ciências da Reabilitação na UFMG que fizeram parte deste trabalho.

RESUMO

A modelagem de sistemas dinâmicos como auxiliar na fusão de sensores é ferramenta corrente da engenharia mecatrônica. Essa abordagem se apoia no conhecimento físico dos problemas e na capacidade de escrever relações quantitativas para eles. Por outro lado, abordagens baseadas em inteligência artificial apelam para a construção de aproximações opacas de sistemas dinâmicos que contornam dificuldades de simulação de problemas com representação vaga ou matematicamente intratável. Este trabalho propõe aproveitar as vantagens dessas duas visões de forma coerente em alguns problemas de aplicação de diversas áreas. A combinação entre modelagem e inteligência artificial é usada para: 1 - rastrear movimento humano a partir de sensores inerciais; 2 - detectar falhas de desgaste em painéis fotovoltaicos usando imagens obtidas por câmeras de drones; 3 - auxiliar na detecção de características tridimensionais de imagens em um microscópio. Os resultados apontam para uma futura organização de como utilizar essas técnicas em conjunto.

Palavras-chaves: Aprendizado de Máquina, Inteligência Artificial, Redes Neurais Convolucionais, Modelagem de sistemas, Identificação de sistemas, Sistemas dinâmicos.

ABSTRACT

System modeling as auxiliary in sensor data fusion is a frequent tool in mechatronics. This approach relies on the physical knowledge we have of problems and skill to write down mathematical relations for them. On the other side, approaches based in artificial intelligence resort to opaque approximations of dynamical systems that avoid simulation and mathematical representation difficulties. This work proposes to seize on the advantages of these two thoughts in problems stemming from diverse areas. This combination of modelling and AI is used to: 1 - track human movement with inertial sensors; 2 - detect faults in photovoltaic panels using images taken with drones; 3 - help to measure morphometric properties in images taken with a microscope. The results indicate a future organization in the order and ways of using these techniques combined.

Keywords: Machine Learning, Artificial Intelligence, Convolutional Neural Networks, Systems modelling, Systems Identification, Dynamic Systems.

LISTA DE ILUSTRAÇÕES

Figura 2.1 – Sistema de aquisição de movimento. Fonte: (LEE <i>et al.</i> , 2018)	14
Figura 3.1 – Ambiente de desenvolvimento com suas abas de inspeção.	21
Figura 3.2 – Cenário com um plano e uma esfera rígida com massa. A localização de uma fonte de luz e de um ponto de vista (câmera) está visíveis, além do sistema de coordenadas associado.	22
Figura 3.3 – Painel de propriedades físicas.	23
Figura 5.1 – Avaliação de pirueta no balé por meio do celular. (ANDRADE <i>et al.</i> , 2020)	29
Figura 5.2 – Tarefa de passar de sentado para de pé, avaliada usando um celular. Fonte: (PINTO <i>et al.</i> , 2020)	30
Figura 5.3 – O aparelho de celular possui sensores de aceleração, campo magnético e velocidade angular em relação aos três eixos.	31
Figura 5.4 – A função das equações de modelo apresentadas é de calcula uma estimativa para os ângulos de rotação e duas estimativas independentes para a velocidade angular.	34
Figura 5.5 – O autoencoder tenta reproduzir a saída usando um intervalo de tempo de comprimento T dos dados obtidos pelo modelo. A estimativa da velocidade angular na saída é feita pela média das duas. $\Omega(t) = \Omega_1(t) + \Omega_2(t)$	34
Figura 5.6 – Interface de treinamento para o autoencoder e outras formas de redes neurais com Matlab.	35
Figura 5.7 – Gráficos comparativos para um teste de estimativa com ângulos e velocidades angulares.	36
Figura 5.8 – Falhas visíveis em painel fotovoltaico usando foto de espectro infravermelho.	37
Figura 5.9 – Funcionamento da detecção de falhas em uma situação prática.	39
Figura 5.10–Exemplo de imagem gerada pelo programa de simulação.	40
Figura 5.11–Esquema de funcionamento do microscópio tridimensional.	43
Figura 5.12–Modelo de rede mista para entrada de imagem e altura.	46
Figura 5.13–Modelo de rede para entrada de imagem.	46
Figura 5.14–Modelo de rede mista para entrada de imagem em escala em cinza e altura. .	47
Figura 5.15–Modelo de rede para entrada de imagem em escala em cinza.	47
Figura 5.16–Exemplos de fósseis usados no treinamento.	49

SUMÁRIO

1	Introdução	11
1.1	Motivação	11
1.2	Motivação a partir da literatura	12
1.3	Aplicações e estrutura do trabalho	13
2	Revisão bibliográfica	14
2.1	Aplicações mecatrônicas	14
2.1.1	Biomecânica	14
2.1.2	Simulação tridimensional	15
2.1.3	Análise e classificação de imagens por visão	16
2.2	Posicionamento dinâmico	16
2.3	Inteligência Artificial	16
2.4	Aprendizado de máquina (<i>Machine learning</i>)	17
2.5	Redes neurais	17
2.6	Aprendizado profundo (<i>Deep learning</i>)	17
2.7	Visão computacional	17
2.8	Linguagens e pacotes	18
3	Simulação de ambientes tridimensionais	19
3.1	Introdução	19
3.2	Motores gráficos e de física	20
3.3	Exemplo de simulação	21
3.3.1	Montagem de cenário	21
3.3.2	Definições físicas	22
3.4	Aplicação de modelos dinâmicos	22
3.4.1	Leitura de variáveis dinâmicas	24
3.5	Conclusões	25
4	Inteligência artificial aplicada a sistemas mecatrônicos	26
4.1	Redes neurais	26
4.1.1	Definições	26
4.1.2	Arquitetura e treinamento	27
4.2	Interpretações	28
5	Estudos de caso	29
5.1	EC 1 – Fusão de dados em sensores inerciais	29
5.1.1	Descrição do problema	29
5.1.2	Divisão de modelo	29
5.1.3	Treinamento e teste de rede	33
5.1.4	Resultados	35

5.2	EC2 – Falhas em painéis fotovoltaicos	36
5.2.1	Descrição do problema	36
5.2.2	Localização de falhas na imagem	37
5.2.3	Localização da imagem no sistema do drone	38
5.2.4	Produção de imagens em simulação	39
5.2.5	Ajuste por rede neural	41
5.2.6	Conclusões	42
5.3	EC 3 – Microscópio tridimensional	42
5.3.1	Descrição do problema	42
5.3.2	Reconstrução tridimensional	43
5.3.3	Erros de reconstrução	44
5.3.4	Ajuste por rede neural	45
5.3.5	Produção de imagens em simulação	48
5.3.6	Modelagem de Redes Neurais	48
5.3.7	Treinamento e teste de rede	48
5.3.8	Conclusões	50
6	Conclusões e Estudos Futuros	51
6.1	A combinação de modelos e redes neurais	51
6.2	CrITÉrios empÍricos de separação	51
6.3	Redes neurais como aproximadores	52
6.4	Trabalhos e aplicações possíveis	53
	Referências	55
	ANEXO A Código e documentação para dados de sensores inerciais	64
	ANEXO B Código e documentação para geração de imagens de microscópio	79
	ANEXO C Código e documentação para treinamento de rede para microscópio	82
	ANEXO D Código e documentação para geração de imagens de falhas em painéis	96
	ANEXO E Código e documentação para detecção de falhas em painéis	109
	ANEXO F Publicações relacionadas com esta tese	116

1 INTRODUÇÃO

1.1 Motivação

Em diversas áreas da engenharia encontramos a necessidade de obter medidas de variáveis dinâmicas que não estão facilmente acessíveis e para as quais não se pode recorrer a sensores diretamente. Não se pode obter a temperatura em todos os pontos relevantes em uma coluna de destilação. Não se sabe a deformação de um corpo sob tensão em um ponto qualquer de seu interior.

Além dessas situações, tem-se tornado mais comum o uso de valores de grandezas que não tem definição quantitativa clara em uma aplicação mas que possuem grande importância de uso prático. Por exemplo, medidas baseadas na teoria de informação como entropia multiescala, como abordada em (GAO *et al.*, 2015). Em um problema de detecção de falhas, a severidade de uma falha é uma grandeza que está relacionada a vários parâmetros físicos mas que não representa nenhum deles exatamente.

Para lidar com o problema de obter valores a partir da medida de outras grandezas relacionadas desenvolveu-se a teoria dos observadores (FRANKLIN *et al.*, 2014). Nesse contexto, o conhecimento das equações dinâmicas de um processo permite avaliar uma variável desejada a partir de outras que sejam medidas. As equações do modelo servem de direcionamento para o cálculo, (KLUEVER, 2014; GARRIDO, 2018).

Isso pode ser conseguido de uma forma causal, ou seja, usando medidas passadas e calculando instantaneamente os valores desejados. Também pode ser feito de forma não-causal, que significa usar as medidas anteriores e posteriores para estimar o valor da variável em um ponto. Essa abordagem só é possível nos casos que em que um trecho de medida está disponível para que se calculem as variáveis desejadas após a realização prática da medida. Esse formato exige explorar a teoria de otimização (YONG, 2018).

Para a questão de medidas de conhecimento físico desconhecido ou sem interpretação direta desenvolvem-se diversas áreas de conhecimento agrupadas sob o nome de *aprendizado de máquina*, especialmente no ramo da inteligência artificial (GOODFELLOW *et al.*, 2016). Para os problemas tratados neste trabalho, o subconjunto das técnicas desse ramo que se usam são as redes neurais.

Usando essas abordagens, características difusas medidas por sensores podem ser quantificadas, ou mesmo localização de elementos em imagens podem ser isolados ainda que não tenham uma definição geométrica que possa ser reduzida a filtros ou ferramentas do processamento de imagens, ver (SZELISKI, 2010).

1.2 Motivação a partir da literatura

Os observadores de estado possuem uma longa história dentro da teoria de sistemas lineares e quando combinados com a estatística normal produzem a solução do filtro de Kalman, de uso onipresente na eletrônica e nas telecomunicações, (GARRIDO, 2018). Quaisquer soluções que visem obter medidas de variáveis usando modelos físicos dependem do conhecimento da área estudada, (BESANCON, 2007; LUENBERGER, 1971). No caso das aplicações aqui usadas, foi necessária a cinemática tridimensional para obter medidas relevantes, (SICILIANO; KHATIB, 2007; CORKE, 2013; THRUN *et al.*, 2005).

As redes neurais representam a ferramenta mais usada do campo conhecido como inteligência artificial e certamente a mais apropriada para o uso com dados de sensores. Os livros introdutórios sobre o assunto apresentam geralmente uma origem relacionada aos estudos de modelagem do sistema nervoso humano. Inicialmente houve também um interesse do ponto de vista de ciência cognitiva. Esse último elemento tem se reanimado com obras que voltaram a uma explicação computacional da cognição com as aplicações práticas de redes neurais, (BLANCHARD *et al.*, 2019).

Em algumas aplicações, as redes neurais possuem poucas camadas, especialmente quando se trata da representação de dados de poucas dimensões. A literatura mais recente tem dado grande destaque às redes de grande número de camadas, especialmente nas aplicações que tem imagens como entradas, (ROSA, 2018; NIKLAUS *et al.*, 2017).

Na prática, as redes são mais eficientes com organizações de camadas diferentes conforme as aplicações. Essas diferentes organizações são adaptadas a famílias de problemas específicas como

- Segmentação – a propriedade de marcar sobre uma imagem de entrada os pontos que representam um objeto a ser detectado. Por exemplo, encontrar um cachorro em fotos, encontrar um rosto humano, entre outros, (SULTANA *et al.*, 2020).
- Classificação – determinar a presença de algum tipo de objeto sobre uma imagem, (WANG *et al.*, 2016). Neste caso trata-se de apenas dizer se um tipo de objeto ou situação está presente na imagem. Por exemplo, detectar se há pessoas de pé, ou sentadas.
- Linguagem natural – determinação de ideias particulares treinadas ou sentimentos a partir de textos que são cadastrados em um vocabulário de palavras, (NADKARNI *et al.*, 2011).
- Reconhecimento de fala – determinação de palavras pronunciadas a partir de sequências de pontos de arquivos sonoros, (GRAVES *et al.*, 2013).

1.3 Aplicações e estrutura do trabalho

Neste trabalho tratam-se de três aplicações como exemplos do uso de redes neurais. São usadas em colaboração com modelos matemáticos para obtenção de medidas relevantes para um medidas que não são diretamente acessíveis pelos sensores disponíveis.

O capítulo 2 relaciona os principais conceitos usados no trabalho com a literatura que os sustenta.

O capítulo 3 instrui sobre o uso de imagens artificiais que podem ser usadas em redes neurais para o treinamento. Ou seja, produção de imagens para treinamento em computador. Conforme publicações recentes tem apresentado, é possível usar imagens sintetizadas por sistemas de simulação devido à capacidade das redes de generalizar comportamento a partir de modelos mais simples.

No capítulo 4 são citadas as principais ideias da área de redes neurais no sentido de apresentação das escolhas feitas ao utilizar os pacotes de software. Além disso, coletam-se as conclusões recentes da pesquisa na área que podem auxiliar este trabalho.

O capítulo 5 apresenta os três estudos de caso do trabalho. Estão divididos em seções.

A seção 5.1 apresenta o estudo de caso de aplicação em biomecânica. Ali, usam-se leituras de variáveis cinemáticas através de sensores inerciais. Para isso, é usada uma rede neural do tipo autoencoder em colaboração com um modelo do comportamento dos sensores.

A seção 5.2 apresenta o problema de detecção de falhas em painéis fotovoltaicos de forma independente do processamento de imagens. A solução proposta usa imagens de drones que são obtidas com conhecimento de posicionamento da câmera. As imagens e os painéis fotovoltaicos são identificados e mapeados por uma rede neural treinada com imagens artificiais.

A seção 5.3 apresenta o uso de redes neurais para melhorar estimativas de medidas tridimensionais em um microscópio que funciona com variação da distância focal. Evitando técnicas de processamento de imagem, uma rede neural pode estimar a distância focal e a inclinação de imagens obtidas.

As conclusões e ideias gerais de aplicação são sintetizadas no capítulo 6 e seguem os anexos com os códigos para as aplicações e as publicações relacionadas.

2 REVISÃO BIBLIOGRÁFICA

2.1 Aplicações mecatrônicas

2.1.1 Biomecânica

A medida de movimentos em biomecânica é fundamental para avaliações experimentais com fins terapêuticos, de avaliação esportiva e também no estudo experimental de hipóteses e modelos. A principal ferramenta para esse tipo de problema é o uso de sistemas de aquisição de movimento, (KOLAHÍ *et al.*, 2007). Tais sistemas capturam as coordenadas tridimensionais de pontos em movimento. Os sistemas rastreiam a posição de pontos específicos na forma de elementos refletivos que são rastreados por sua distância até câmeras específicas por princípios de sinal infravermelho ou de luz visível. A determinação de posição consiste em determinar coordenadas que satisfaçam as distâncias medidas em diversos elementos sensores simultâneos, como câmeras de infravermelho ou de luz visível.

Esse tipo de equipamento é custoso e complexo e só pode ser usado em condições de laboratório dada sua distribuição especial e também a necessidade de instalação dos elementos refletivos em pontos do corpo para obter seu movimento. A instalação de tais dispositivos também apresenta a desvantagem de representar uma perturbação das condições naturais de execução de qualquer tipo de movimento, prejudicando as medidas e conclusões que podem ser obtidas, (POPOVIC, 2013).

Uma técnica que vem sendo estudada é a detecção de movimentos diretamente pelo rastreamento de posições do corpo a partir de imagens. Assim, utilizam-se redes neurais para rastrear imagens e posicionar centros estimados de articulações. A utilização de diversas câmeras com esse propósito permite a obtenção de coordenadas tridimensionais, (BULAT *et al.*, 2020; CHEN *et al.*, 2016; PARK; RAMANAN, 2015). Tais técnicas são chamadas de estimativa

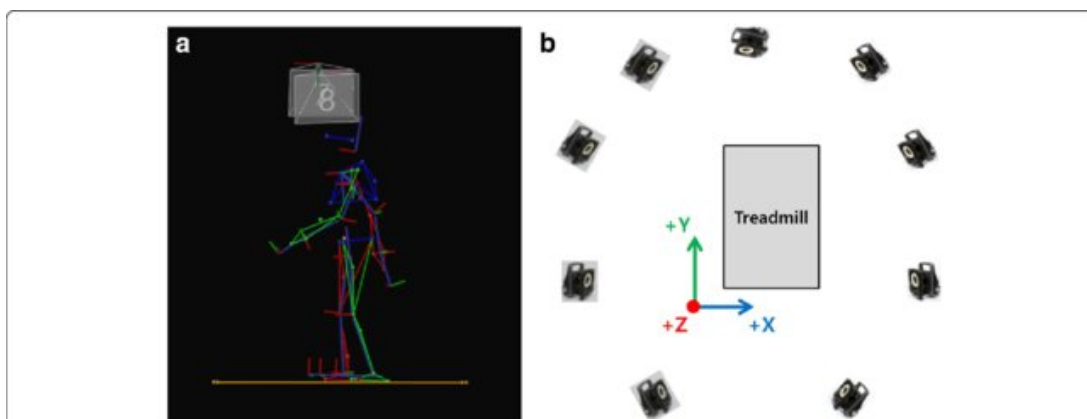


Figura 2.1 – Sistema de aquisição de movimento. Fonte: (LEE *et al.*, 2018)

de postura (*pose estimation*) e já são apresentadas como soluções para interfaces sem contato ou mais ricas do que as baseadas em controles com botões.

Uma forma intermediária está disponível nos sensores que obtêm mapas de altura para um ângulo de abertura e distância limitadas. Esses sensores ganharam projeção e escala de produção que os tornaram acessíveis com a indústria de jogos eletrônicos, (DARBY *et al.*, 2016). Também foram usados anteriormente sensores baseados em princípios inerciais em controles manuais, como o dos aparelhos da Nintendo. Esses controles usam acelerômetros e magnetômetros para estimar movimentos e direções de movimento com objetivo de controle de interfaces.

Aproveitando-se dessas ideias, usam-se os sensores atualmente embutidos em telefones celulares. Esses aparelhos também possuem acelerômetros, magnetômetros e giroscópios que podem ser usados para rastreamento de movimentos, (LANE *et al.*, 2010).

As aplicações citadas não atingem precisão e repetibilidade necessárias às pesquisas biomecânicas em caso geral. Porém, com uso de um esquema de calibração, podem ser usadas para tarefas simplificadas. Quando se deseja obter uma variável biomecânica condensada, por exemplo a inclinação de um membro, ou velocidade ou aceleração em uma direção, tais sensores podem ser ajustados em comparação com um sistema de aquisição de movimentos.

2.1.2 Simulação tridimensional

Os motores de desenvolvimento de jogos e de sistemas de simulação estão entre as ferramentas de software mais complexas do mercado. Entre os mais populares estão Unreal, Unity 3D, Godot, (LINTRAMI, 2018; KUCERA *et al.*, 2018; JULIANI *et al.*, 2018; HALLADAY, 2019; QIU; YUILLE, 2016; GRABSKA-GRADZIŃSKA *et al.*, 2020). Sua aplicação em problemas científicos é crescente e já possui algum histórico, (LEWIS; JACOBSON, 2002; FRIESE *et al.*, 2008).

Todos eles são programas construídos para a simulação de mundos virtuais que representam a realidade de diversos jogos. Permitem ajustes do funcionamento desses mundos virtuais para representar físicas mais realistas ou menos conforme se adequem a requisitos lúdicos dos projetos, (GUANA *et al.*, 2015).

Essa versatilidade é útil para os objetivos de treinamento de redes neurais pois permitem ajustar o nível de complexidade e detalhe que se deseja usar para obter valores de variáveis e medidas de simulação, como por exemplo em (GECGEL *et al.*, 2019).

Simplificadamente, tais sistemas desenham na tela as projeções de formas geométricas geralmente tridimensionais usando a geometria projetiva. Essa forma de representação permite velocidade de cálculos geométricos e projeções, usando representações que evitam cálculo de divisões, por exemplo, (GREGORY, 2018).

2.1.3 Análise e classificação de imagens por visão

O uso de imagens para diagnóstico de processos ou obtenção automática de informações é baseada em longo histórico de pesquisas em processamento de sinais. Isso foi possível através de técnicas que combinam a teoria de filtros e geometria.

Após essa etapa, a pesquisa passou a incluir abordagens heurísticas que envolviam a definição de medidas abstratas sobre as imagens, chamadas de *características (features)*. Sobre essas medidas, aplicam-se as técnicas do aprendizado de máquinas para obter informações. É essa abordagem que permitiu que a detecção de formas geométricas complexas bem como reconstrução estereoscópica se tornassem possíveis como aplicações práticas.

2.2 Posicionamento dinâmico

O problema de posicionamento dinâmico é central tanto à biomecânica quanto à robótica. No primeiro caso para análise e no segundo para a síntese. Esse problema é geralmente abordado usando sensores que avaliam diretamente grandezas cinemáticas, como posição linear, angular, velocidades e acelerações, (MCGINLEY *et al.*, 2009; HEDRICK, 2008).

Porém, medidas indiretas se tornam possíveis a partir do processamento das informações complexas. Por exemplo, imagens pode ter sua informação explorada usando as técnicas de visão computacional, (Wu *et al.*, 2015; MA *et al.*, 2018).

A tecnologia dos sensores inerciais também é parte fundamental de uma renovação das medidas de posicionamento, que fazem parte de aparelhos de telefone e também de veículos aéreos, (Flayols *et al.*, 2017; MESSAY *et al.*, 2016; Yi *et al.*, 2009).

2.3 Inteligência Artificial

As definições para o campo da inteligência artificial são muitas e amplas. Havendo inclusive sobreposição com o aprendizado de máquina. Originalmente pensada como uma área que estudasse sistemas artificiais que exibissem comportamente inteligente que se assemelhasse aos organismos naturais em algum contexto. Atualmente, consiste de um amplo conjunto de técnicas e teorias orientadas para produzir respostas automáticas em problemas mal definidos matematicamente mas que possuam alguns tipos de padrão, (PANCH *et al.*, 2018; DOBREV, 2012; MARCUS, 2018).

O debate é ainda aberto sobre a medida em que as técnicas da inteligência artificial representam uma técnica do tipo *caixa preta* ou se representam em si uma outra forma de representação de conhecimento alternativa às formulações matemáticas baseadas em cálculo e álgebra.

Porém, para os fins deste trabalho, trata-se a inteligência artificial como uma forma baseada em otimização de erros de lidar com informação incompleta sobre o que se está estu-

dando concentrando a entropia de informação que não seja modelada, (VISVIKIS *et al.*, 2019; YOSINSKI *et al.*, 2015; HELM *et al.*, 2020).

2.4 Aprendizado de máquina (*Machine learning*)

O aprendizado de máquina é geralmente classificado como um subconjunto das técnicas de inteligência artificial. Enquanto a IA tenta reproduzir comportamento inteligente, o aprendizado de máquina almeja repetir padrões que possam ser capturados a partir de um volume de dados. Para repetir esses padrões, usam-se diversas técnicas de aproximação e otimização, (MICHALSKI JAIME G. CARBONELL, 1984).

2.5 Redes neurais

Aqui usamos especificamente as redes neurais como única técnica representativa da inteligência artificial e do aprendizado de máquina. Trata-se de uma técnica bastante desenvolvida para generalizar comportamentos de sistemas naturais quando usada como inteligência artificial, ou como uma forma de interpolação quando em situações de disponibilidade limitada de informação, no sentido do aprendizado de máquina, (HAEBERLE *et al.*, 2019; TOPOL, 2019; NAYLOR, 2018).

As arquiteturas são bastante variadas configurando muitas possibilidades de uso. Essa variação de arquiteturas é devido ao fato de que cada uma dessas é mais apropriadas para um grupo de problemas semelhantes.

2.6 Aprendizado profundo (*Deep learning*)

Dentro do contexto das redes neurais, os desenvolvimentos recentes de processadores gráficos permitiu a criação de pacotes computacionais que fizessem uso de arquiteturas com muitos níveis, ou camadas. Diz-se que essas arquiteturas são de aprendizado profundo. É comum que sem utilizadas mais de 30 camadas para uma rede neural detectar rostos ou gestos na interface de equipamentos comerciais como câmeras digitais e controles de acesso, (GOOD-FELLOW *et al.*, 2016).

2.7 Visão computacional

O aprendizado profundo em redes neurais abriu novos caminhos dentro da área de visão computacional pois permitiu a detecção de características e objetos complexos sem que houvesse a necessidade de interpretá-los como formas definidas ou relações de filtros de imagem. Diversas aplicações atuais de reconhecimento de objetos e classificação automática de imagens são viabilizadas por esse avanço.

Como outros exemplos, diversas aplicações em áreas de clínica médica e saúde pública são viabilizadas como visão computacional obtida com redes neurais profundas, (BINI, 2018).

2.8 Linguagens e pacotes

As técnicas de treinamento e aplicação de redes neurais podem ser testadas diretamente em pacotes amplamente disponíveis. Há poucos anos os estudos de algoritmos para comparar treinamentos precisavam ser programadas diretamente e de forma dependente da estrutura da rede. Os pacotes recentes são generalizadas para se adaptarem à estrutura da rede especificada usando uma linguagem de alto nível através do conceito de tensor. Os pacotes mais usados são TensorFlow ((ABADI *et al.*, 2016; DILLON *et al.*, 2017; GOLDSBOROUGH, 2016)) e PyTorch ((PASZKE *et al.*, 2019; KETKAR, 2017)). Essas ferramentas são desenvolvidas em linguagens de baixo nível mas todas são operadas por meio de pacotes de acesso a aplicações em linguagens modernas. A linguagem Python é a mais comum e usada neste meio. Alguns ambientes de desenvolvimento como Matlab, Scilab e R também permitem treinamento com capacidade limitada para algumas estruturas de rede, (BERGMEIR; SANCHEZ, 2012; PRIYADARSHNI; SOHAL, 2016; VEDALDI; LENC, 2015).

3 SIMULAÇÃO DE AMBIENTES TRIDIMENSIONAIS

3.1 Introdução

O uso de sistemas de simulação é já técnica comum na engenharia em projetos mais cuidadosos, (FUJIMOTO *et al.*, 2017). A função esperada de tais sistemas é de antever valores de grandezas físicas com precisão conhecida. Permite avaliar a sensibilidade de técnicas e projetos como um todo. Para tanto o sistema de simulação contém relações quantitativas necessárias na forma de modelos que digam respeito ao que se está modelando ou analisando.

Essa abordagem é universal em áreas como projeto estrutural (CRAIG; KURDILA, 2006), controle de sistemas dinâmicos (BOSCH; KLAUW, 1994), sistemas regidos por fenômenos de transporte (CHUNG, 2010), entre outros. Esses sistemas se tornam especialmente úteis quando usados em conjunto com experimentos práticos que são representados por seus modelos, ainda que sejam modelos de escala reduzida. A principal barreira conceitual que permitem vencer é a de medir grandezas em locais e naturezas inacessíveis para a simples medida usando sensores. Por exemplo, um modelo de uma planta de destilação pode avaliar temperatura e pressão em pontos intermediários do processo termodinâmico sendo conduzido em locais em que não se possam instalar sensores por limitações tecnológicas, (JANA, 2018).

Nesse sentido, os modelos de simulação assumem a função de observadores. Eles permitem estimar grandezas que não podem ser medidas diretamente usando conhecimento da dinâmica do sistema e medidas de outros pontos que sejam acessíveis, (TRINH; FERNANDO, 2012).

Tais modelos podem ser descritos usando os princípios físicos das áreas coerentes de engenharia. Esses princípios são descritos quase sempre pelas ferramentas matemáticas das equações diferenciais. Esses modelos podem ter encontrados com uma combinação entre as equações e os dados experimentais. Esse é o conteúdo da teoria da identificação, (ZADEH, 1956; LJUNG, 2017; LJUNG *et al.*, 2020; KOCIJAN *et al.*, 2005).

Em contrapartida, podemos acrescentar os modelos obtidos de forma comparativa, ou seja, que sejam descritos por relações matemáticas que não tenham interpretação direta, mas que são ajustadas para que correspondam às situações realistas a partir de medidas. Modelos desse tipo já foram chamados de *caixa preta*¹ pela ideia de que produzem resposta com entradas fornecidas mas suas relações internas são opacas quanto aos princípios físicos que descrevem. Modernamente, estão associados a técnicas como aproximações lineares ou polinomiais, exponenciais, e principalmente para este trabalho, por redes neurais. O uso de redes neurais

¹ Essa denominação também é usada dentro da teoria da identificação para modelos que sejam descritos por formatos matemáticos conhecidos, embora com aproximações vagas, sem conhecimento físico de cada parâmetro.

como formas de modelagem direta já tem alguma literatura desenvolvida, como os trabalhos (THEOCHARIS; PETRIDIS, 1994; ABDOLLAHI *et al.*, 2006; KIM *et al.*, 1997).

Os sistemas de simulação podem funcionar como auxiliares nesse processo. Na medida em que produzem valores de variáveis dinâmicas, eles podem ser usados para ajustar os modelos de redes neurais para que se separem de forma interessante as partes conhecidas e ignoradas do comportamento de sistemas dinâmicos. Esse argumento é inspirado em resultados recentes que mostram que sistemas baseados em redes neurais podem reproduzir comportamentos de sistemas do mundo real mesmo quando treinados por simulações que são de toda forma menos complexas do que a realidade que representam, (GECGEL *et al.*, 2019; SHI *et al.*, 2020; NAGABANDI *et al.*, 2020; ANDRYCHOWICZ *et al.*, 2020).

Com essa linha de raciocínio, torna-se muito útil o uso de sistemas de simulação já plenamente desenvolvidos para representar famílias de problemas e principalmente, obter dados que permitam treinar tais redes neurais. No caso deste trabalho, será usado um motor de desenvolvimento de jogos que habilita a construção de ambientes virtuais com riqueza de detalhes que pode ser ajustada conforme a necessidade da aplicação e que permite qualquer formato de obtenção de medidas dos sistemas que simula.

3.2 Motores gráficos e de física

Todos os objetos tridimensionais em computador, são representados por combinações de pontos e formas que intercalam esses pontos, como planos, curvas paramétricas, esferas, entre outros. Evidente que tais formas são abstrações que facilitam o trabalho de quem modela, pois do pontos de vista do cálculo, o programa converte todas as formas geométricas em planos para o cálculos de colisões e projeções em tela. O número e divisão desses planos depende da precisão que se configura para o resultado da simulação, (LINTRAMI, 2018).

Sobre essas superfícies, decidem-se parâmetros de aparência como luminosidade a partir de fontes simuladas de luz e seu efeito sobre essas superfícies. Esses efeitos incluem reflectância, difração e cor. Além dessas definições de aparência, os modelos de elementos tridimensionais possuem uma movimentação física em um sistema de coordenadas. Os sistemas desse tipo atribuem sistemas de coordenadas em todos os objetos. A movimentação é influenciada por equações calculadas a cada instante de um tempo de simulação. As equações são programadas nos próprios motores e possuem parâmetros ajustáveis, permitindo variar gravidade, viscosidade de fluidos, entre outros. Evidente que isso se reproduz também nos objetos, variando atrito e inércia, por exemplo.

As interações de corpos são reguladas pelas equações e disparadas pela detecção de colisões entre objetos. Os efeitos das colisões podem ser programados livremente usando os *scripts*, (THORN, 2015).

Neste trabalho, como se almejam treinar redes neurais para identificar proprieda-

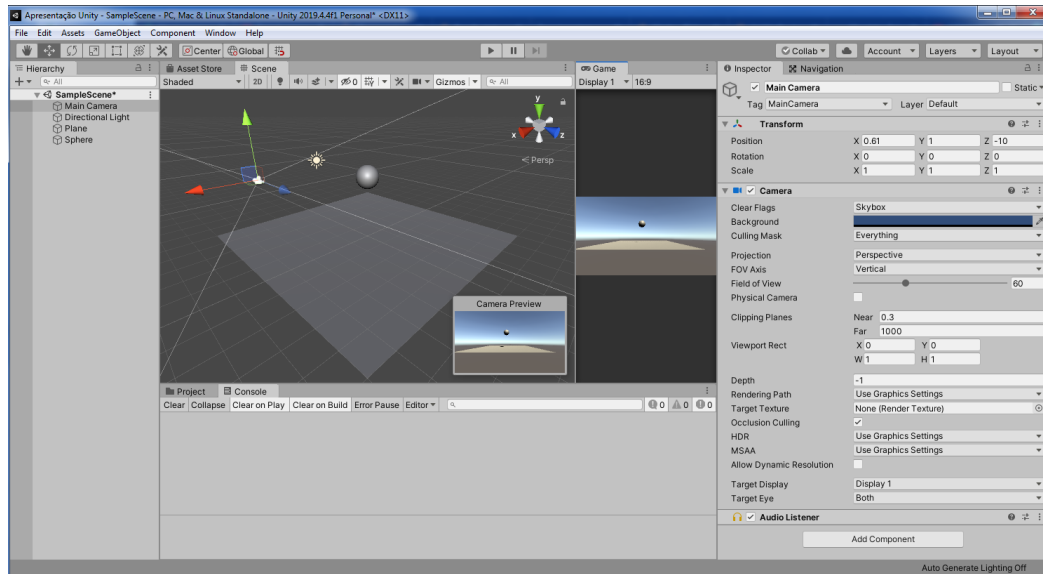


Figura 3.1 – Ambiente de desenvolvimento com suas abas de inspeção.

des em imagens, produziram-se modelos com visual relevante e uma câmera virtual era posicionada de forma conveniente para que imagens fossem capturadas e salvas em arquivos. Assim, tais arquivos continham exemplos de fotos que se pretendiam usar para treinar redes neurais. Como as imagens foram produzidas em simulação, estavam disponíveis medidas de quaisquer variáveis dinâmicas ou de posição para que as imagens fossem automaticamente rotuladas com a informação relevante ao treinamento. Ou seja, produziam-se os conjuntos de dados necessários ao treinamento que seriam muito difíceis ou impossíveis de se obter em teste real.

Tais técnicas tendem a ser tornar muito frequentes em projetos que exigem medidas complexas como indicam trabalhos recentes, (QIU; YUILLE, 2016; ULLMAN *et al.*, 2017), desde que sejam conhecidos os parâmetros que se deseja generalizar e que sejam corretamente modelados os comportamentos que se deseja simular.

3.3 Exemplo de simulação

Apresenta-se na sequência um exemplo ilustrativo de modelagem e obtenção de dados a partir de um motor de jogos. Imagina-se o problema de obter imagens do fenômeno de uma bola com elasticidade solta a partir de uma altura inicial e desejam-se obter imagens de vários estágios da queda e subsequentes repiques sobre o solo bem como as alturas associadas e os instantes de tempo em que ocorrem.

3.3.1 Montagem de cenário

Cenários são combinações de modelos tridimensionais. São geralmente feitos com combinações de outros modelos prontos. Usamos aqui um modelo de um plano de duas dimensões para representar o solo e uma esfera para representar a bola. Não serão modeladas

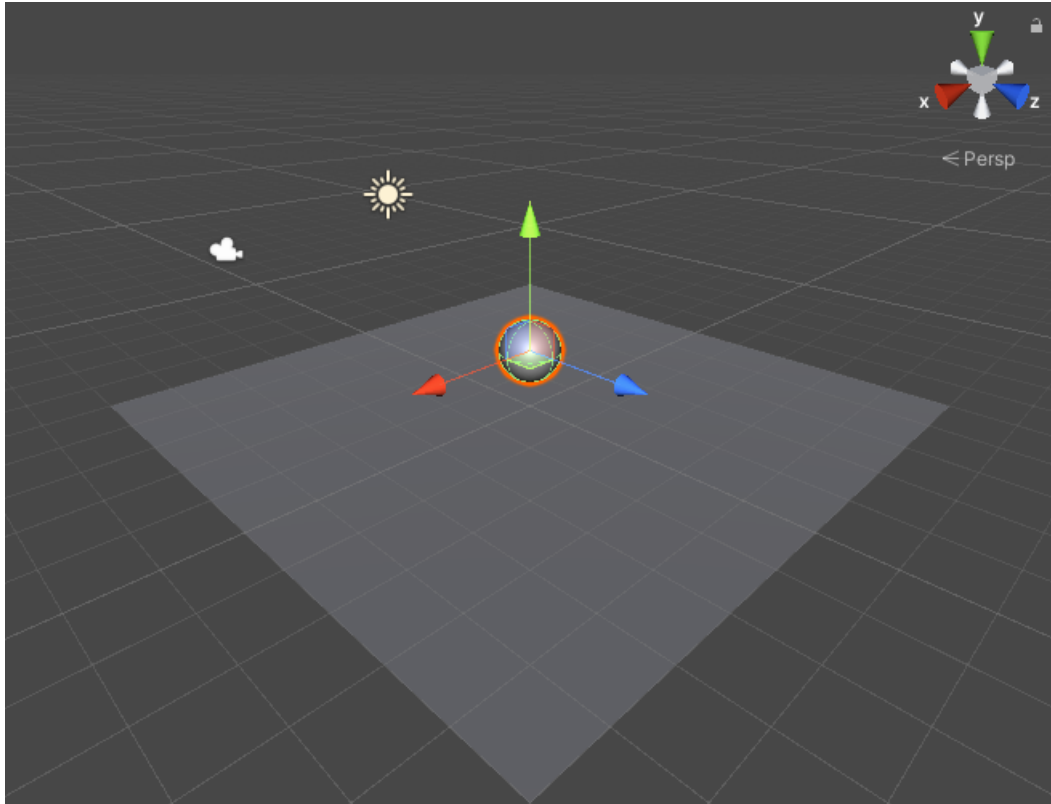


Figura 3.2 – Cenário com um plano e uma esfera rígida com massa. A localização de uma fonte de luz e de um ponto de vista (câmera) está visíveis, além do sistema de coordenadas associado.

deformações aqui por simplicidade. A figura 3.2 ilustra o cenário.

3.3.2 Definições físicas

Atribui-se ao solo e à bola as propriedades de colisão que impedem que suas formas invadam os volumes um do outro e permitem a detecção da colisão entre eles. À bola, atribui-se uma massa, que resultará em uma trajetória automática de queda quando posicionada em um local com gravidade ativa e livre de superfícies que restrinjam seu movimento. As propriedades físicas da esfera, como de qualquer objeto do tipo corpo rígido, podem ser definidas em um painel como o da figura 3.3.

3.4 Aplicação de modelos dinâmicos

Inserir-se um comportamento programado no detector de colisão que permitirá estabelecer o que ocorre quando os objetos entram em contato. Neste caso, atribui-se uma elasticidade à bola, pois produz-se uma força oposta que é proporcional à velocidade de colisão e uma perda de velocidade que corresponde a perda de energia por atritos com o solo, som e comportamento visco-elástico da bola. Nesse ponto, entram as considerações sobre o realismo do modelo e também sobre a necessidade de mais ou menos detalhes para os fins projetados.

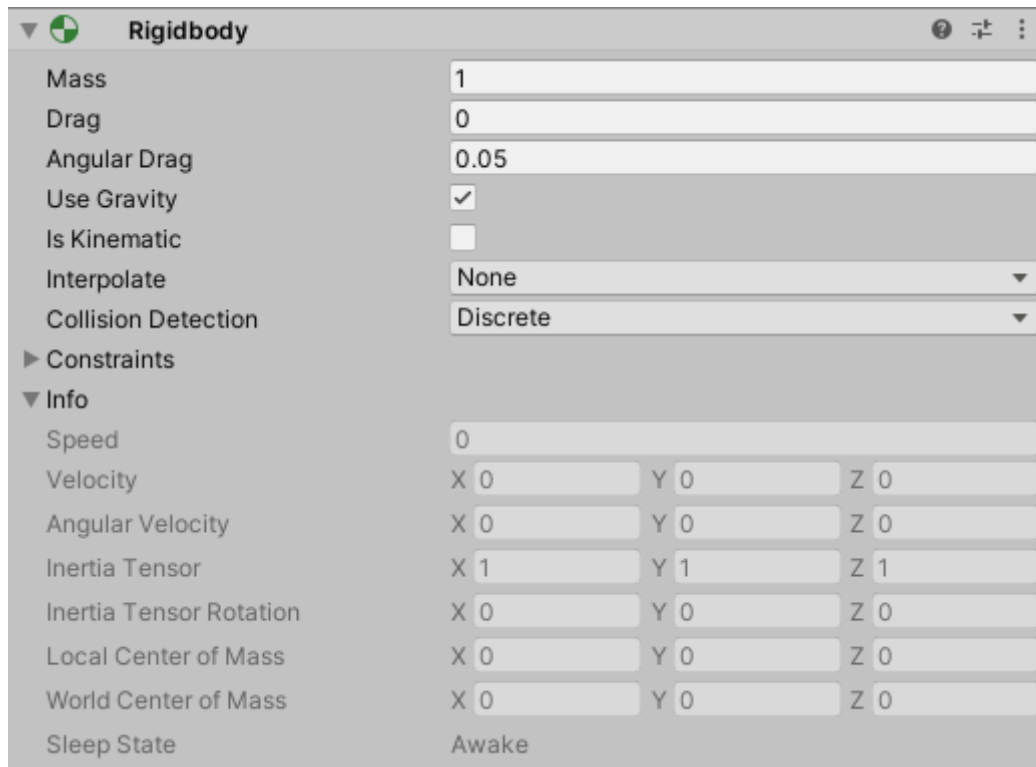


Figura 3.3 – Painel de propriedades físicas.

Neste exemplo, usa-se o método que detecta colisões para aplicar uma inversão de velocidade vertical. Essa inversão decresce com o passar das colisões, simulando uma perda de energia. Uma variável conta o número de colisões e o método que é automaticamente chamado quando colisões ocorrem deve ser preenchido com o código para esse comportamento. O código mostra também métodos que ocorrem em todos os objetos de jogo, como o Start que é executado quando o objeto é criado e Update que é executado a cada atualização do que se vê na tela.

O código que deve ser acrescentado ao objeto da esfera é o seguinte:

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class SpringBall : MonoBehaviour
6 {
7
8     public Rigidbody rb;
9     int i;
10
11     // Start is called before the first frame update
12     void Start ()
13     {

```

```

14     rb = GetComponent<Rigidbody>();
15     i = 0;
16 }
17
18 void OnCollisionEnter(Collision c) {
19     rb.velocity = new Vector3(0f, 5f-0.5f*i, 0f);
20     i++;
21 }
22
23 // Update is called once per frame
24 void Update()
25 {
26
27 }
28 }

```

3.4.1 Leitura de variáveis dinâmicas

A cada intervalo de simulação, pode-se programar que uma imagem da câmera seja capturada e seja simultaneamente salva a posição em que a bola se encontra a partir da translação do objeto que a representa. Isso pode ser feito na forma de um script sobre o objeto da câmera.

Neste caso, uma outra variável pode ser acrescentada para acompanhar as atualizações de cena e disparar a captura de uma imagem de tela. A atualização do código é mostrada a seguir.

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class SpringBall : MonoBehaviour
6 {
7
8     public Rigidbody rb;
9     int i;
10    int k;
11
12    // Start is called before the first frame update
13    void Start()
14    {
15        rb = GetComponent<Rigidbody>();
16        i = 0;
17        k = 0;
18    }

```



```
19
20     void OnCollisionEnter(Collision c) {
21         rb.velocity = new Vector3(0f, 5f-0.5f*i, 0f);
22         i++;
23     }
24
25     // Update is called once per frame
26     void Update()
27     {
28         if(k++ % 10000 == 0) {
29             ScreenCapture.CaptureScreenshot ("imagem"+k);
30         }
31     }
32 }
33 }
```

3.5 Conclusões

Os sistemas de motores de jogos são ferramentas avançadas que podem ser usadas para produzir situações de simulação não apenas para avaliação quantitativa e previsão de fenômenos reais, mas também para produzir versões simplificadas da realidade que podem ser usadas para treinar modelos de redes neurais. Isso permite que as complexidades da realidade sejam ignoradas ou modeladas de forma controlada.

4 INTELIGÊNCIA ARTIFICIAL APLICADA A SISTEMAS MECATRÔNICOS

Neste trabalho utilizamos as redes neurais como técnica de inteligência artificial que tem como objetivo lidar com a informação incompleta dos modelos ou com a riqueza de entropia informacional de sensores que dificultam a leitura indireta de variáveis dinâmicas.

4.1 Redes neurais

4.1.1 Definições

Uma rede neural é uma estrutura que relaciona entradas e saídas de forma agnóstica para reproduzir e generalizar um comportamento conhecido que seja objetivo de análise ou interesse de projeto. Entre a entrada e saída há uma relação que define o que se chama de arquitetura da rede neural. As entradas são ponderadas por valores e agrupadas por funções não lineares. As saídas dessas funções, por sua vez, são entradas para outras ponderações até as saídas. Quando as conexões são todas na mesma direção do cálculo e se organizam em mesmo número de agrupamentos de funções, diz-se que redes neurais são organizadas em camadas, (GOODFELLOW *et al.*, 2016).

As conexões entre as unidades são intermediadas por pesos que regulam o comportamento da rede. O ajuste desses pesos é o que chamamos de treinamento. O treinamento é realizado variando os valores desses pesos ao submeter as entradas a valores de um banco de dados e comparando a saída obtida com valores esperados.

Esse ajuste corresponde a um processo de otimização que tem a proximidade entre as saídas obtidas pela rede e as desejadas como objetivo. Os pesos correspondem às variáveis de decisão que deve ser alteradas. As alterações usam algum tipo de método de gradiente, ou seja, variam-se os parâmetros na direção negativa do gradiente. Ao alterar os pesos da rede na direção negativa do gradiente, escolhe-se uma intensidade nessa direção conhecida como taxa de aprendizado.

Se uma camada é representada por uma associação de entradas ponderadas (x_i) por pesos w_i e uma função não linear $f_i(\cdot)$ para a saída vetorial x_{i+1}

$$x_{i+1} = f_i(w_i^T x_i).$$

Uma combinação de camadas entre uma entrada x e uma saída y é

$$y = f_N(f_{N-1}(\dots f_1(w_1^T x_1)))$$

O aprendizado consiste em ajustar os pesos w_i pela variação que eles produzem na saída y em relação ao erro de um valor desejado para a saída \tilde{y} . O valor de w_i em um momento

(iteração k) pode ser obtido de uma iteração anterior por esse ajuste.

$$w_i^k = w_i^{k-1} - \alpha \prod_{j=i}^N \nabla f_j(\cdot),$$

sendo os gradientes calculados em cada camada e α a taxa de aprendizado.

4.1.2 Arquitetura e treinamento

A arquitetura de uma rede diz respeito a quantas unidades internas que relacionam os pesos a uma saída existem e como são conectadas. De um ponto de vista teórico, prova-se que uma rede neural com uma camada de entrada, uma camada de saída e uma camada intermediária pode aproximar qualquer tipo de função de interesse bastando aumentar as unidades da camada intermediária. Na prática porém, tal resultado pode se tornar proibitivo devido à dificuldade de os algoritmos ajustarem os pesos, (JIANG, 2000; JIANG, 2019; COMBES *et al.*, 2018).

As diversas arquiteturas, adaptadas aos problemas estudados são todas treinadas escolhendo uma métrica para o desempenho, ou seja, um critério de proximidade com os resultados esperados e aplicando a derivada desse critério de forma retroativa sobre as unidades que o calculam ao longo da rede. Assim, definir a arquitetura da rede é definir as formas de conexão que as unidades contém e permitir que os pesos sejam ajustados por essas conexões.

As aplicações de redes neurais, organizadas em competições cristalizaram arquiteturas que podem ser baixadas e reutilizadas em outros projetos para novo treinamento ou mesmo mantendo o treinamento disponível publicamente. Essa técnica também permite usar uma rede neural treinada para uma outra aplicação e reutilizar os pesos exceto das últimas camadas. Esses pesos finais requerem novo treinamento, porém reaproveitam um conhecimento sintetizado nas camadas anteriores. Isso se chama transferência de conhecimento, (GOODFELLOW *et al.*, 2016; TAN *et al.*, 2018).

A transferência de conhecimento é possível pelo fato de as redes neurais possuírem uma propriedade de detalhismo crescente em suas camadas. Ou seja, as camadas mais próximas da entrada aparentemente obtém informações mais gerais sobre os dados, como formas geométricas, detecção de contornos, frequências mais baixas em sinais. Enquanto isso, as camadas mais próximas da saída capturam informações de detalhes, que são mais sensíveis ao erro e variações locais do gradiente.

Há duas modalidades de medidas de desempenho para o treinamento de redes neurais: são as baseadas na soma do erro quadrático entre as saídas desejadas e as previstas (*Mean squared error*); e as medidas baseadas em entropia cruzada, que tentam aproximar a informação contida nos sinais de resposta que se deseja reproduzir, (HOFFER; AILON, 2015).

Em princípio, a otimização variando os pesos poderia aplicar qualquer técnica de otimização. Na prática, evitam-se as técnicas baseadas em informações de segunda ordem, ou

seja, variações e adaptações do método de Newton, (BILAL *et al.*, 2018; XIE *et al.*, 2018; LI, 2009; LI *et al.*, 2015).

4.2 Interpretações

O processo de treinamento de redes neurais de médio e grande porte ainda é bastante empírico e mesmo a literatura mais especializada trata os ajustes de seus parâmetros para melhorar o processo de treinamento como um processo cheio de surpresas. É comum fazer variações aleatórias de dimensões de camadas intermediárias e usam-se de ideias bastante gerais sobre os algoritmos de treinamento. Por outro lado, também confiam em arquiteturas já usadas de forma bem sucedida em aplicações.

Algumas conclusões genéricas que são usadas no ajuste de tais redes podem ser coletadas:

- Problemas de classificação de imagens muito diferentes não apresentam nenhum tipo de convexidade de uma imagem para outra. Ou seja, imagens pertencentes ao mesmo conjunto de classificação não são combinações de outros elementos do conjunto. Classificar que uma imagem possui um cachorro e outra também não significa esperar que haja alguma combinação entre as imagens desses cachorros que parece também ser um cachorro. Por isso, é interessante que as direções de otimização sejam ajustadas por critérios baseados na informação estatística que vem dos dados. Nesse caso, os critérios de entropia cruzada, (DOUGLAS; BERARDI, 2005; ZHANG; SABUNCU, 2018), e entropia relativa, (WANG *et al.*, 2018; LIU *et al.*, 2017). São técnicas que aproximam a distribuição estatística dos dados de saída obtidos à distribuição estatística dos dados de treinamento e fazem os ajustes nas direções em que há maior divergência nesse sentido, (IMAZUMI; FUKUMIZU, 2019).
- Para problemas em que variações dos tipos de dados de entrada possuem uma relação de convexidade, os critérios de aprendizado mais adaptados permanecem os baseados em erro quadrático, (JIANG, 2019).
- Como já citado, em redes neurais de muitas camadas, há uma tendência empírica de que as primeiras camadas sintetizem informações mais gerais e amplas sobre os problemas. Exemplificadamente, contornos e cantos em redes para imagens ou respostas de modos lentos em sinais temporais. As camadas posteriores sintetizam informações mais específicas sobre as entradas como cores (quando houver) ou detalhes menores. Esta extrapolação se aproxima das técnicas de processamento de sinais que também procuram aproximar formas com ruído usando funções de base. Ou seja, comparam-se camadas de redes neurais a funções de base em técnicas como decomposição em modos empíricos ou transformada Wavelet, (TAN *et al.*, 2018; BIEBERICH, 2002).

5 ESTUDOS DE CASO

5.1 EC 1 – Fusão de dados em sensores inerciais

5.1.1 Descrição do problema

Neste capítulo, estudamos o problema da pirueta no balé (ANDRADE *et al.*, 2020) e também o problema da caminhada e da postura de passar de sentado para de pé, (PINTO *et al.*, 2020).

Em (ANDRADE *et al.*, 2020), cita-se que um dos parâmetros mais importantes do balé é a inclinação do tronco durante o movimento da pirueta, ilustrada na figura 5.1. Assim, a proposta é de usar um aparelho de celular preso ao tronco de um bailarino por uma cinta. Uma vez que o aparelho possui os sensores triaxiais de aceleração, campo magnético e velocidade angular, esses valores podem ser usados para estimar a inclinação do tronco e também a inclinação do eixo tridimensional de rotação.

Em (PINTO *et al.*, 2020), em que se estuda o controle motor da caminhada comum e do movimento de sentar e levantar, usa-se um objeto que deve ser controlado (uma caneca) e que tem um aparelho de telefone acoplado. A tarefa é ilustrada na figura 5.2. A manutenção do ângulo de inclinação desse objeto é também um indicativo de controle motor.

5.1.2 Divisão de modelo

A disponibilidade de todas essas nove medidas permite usar para obter uma relação mais estável para os valores que se deseja conhecer usando a fusão de sensores. A fusão usa as leituras de vários sensores em tempo real alimentados sobre um modelo dinâmico que visa absorver variações que fujam desse modelo como ruído indesejável. Essa fusão é comumente feita usando a técnica do filtro de Kalman.

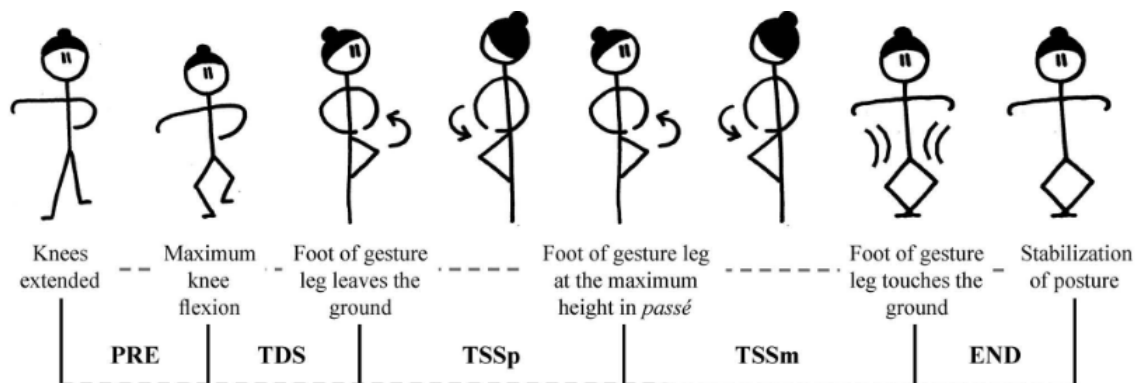


Figura 5.1 – Avaliação de pirueta no balé por meio do celular. (ANDRADE *et al.*, 2020)



Figura 5.2 – Tarefa de passar de sentado para de pé, avaliada usando um celular. Fonte: (PINTO *et al.*, 2020)

No filtro de Kalman, os dados dos sensores são fornecidos a um modelo dinâmico que produz um resíduo a ser minimizado, permitindo estimar variáveis não medidas diretamente por meio das equações do modelo. Ou seja, o filtro de Kalman é um observador robusto, (SEJNOWSKI; POGGIO, 2007).

Porém, nas aplicações aqui descritas, não há necessidade de medidas de tempo real, ou seja, todos os dados pode ser simultaneamente utilizados para estimar as medidas em todo os instantes de tempo. As equações do modelo podem ser satisfeitas na variável do tempo em qualquer ordem. Ou seja, como os dados são previamente medidos, antes do cálculo de uma variável não medida pelo observador, para a obtenção de uma variável a um tempo t , podem-se usar todos os dados para tempos anteriores ou posteriores. Assim, a detecção de uma variável observada nessas condições não precisa respeitar a causalidade do sistema que a produziu.

Assim, a abordagem deste trabalho é coletar os sinais disponíveis e tratá-los com filtragem e uniformizar suas taxas de amostragem. Isso ocorre porque os sensores inerciais disponíveis em telefones são chamados pelo sistema operacional em momentos definidos pelas demandas do sistema e suas sequencia de processos. Por isso, não possuem taxa de amostragem constante. Usando interpolação para a taxa de amostragem média do sensor mais lento dos disponíveis, essa questão é contornada, (CUESTA-VARGAS *et al.*, 2010; BULLING *et al.*, 2014; STANKEVICH *et al.*, 2012; ANDRADE *et al.*, 2020).

De posse desses sinais, podem-se usar as relações dinâmicas a que estão sujeitos pelo fato de serem produzidos por sensores presos ao mesmo corpo rígido. Os sensores disponíveis produzem leituras de variáveis dinâmicas em relação ao corpo do aparelho de telefone, e nisso reside sua principal dificuldade. O acelerômetro mede aceleração em relação ao um sistema de coordenadas preso ao aparelho e portanto que está sempre mudando em relação ao referencial inercial do ambiente. O giroscópio mede a velocidade angular com essa mesma característica.

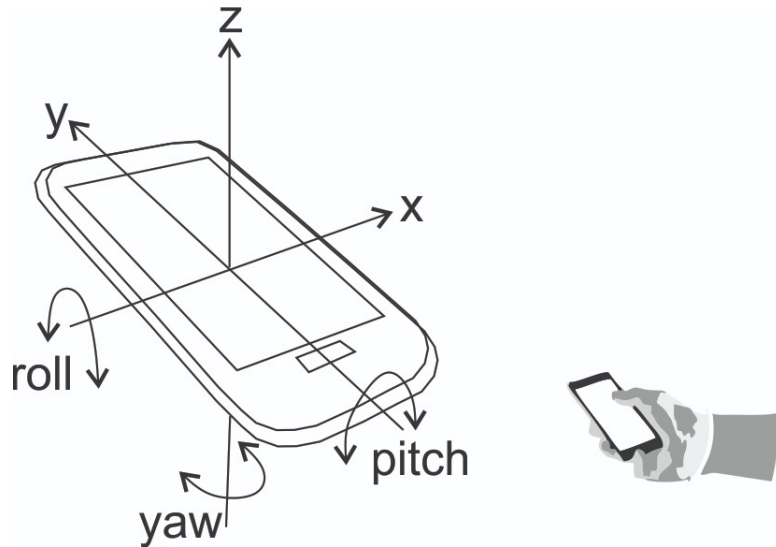


Figura 5.3 – O aparelho de celular possui sensores de aceleração, campo magnético e velocidade angular em relação aos três eixos.

O magnetômetro é o único sensor que possui uma relação direta com o referencial inercial. Isso se deve ao ato de medir campo magnético. Na ausência de campos intensos próximos, avalia o campo magnético da Terra em três eixos, que é constante para as distâncias consideradas de poucos metros. Assim, as variações desse campo representam variações do sistema de coordenadas do aparelho, ou seja, rotações puras. Por meio do magnetômetro apenas pode-se obter a medida do ângulo de inclinação e do ângulo de rotação desejados.

Quando se avalia a partir de uma média de alguns pontos o campo magnético medido, tem-se um vetor inicial \vec{m}_0 . O problema da álgebra de vetores de encontrar a rotação que converte um vetor em outro que tenham mesmo módulo pode ser aplicado, (LANDI; ZAMPINI, 2018). É útil dividir o módulo do vetor do campo magnético para maior estabilidade numérica, uma vez que apenas sua direção é necessária. A direção perpendicular entre os dois instantes do campo magnético pode ser obtida pelo produto vetorial entre ambos. A diferença entre o campo em um instante e o campo do instante inicial quando obtida por uma rotação corresponde à rotação oposta à do aparelho. Esse fato é consequência da constância do campo magnético da Terra.

Ao rescrever a rotação que leva um vetor $\vec{m}(t)$ a um \vec{m}_0 como a matriz $R(\vec{\theta})$, que é função de três ângulos de Euler $\vec{\theta} = [\theta, \rho, \psi]$.

$$\vec{m}(t) = R(\vec{\theta})\vec{m}_0.$$

O problema inverso obtido pelo produto vetorial denotado por Rot

$$\vec{\theta} = \text{Rot}(\vec{m}_0, \vec{m}(t)). \quad (5.1)$$

Por outro lado, a rotação é também medida indiretamente pelo giroscópio, a menos

do conhecimento do seu sistema de coordenadas em relação à referência inercial. Essa rotação é também, como consequência do funcionamento do magnetômetro, a taxa de rotação instantânea da direção do campo magnético em sentido oposto (o celular se gira numa direção, o sensor percebe o campo girando ao contrário). Assim, é possível estimar a velocidade angular nas coordenadas inerciais ($\vec{\Omega}(t)$) de duas formas. A primeira delas é a taxa de variação instantânea da rotação do campo magnético (compensando por seu módulo)

$$\vec{\Omega}_1(t) = \frac{1}{\|\vec{m}(t)\|} \frac{d\vec{m}(t)}{dt} \quad (5.2)$$

e também pelo próprio giroscópio, que medida em sistema de coordenadas local ($\vec{\omega}(t)$)

$$\vec{\Omega}_2(t) = R(\vec{\theta})\vec{\omega}(t) \quad (5.3)$$

que corresponde a uma outra medida, e portanto, outra fonte de informação a se considerar na fusão de dados.

Uma fusão de sensores numericamente estável deve fazer a diferença acumulada ao longo do tempo, ou em uma janela dele, de

$$e(T) = \left\| \int_0^T \frac{1}{\|\vec{m}(t)\|} \frac{d\vec{m}(t)}{dt} - R(\vec{\theta})\vec{\omega}(t) dt \right\| < \varepsilon \quad (5.4)$$

estar limitada a um valor máximo de erro (ε).

O cálculo de $\vec{\theta}(t)$ pode ser feito usando uma combinação das duas formas apresentadas (5.2 e 5.3). Para levar em conta a dinâmica do problema, é útil observar que a expressão para o erro 5.4 é convexa quando se consideram os termos de velocidade. Ou seja, espera-se uma variação do erro que cresce conforme se se afasta da relação correta para as variáveis obtidas.

As variações esperadas entre as medidas devem corresponder ao ruídos que podem se originar das diferenças entre os tempos de amostragem dos sensores mas também às dinâmicas eletrônicas de seus transdutores. Tais equipamentos podem apresentar histerese e atratores dinâmicos. Espera-se que tais efeitos sejam pouco relevantes nos intervalos de tempo considerados. Nesse sentido, propõe-se um tempo de observador não causal que estime a posição angular do aparelho usando todas as medidas disponíveis e colocadas representando uma relação que se aproxima da ideal de forma convexa. A forma convexa para o erro significa supor que suas variações quando combinadas oscilam em torno do valor esperado. Ou seja, neste caso, um estimador (h) para a inclinação tridimensional $\vec{\Theta}(t)$ usando $\vec{\theta}(t)$, $\vec{\Omega}_1(t)$ e $\vec{\Omega}_2(t)$.

$$\vec{\Theta}(t) = h(\vec{\theta}(t), \vec{\Omega}_1(t), \vec{\Omega}_2(t)) \quad (5.5)$$

lembrando que tal estimador pode usar valores de medidas em um horizonte de eventos antes e após t .

Uma abordagem direta sobre o problema seria a tratá-lo como minimização do erro $e(T)$ em que a variável de decisão sejam os ângulos θ . Como se trata de uma função, isso levaria

na prática a encontrar um conjunto de pontos para uma malha de tempo. Produze-se assim o triplo (três eixos) do número de pontos do tempo como variáveis para o problema de minimizar o erro. Essa quantidade de variáveis torna o problema intratável mesmo para um horizonte de tempo T curto.

5.1.3 Treinamento e teste de rede

Uma rede neural treinada para contemplar as variações inerentes ao sistema de medida pode se aproveitar das características discutidas na seção anterior usando um intervalo de tempo em que se toma a entrada dos sensores, que são por sua vez processadas para obter as três medidas vetoriais indicadas na equação 5.5. Ao fornecer como saída os três valores que estimam a posição angular em um instante de tempo consegue-se aproveitar o fato de que o modelo das relações entre os sensores já foi contemplado fora da rede neural, bem como estabelecer um continuidade entre os pontos de dados, já que se usam pontos anteriores e posteriores como alimentação da rede neural para produzir os valores de um instante de saída.

Os resultados de referência podem ser obtidos usando um teste de comparação. Um movimento de pirueta é realizado usando a tanto o aparelho celular quanto os marcadores do sistema de aquisição de movimento presos ao aparelho. Dessa forma, pode-se estimar os ângulos de inclinação e rotação. Os dados podem ser usados para o treinamento de uma rede neural usando os ângulos estimados como saída.

Porém, as medidas encontradas pelas equações 5.1, 5.2 e 5.3, que são obtidas com os sensores de forma independente representam a cinemática completa que se deseja medir. Supondo que os erros dinâmicos dos sensores sejam de natureza diferentes e independentes, o ruído será reduzido se ao longo do tempo se adotar a continuidade dos sinais e que essas variáveis medidas mantém as mesmas relações. A relação de velocidade angular é de grande importância e deve-se evitar o erro comum de relacionar a posição angular com a integração direta da velocidade angular. Essas grandezas não são relacionadas devido às relações de influência que um eixo tem sobre o outro neste caso e a não comutatividade de rotações em três dimensões. Essa questão é tratada com detalhes em (BOYLE, 2017).

Na área de sistemas especialistas, quando se coletam informações de diversos usuários para produzir sistemas de recomendação, tem-se o que se chama filtragem colaborativa. A solução de inteligência artificial mais comum para essas aplicações é o uso de um autoencoder, (HAGHIGHI *et al.*, 2019). O conceito de várias fontes de informação contribuindo para construir uma informação complexa pode ser reaproveitado aqui no sentido de que se usam diversos sensores para medir grandezas que são teoricamente redundantes. Porém, os sensores diferentes constituem pontos de vista diferentes sobre o funcionamento do sistema. Além disso, o autoencoder é uma arquitetura comum para interpolar dados, (BERTHELOT *et al.*, 2019).

Usando um intervalo de pontos das três medidas citadas, a cinemática que se deseja

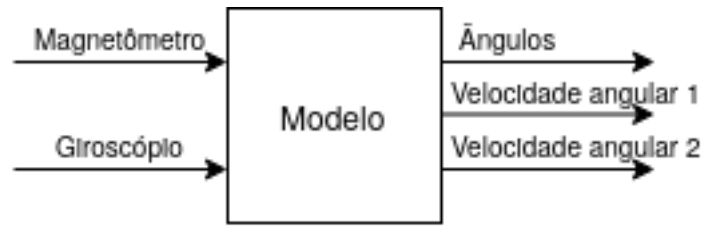


Figura 5.4 – A função das equações de modelo apresentadas é de calcular uma estimativa para os ângulos de rotação e duas estimativas independentes para a velocidade angular.

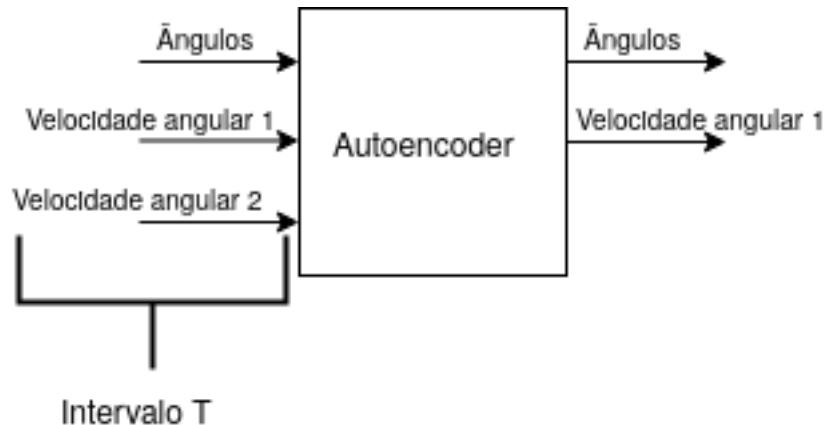


Figura 5.5 – O autoencoder tenta reproduzir a saída usando um intervalo de tempo de comprimento T dos dados obtidos pelo modelo. A estimativa da velocidade angular na saída é feita pela média das duas. $\Omega(t) = \Omega_1(t) + \Omega_2(t)$.

modelar está incluídas nessas equações e as relações entre elas dadas pelo sensores estão incluídas quando o autoencoder é treinado para reproduzir as mesmas relações para várias entradas dos mesmos tipos.

A sequência de passos para essa solução pode ser resumida como segue:

- Treinar uma rede neural do tipo autoencoder usando dados obtidos de um modelo.
- Usar a rede treinada sobre os mesmos dados para obter uma relação melhorada dos valores de rotação e velocidade angular.

Para treinar a rede, os passos são os seguintes:

- 1 Calcular a estimativa de ângulo de rotação como em 5.1.
- 2 Estimar a velocidade angular como em 5.2.
- 3 Estimar a velocidade angular como em 5.3.
- 4 Escolher uma janela de tempo T para tratar os sinais. Deve contemplar a dinâmica dos sinais. Escolher também o número de unidades na camada intermediária do autoencoder.

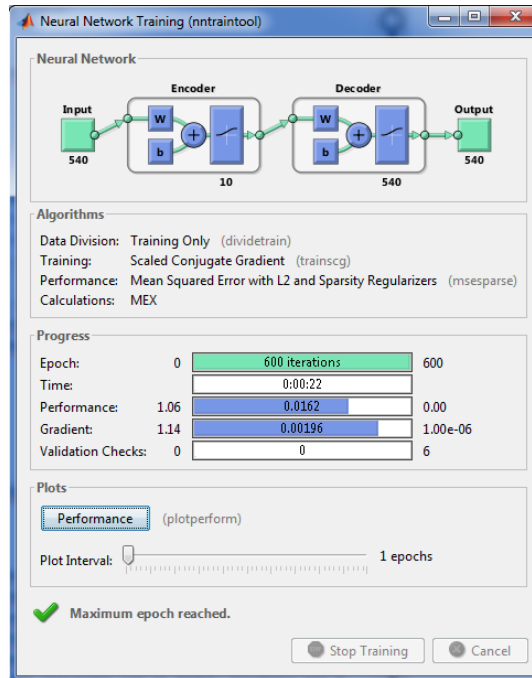


Figura 5.6 – Interface de treinamento para o autoencoder e outras formas de redes neurais com Matlab.

- 5 Empilhar as medidas dos itens 1, 2 e 3 em blocos de tamanho da janela.
- 6 Variar um ponto a cada janela, a sobreposição garante a continuidade dos dados.
- 7 Aplicar o treinamento do autoencoder para os dados empilhados. Como o autoencoder deve produzir duas saídas próximas, toma-se a média das duas.

5.1.4 Resultados

Houve um teste apresentado no trabalho (ANDRADE *et al.*, 2020) que comparou as medidas da técnica apresentada com as obtidas por meio de um sistema de aquisição de movimento. Ali, verificou-se proximidade com os dados experimentais. No casos dos dados aqui apresentados medidos apenas com o celular, não há comparação direta de erro a ser feita. Porém, como citado, um indicador de erro é a da distância entre as diferentes formas de medidas velocidade angular. A hipótese necessária para isso é a de que os sensores diferentes possuem tipos diferentes de erro.

O apêndice A apresenta os detalhes de implementação. A figura 5.6 ilustra um erro médio ao longo das iterações de treinamento. A média de erro quadrático médio para o ajuste dos dados pelo autoencoder foi de 0,0164 com um gradiente na otimização da rede neural médio de 0,00106 após 600 iterações. Evidente que é preciso treinar um autoencoder para cada grupo de dados.

A norma da diferença entre os sinais de velocidade angular calculados pelas relações 5.2 e 5.3 sem o uso do autoencoder atinge um valor médio de 36,895 rad em comparação

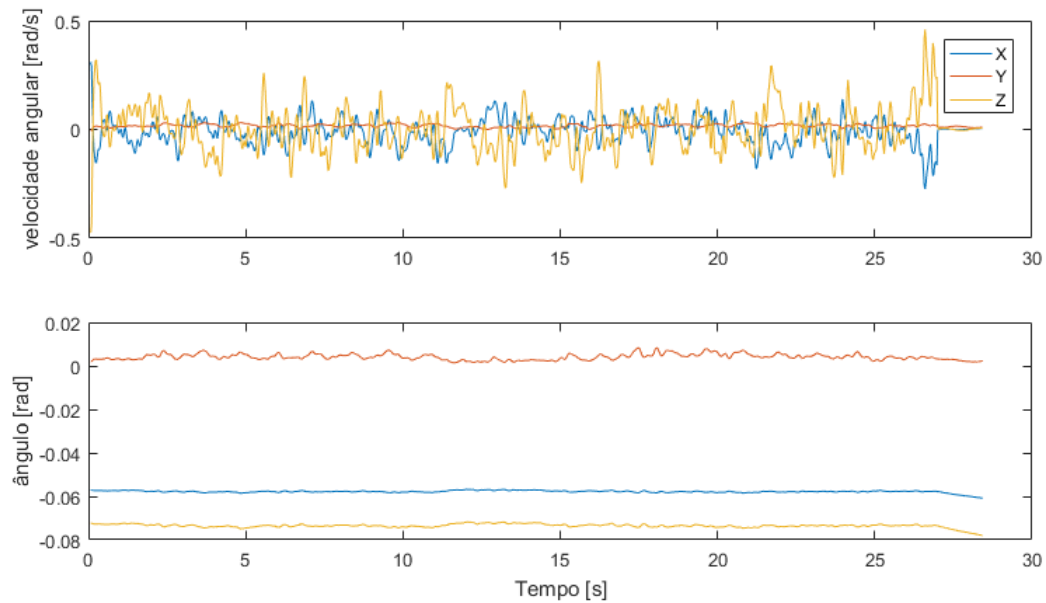


Figura 5.7 – Gráficos comparativos para um teste de estimativa com ângulos e velocidades angulares.

com um valor 3,8975 rad com o uso do autoencoder. Isso mostra que o tratamento dos sinais por um tipo de rede neural foi fundamental para atingir precisão.

Os resultados na figura 5.7 mostram a usabilidade prática da técnica e estabelecem a técnicas de redes neurais como uma forma de combinação de modelos com medidas experimentais especialmente na situação de fusão de dados de sensores diferentes.

5.2 EC2 – Falhas em painéis fotovoltaicos

5.2.1 Descrição do problema

Os painéis para geração de energia fotovoltaica estão crescentemente em uso com propósito de microgeração distribuída e também para fornecer energia elétrica em locais distantes de linhas de alta tensão, (CUCCHIELLA *et al.*, 2016; ZAHEDI, 2006).

Sua vida útil atingiu o ponto de compensarem seu alto custo de instalação. Uma dificuldade de sua manutenção quando instalada em locais de acesso difícil é a substituição de elementos defeituosos, (CARNEVALE *et al.*, 2014; RAUGEI; FRANKL, 2009).

As falhas de desgastes são usualmente visíveis como marcas aproximadamente circulares mais claras. Essas marcas também ficam mais nítidas em imagens de espectro infravermelho, como na figura 5.8. Isso ocorre porque pontos de conversão ineficiente ou inexistente da luz solar devem aquecer mais. Ou seja, a ausência de efeito fotoelétrico que direcione a energia dos fótons de luz faz que mais energia se converta em calor.

O diagnóstico de falhas em painéis fotovoltaicos em instalações pouco acessíveis

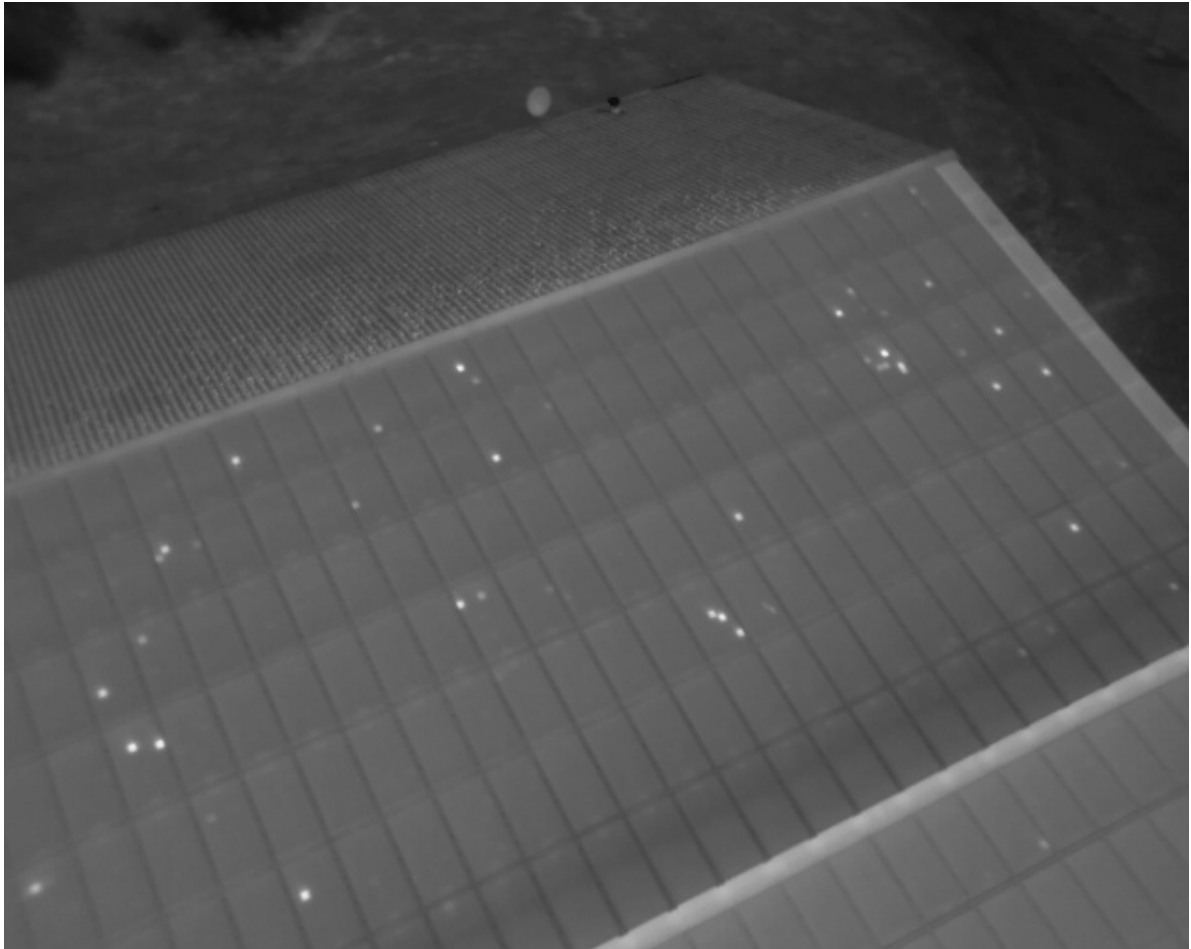


Figura 5.8 – Falhas visíveis em painel fotovoltaico usando foto de espectro infravermelho.

usa imagens obtidas com drones, (HENRY *et al.*, 2020). Os drones podem ser equipados com câmeras infravermelhas para aproveitar o efeito mencionado, (ALSAFASFEH *et al.*, 2018). Porém, as imagens obtidas por drones são observadas individualmente para o diagnóstico. Nesse sentido, este trabalho propõe técnicas que combinam redes neurais e informações do cinemáticas do drone para a localização de falhas.

O problema de localização de falhas poderia ser dividido em dois: o de localizar as falhas na imagem e o de localizar os painéis fotovoltaicos. O cruzamento dessas duas respostas permite localizar as falhas com uma localização aproximada dentro dos painéis, bem como garantir maior robustez ao método, pois falhas localizadas fora da área dos painéis não são falhas de fato.

5.2.2 Localização de falhas na imagem

Uma imagem gerada por drone geralmente é enviada sem fio para um computador pelo programas já usados para o controle de drones. Disponível assim para automação de um processo de análise imediatamente após a captura.

Uma imagem típica capturada por um drone com a presença de falhas apresentará

o painel com pontos mais claros, como na figura 5.8 em que a imagem apresentada é de uma fotografia por infravermelho.

Conforme discutido no capítulo 4 sobre redes neurais, a localização de objetos extensos em imagens depende de um conjunto volumoso de imagens com esses objetos marcados. Ou seja, de uma entrada que contenha a posição na imagem do que se deseja localizar. Para o treinamento de redes neurais, é comum que isso seja na forma de uma máscara, isto é, de uma matriz com as mesmas dimensões em pixels que a imagem. Para cada posição da matriz há uma marcação de que aquele pixel representa um ponto do objeto a ser detectado ou não. Em alguns casos podem-se também definir pixels como pertencentes à fronteira do objeto. Quando se definem apenas duas possibilidades (pixel do objeto ou não) a matriz pode ser armazenada como uma matriz de valores booleanos.

A produção de imagens e de rótulos para o treinamento de uma rede neural, como em outros problemas, é custosa. Cada imagem precisaria ter seus pixels marcados manualmente para definir a presença das falhas. Trata-se de mais um contexto para a utilização de imagens sintetizadas em computação gráfica como entradas de treinamento usando o conceito de um mundo simulado como apresentado no capítulo 3.

5.2.3 Localização da imagem no sistema do drone

Todos os drones possuem sistemas de navegação complexos que se baseiam em fusão de sensores, (DAPONTE *et al.*, 2015). A tecnologia já tem maturidade permitindo conhecer de forma precisa a posição em coordenadas geo-estacionárias e outros parâmetros cinemáticos como altitude e inclinação, (SORBELLI *et al.*, 2018). Um esquema de funcionamento do sistema é apresentado na figura 5.9.

A posição sobre o solo e os parâmetros cinemáticos de altitude rotação e inclinação são conhecidos com ordens de grandeza diferentes. A posição é obtida pelo sinal de GPS, e os outros parâmetros pelas leituras de acelerômetros e magnetômetros. A fusão de sensores é usada com filtros de Kalman para obter essas medidas e também suas derivadas como velocidades lineares e angulares, (DAPONTE *et al.*, 2015).

Usando esses parâmetros, a direção de visada da câmera é resultado direto das medidas do drone. E portanto, artefatos localizados em uma imagem capturada pela câmera podem ter sua posição estimada em relação ao mundo. Ao detectar uma falha, através da câmera e localizá-la em termos da posição na imagem, pode-se obter a localização da falha no painel usando a separação proposta pela rede neural. Além disso, a localização do painel na imagem, permite a determinação de que painel se trata em um banco de dados que tenham localizações de painéis registradas.

A localização do painel identificado em uma imagem pode ser extrapolada para uma localização no mundo usando a projeção do sistema de coordenadas da câmera para o sistema

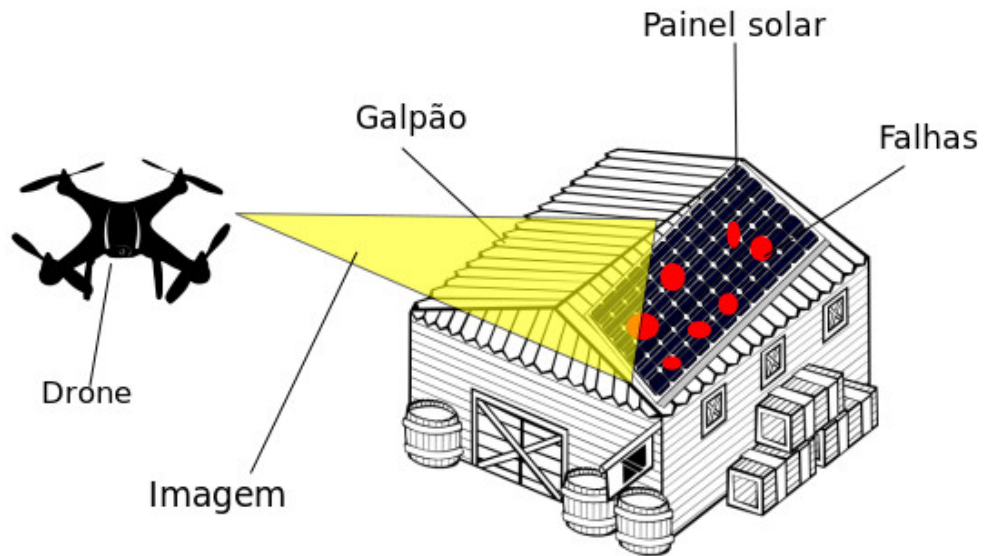


Figura 5.9 – Funcionamento da detecção de falhas em uma situação prática.

de coordenadas no mundo pela rotação conhecida dos sensores do drone. A única ambiguidade restante é a distância até a câmera. Uma imagem apenas não pode diferenciar entre um objeto grande distante e um objeto pequeno próximo. Porém, tal ambiguidade é resolvida quando se supõe o conhecimento das dimensões dos painéis. Quando ajustadas as dimensões reais para as dimensões na imagem, o fator encontrado é a distância da câmera.

5.2.4 Produção de imagens em simulação

A detecção e localização de falhas por redes neurais para esse problema segue a proposta deste trabalho como apresentada no capítulo 3 sobre sistemas de simulação. Assim, usando a propriedade de generalidade que se pode obter com redes neurais com as variações de imagens apresentadas e inspirado pelo sucessos de outros trabalhos (LEVINE *et al.*, 2018; OPENAI *et al.*, 2018), as imagens e as falhas foram geradas automaticamente usando um sistema de simulação construído na plataforma Unity 3D. Um exemplo de imagem gerada é mostrado na

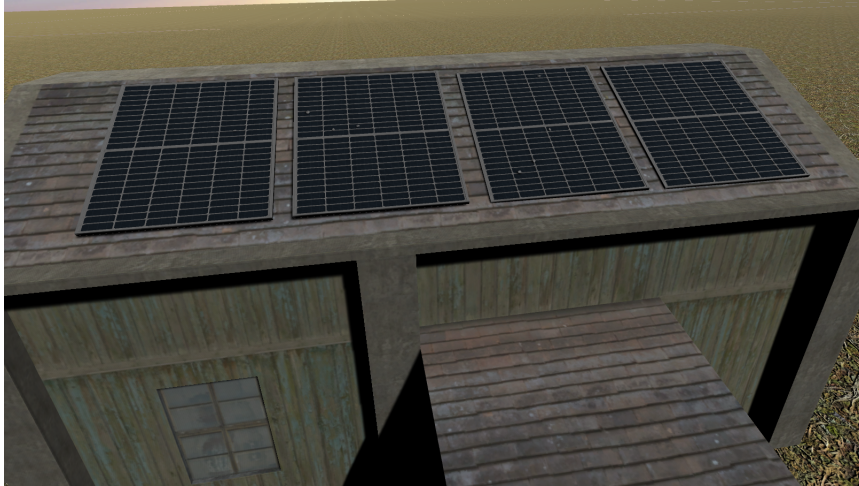


Figura 5.10 – Exemplo de imagem gerada pelo programa de simulação.

figura 5.10.

O sistema de geração de imagens com falhas usa a seguinte sequência:

- 1 Gera-se um modelo de telhado de construção e com quatro painéis fotovoltaicos posicionados.
- 2 Gera-se uma trajetória de pontos e direções para a câmera virtual do drone que fotografa os painéis. Essas posições são ajustadas para representar diversos ângulos mas devem conter a imagem dos painéis totalmente ou na maior parte.
- 3 Geram-se modelos de falhas (pequenas esferas esbranquiçadas).
- 4 Sorteia-se um número de falhas de cada painel.
- 5 Para o número de falhas sorteado, geram-se valores de distorções de escala em dois eixos. Quando as falhas são afetadas desses valores ficam maiores ou menores dentro de uma faixa. Ao ficarem maiores ou menores, emula-se uma severidade maior das falhas.
- 6 As posições das falhas são sorteadas sobre coordenadas horizontal e vertical sobre cada painel com distribuição uniforme.
- 7 A câmera que visualiza o ambiente é posicionada em uma nova posição.
- 8 Uma imagem é capturada e salva em arquivo.
- 9 As coordenadas tridimensionais das falhas e dos cantos de cada painel são obtidas e salvas num arquivo. A posição da câmera, bem como sua inclinação são também registradas.
- 10 As coordenadas das falhas e dos cantos dos painéis são projetadas no plano de visão câmera que representa a captura da imagem e convertidas para pixels. Esses valores são também salvos em arquivo.

11 Repetem-se os passos de 4 a 10 até o fim das posições válidas para a câmera virtual.

Em uma sequência de simulação como a descrita, não há a necessidade de simulações físicas e portanto a escala de tempo pode ser desativada. Apenas se reposicionam falhas e câmera para obter mais imagens.

O código e detalhes completos de simulação estão disponíveis em um projeto hospedado publicamente e são também apresentados de forma documentada no apêndice D.

5.2.5 Ajuste por rede neural

O problema de detecção de objetos em imagens já é bastante explorado em diversos contextos, com e sem o uso de transferência de conhecimento. As duas abordagens são testadas aqui.

Os arquivos gerados com as coordenadas de imagem para painéis e falhas foram convertidos em máscaras. Isso permite o uso nos principais formatos de treinamento de redes neurais. Rigorosamente, a detecção do painel é um tipo de detecção de objeto, enquanto a detecção das falhas é uma tipo de detecção de características de imagens. Como o formato é simples, a detecção por redes neurais deve conferir robustez em relação a uma técnica baseada apenas em processamento de imagem para detectar pequenas manchas claras.

Porém, treina-se a rede para ambos os tipos de detecção usando as mesmas ferramentas e dados de entrada. Neste caso, imagens e suas máscaras indicativas de presença.

O problema pode ser dividido em duas etapas, a detectar painéis e a de detectar falhas. A abordagem de solução, porém, pode acontecer com duas redes neurais separadas ou com uma só. Em uma abordagem com duas redes neurais, tem-se soluções especializadas que deve ser combinadas. É esperado que as redes aprendam seus comportamentos de objetivo de forma mais fácil e com menos amostras. Porém, é também apontado que algumas redes podem se aproveitar de possuírem mais de um objetivo de detecção no mesmo problema pois suas camadas reaproveitam as informações disponíveis de um objetivo para outro, especialmente quando trabalhando sobre os mesmo valores de entrada.

Para avaliar essa diferença, as duas abordagens foram testadas. Como se trata de um problema usual de detecção de objetos, usou-se o padrão apresentado no pacote Tensorflow. Essa arquitetura padrão de rede foi treinada usando como o objetivo detectar o painel, uma outra vez a mesma arquitetura foi treinada com o objetivo de detectar as falhas. E além desses dois testes, a estrutura da rede foi adaptada para identificar tanto contornos de painéis quanto de falhas simultaneamente.

Os treinamentos para tais testes com as imagens sintéticas são apresentados no apêndice D, bem como os detalhes de programação e documentação do código. Foram geradas 1000 imagens, sendo 700 usadas no treinamento da rede e 300 usadas como teste. Notificam-se

resultados de aprendizado das falhas com sucesso de 82% da captura de falhas, perdendo-se as falhas de menores dimensões relativamente à imagem. Para as imagens localizadas a soma dos erros das coordenadas de falha corresponde a um equivalente de médio de 50 pixels.

O argumento apresentado para a convexidade das imperfeições de modelo no capítulo 4 leva ao uso de técnicas de treinamento baseadas em métodos de gradiente, como a usada de forma padrão no pacote Tensorflow e outros critérios de treinamento levaram a resultados piores com tempos de processamento muito lentos para convergência.

5.2.6 Conclusões

A abordagem de localizar falhas usando redes neurais provou-se robusta e com precisão adequada. Evidente que testes experimentais com imagens realistas em grande número ainda devem ser executados antes de aplicações comerciais. Os códigos de simulação e treinamento de redes já estão ajustados para obter maior ou menor número de imagens para o treinamento e também para variações diferentes de falhas e posições de câmera. A combinação com o modelo de localização do drone depende também das especificidades do equipamento e tenciona-se explorar esse caminho usando um modelo comercial.

5.3 EC 3 – Microscópio tridimensional

5.3.1 Descrição do problema

Neste capítulo considera-se o problema de usar um microscópio portátil posicionado em diversas distâncias de um objeto para obter imagens tridimensionais. As aplicações são importantes em paleontologia no estudo de microfósseis tanto para identificação de espécies quanto para estudo de características morfológicas de um fóssil em particular, que pode revelar informações de idade, condição de morte, ambiente e predação, (CARACUEL *et al.*, 2000; CARACUEL; DELGADO, 2002; CRUZADO-CABALLERO *et al.*, 2019; LELIS *et al.*, 2017).

Isso é conseguido porque a distância focal varia e em cada região da imagem apresentará como maior ou menor definição. Os pontos de maior definição obtidos em uma sequência de imagens serão os pontos que estão focalizados. Como cada imagem está em uma distância focal conhecida, pode-se inferir que os pontos mais focalizados em cada imagem correspondem a pontos na distância focal. Assim, atribui-se uma altura para cada ponto e tem-se um modelo de alturas.

A partir de um modelo de alturas dessa natureza pode se calcular o modelo tridimensional usando hipóteses sobre o formato da superfície. Em uma abordagem mais simples pode-se apenas supor continuidade da superfície para a reconstrução.

A reconstrução de imagens com diferentes pontos focais é uma técnica bastante

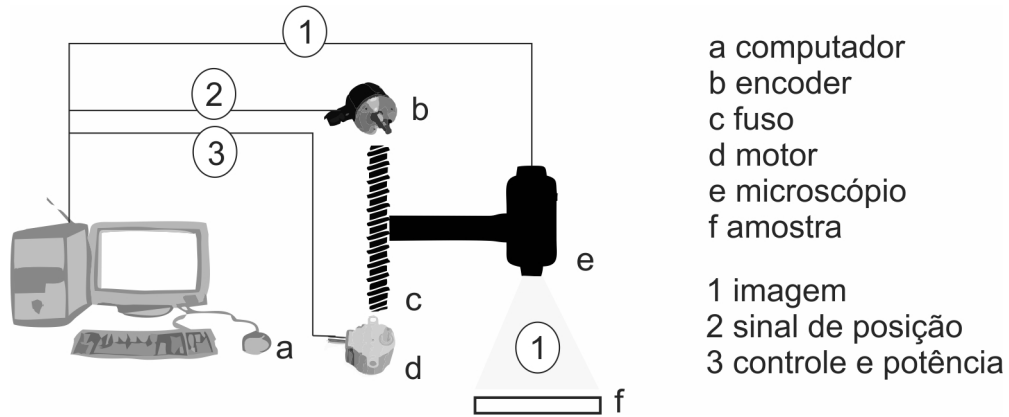


Figura 5.11 – Esquema de funcionamento do microscópio tridimensional.

usada da fotogrametria e particularmente usada em fósseis, devido à possibilidade de se posicionar uma câmera e usar as fotos combinadas.

Uma imagem melhor focalizada significa uma imagem mais nítida e portando com maior discrepância entre os valores de seus pixels. Ao se tomar uma sequência de imagens, deve-se ajustá-las para obter uma nova escala. Os pixels que estiverem na mesma posição em imagens assim corrigidas correspondem à mesma posição.

Para cada imagem obtida, calcula-se o gradiente e o seu módulo. A taxa de variação que o gradiente representa mostra o quanto um pixel varia em relação aos próximos. Dessa forma, em uma sequência de imagens, cada pixel terá vários valores de módulo de gradiente, o que for maior representa a maior discrepância entre os próximos e portanto maior nitidez. Como o ponto de maior nitidez é o melhor focalizado, assume-se que a altura em que foi tomada essa imagem é a altura do ponto representado naquele pixel.

Essa abordagem funciona na medida em que se conhecem as alturas em que as imagens foram coletadas. Para esse fim, é necessário um sistema de posicionamento. Além disso, a precisão com que as alturas são determinadas sobre a imagem depende do quão próximos são os passos entre a tomada de uma imagem e outra. O processo é ilustrado na figura 5.11.

5.3.2 Reconstrução tridimensional

Uma sequência de etapas para a produzir as imagens para reconstrução tridimensional pode ser a seguinte:

- Posicionar a câmera à distância que dê visibilidade de todo o objeto a ser reconstruído.
- Captura de imagem.
- Posicionar a câmera na vertical ligeiramente acima de uma distância tão pequena quanto seja possível medir a diferença.

- Repetir os passos anteriores até que se tenha varrido a altura do objeto pelo curso da distância focal.

Uma vez com as imagens e o registro das alturas em que foram tiradas é possível aplicar os passos a seguir para a reconstrução:

- Redimensionar todas as imagens para que coincidam em área visível. Ou seja, remover o espaço entre as imagens mais distantes e aumentar proporcionalmente a imagem restante para o mesmo tamanho original. A proporção para o redimensionamento é obtida pela distância movida entre uma imagem e outra.
- Calcular o módulo do gradiente por pixel de cada imagem.
- Para toda a sequência de imagens indicar em qual delas cada pixel tem o maior valor de módulo do gradiente.
- Atribuir a altura da foto tirada em que o pixel tem maior gradiente como sendo a altura daquele ponto subtraindo a altura da primeira imagem.
- Produz-se uma imagem combinada com os pixels de maior definição escolhidos entre todas as fotos com as alturas selecionadas como descrito nos passos anteriores.

5.3.3 Erros de reconstrução

A reconstrução de superfícies tridimensionais pelo método descrito está sujeita a algumas fontes de imprecisão, as principais.

- A precisão é limitada pelo posicionamento vertical da câmera utilizada. O intervalo entre as posições de fotos é limitado pelas capacidade de posicionar a câmera de forma precisa.
- Qualquer mecanismo de posicionamento para a câmera pode produzir imprecisões de ângulação que mudariam a posição relativa dos pixels entre uma imagem e outra e prejudicam potencialmente a avaliação do gradiente e a consequente determinação da altura da imagem em que se obteve o máximo gradiente.
- A qualidade de imagem obtida pode variar e produzir diferentes efeitos de especularidade. Reflexões podem criar brilho na imagem que também prejudicam a sensibilidade de variação do gradiente pois podem deixar a imagem menos nítida.

Procura-se atacar os dois primeiros tipos de erros citados aqui com uma estratégia que aproveita a semelhança entre duas imagens contínuas em uma pilha de imagens obtidas, uma vez que se tratam de imagens obtidas com o mesmo objeto. Essas imagens devem variar

de dimensões relativas e uma rotação tridimensional, o que contabiliza os dois tipos de erros citados.

Nesse sentido, tratam-se de erros com comportamento convexo, pois pequenas variações de quais desses parâmetros (ângulos e altura) correspondem a erros pequenos que tendem para um único zero com a proximidade do valor correto.

5.3.4 Ajuste por rede neural

Pela argumentação teórica de que as rede neurais servem como aproximadores convexos, elas são uma arquitetura que podem absorver o comportamento dos erros previstos para o problema descrito.

As imagens na sequência produzida possuem também semelhanças umas com as outras que podem ser capturadas pela estrutura de auto-semelhança das redes neurais.

Assim, propõe-se aplicar uma rede neural para rastrear a correta altura e também inclinação da imagem obtida. Para tanto, basta obter um conjunto de imagens em que sejam produzidas pequenas variações de ângulo e grandes variações de altura e aplicar essas imagens com esses dados de treinamento. Para garantir a generalidade, a rede pode ser treinada com diversas imagens de fosséis. O objetivo é que a rede capture a alterações visuais que se produzem com essas operações geométricas.

A rede neural deve ser treinada tendo entradas mistas, em que se consideram as seguintes configurações:

A Entradas: imagem e entrada de altura de captura. Saídas: ângulos da câmera.

B Entradas: imagem. Saídas: ângulos da câmera e altura medida. a imagem e a altura da imagem lida.

Para cada uma das propostas, foi necessário implementar um tipo de rede neural, para o item A foi necessário implementar uma rede mista onde as imagens passam por uma série de camadas convolucionais, o resultado dessas camadas é achatado para o formato de um vetor que é concatenado com o valor da altura, em seguida esse vetor passa por uma rede densa que reduz as dimensões das camadas até ter apenas 3 neurônios na última camada. A rede do item B a rede foi muito similar porém era uma rede mais simples com entrada única que passa por uma série de camadas convolucionais seguidas por uma rede densa.

Esses dois modelos também foram testados com as mesmas imagens em escala em cinza, essa proposta visa reduzir o tamanho da rede, no entanto foi necessário redefinir a rede densa para o formato a seguir.

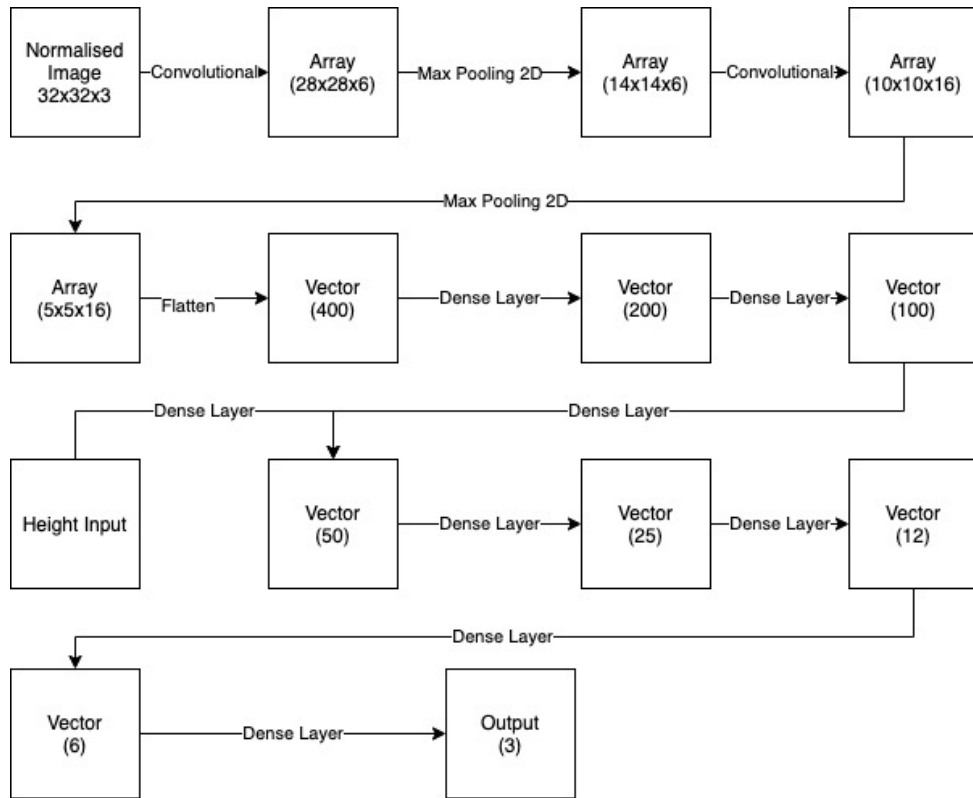


Figura 5.12 – Modelo de rede mista para entrada de imagem e altura.

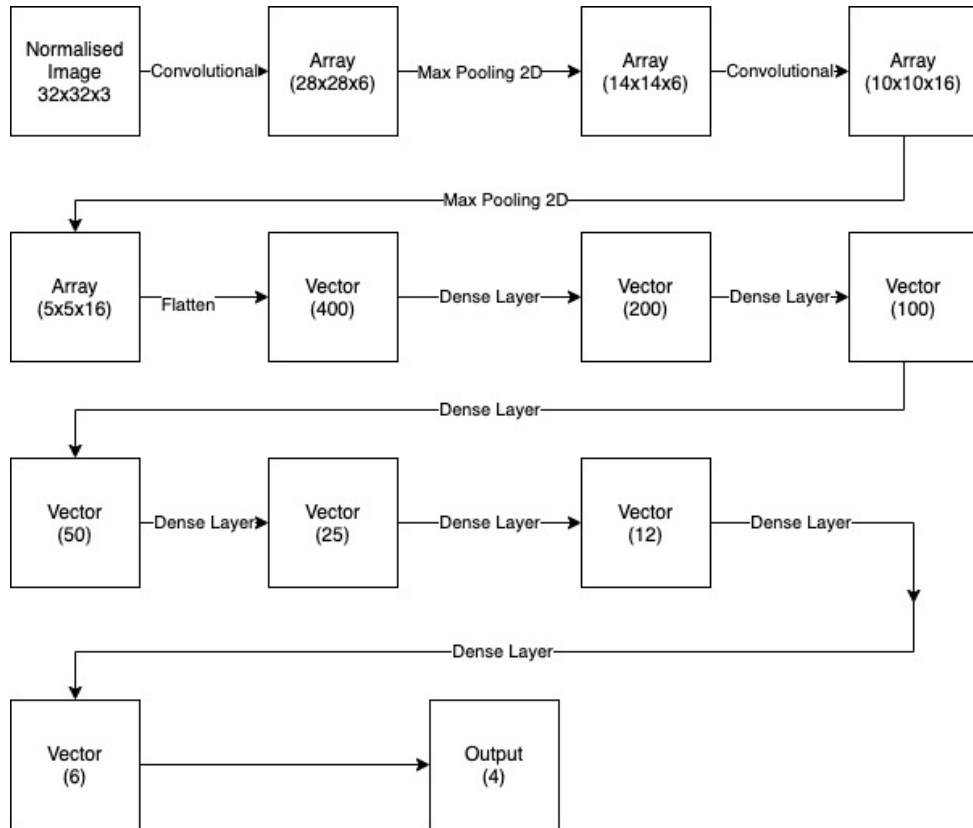


Figura 5.13 – Modelo de rede para entrada de imagem.

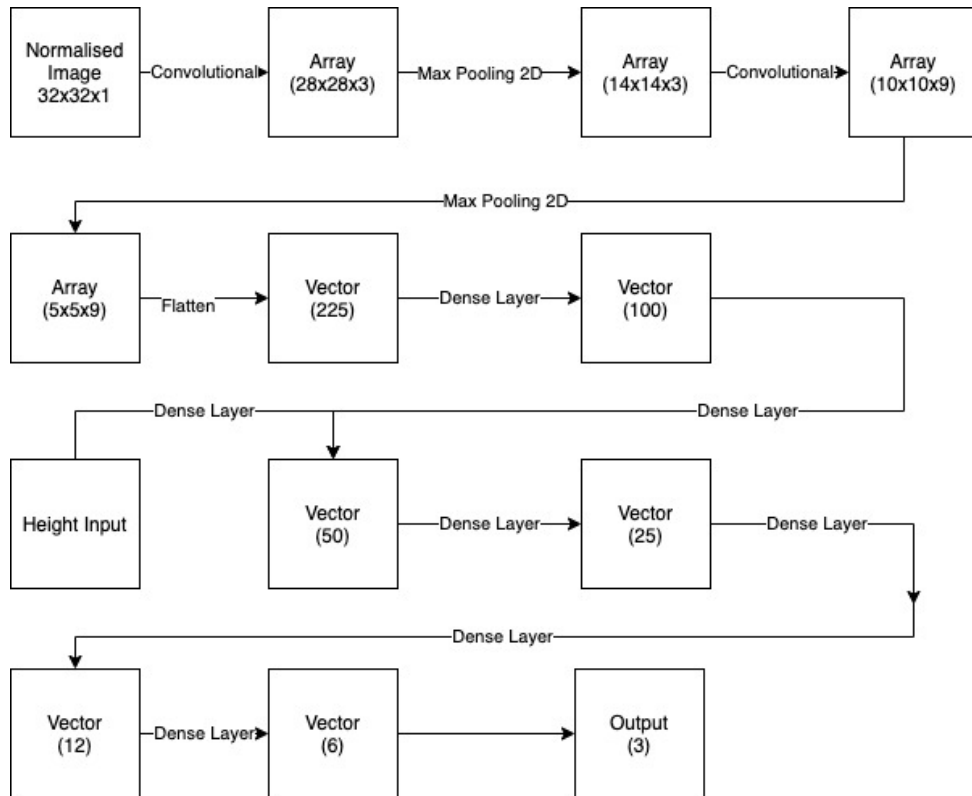


Figura 5.14 – Modelo de rede mista para entrada de imagem em escala em cinza e altura.

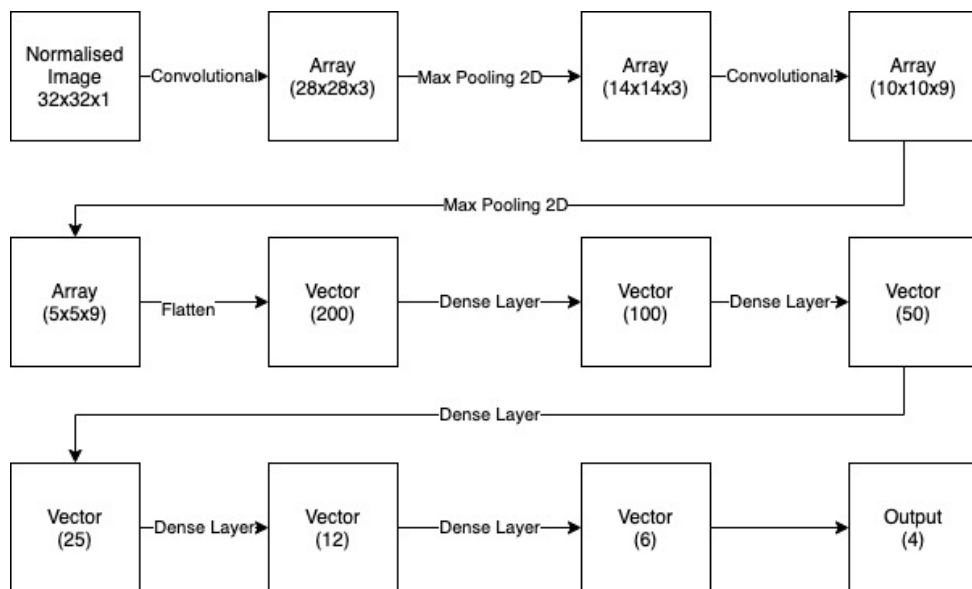


Figura 5.15 – Modelo de rede para entrada de imagem em escala em cinza.

5.3.5 Produção de imagens em simulação

Como já argumentado, ao se tomarem cuidados com a generalidade dos modelos de redes neurais, para diversar aplicações as redes podem ser treinadas com dados produzidos por modelos. Esse treinamento permite a produção de grandes volumes de imagens rotuladas com precisão. As redes neurais assim obtidas representam ferramentas rápidas de aproximação desses modelos além de possuem robustez a variações não modeladas.

Para o caso aqui estudado, podem-se gerar imagens usando modelos tridimensionais de fosséis obtidos em equipamentos precisos. Usando ambientes de simulação, as imagens podem ser geradas considerando as variações de altura que permitem a reconstrução e também as variações de ângulo que representariam os erros de inclinação.

As imagens foram produzidas juntamente com seus dados para o treinamento da rede. Posicionou-se a câmera com uma distância em que o fóssil cobrisse quase toda a área da imagem e a simulação consistia em capturar imagens a cada quadro variando os ângulos tridimensionais da câmera em uma faixa de -9 a +9 graus enquanto a altura era aumentada até que o fóssil ocupasse menos do que um quarto da menor direção da imagem. Para cada iamgem, registram-se os dados de ângulação e altura.

5.3.6 Modelagem de Redes Neurais

A construção de redes neurais possuem alguns recursos com poucas regras de uso, toda rede neural precisa ser adaptada de acordo com o problema e com excessão de poucas regras como a redução pela metade entre uma camada densa e outra não existem muitas alternativas além da tentativa e erro para modelar uma rede. Durante a montagem várias coisas precisam ser consideradas, se existe a necessidade de memória dado que a sequência de dados é um fator importante para o resultado da rede, se a rede vai receber imagens na entrada, que tipo de saída a rede vai utilizar, se ela é uma regressora ou uma classificadora, se o aprendizado vai precisar do uso de "drop out" para controlar o overfitting, todas essas informações e outras são necessárias para a definição de uma rede adequada. No caso em específico deseja-se utilizar imagens e dados numéricos para definir outros valores numéricos, sendo assim necessário camadas convolucionais para processar a imagem conectadas e em seguida conectar com uma camada densa que vai combinar com os outros valores de entrada com a imagem que vai correlacionar as informações das entradas até se encaixar no formado da saída.

5.3.7 Treinamento e teste de rede

Foram usados 11 fósseis para o treinamento da rede, com cada um deles foram tiradas diversas fotos com a camera ligeiramente rotacionada o que simula um desalinhamento da câmera, essa rotação com relação ao eixo normal é registrado para o uso durante o aprendizado e validação da inteligência artificial. As imagens possuíam dimensões de 32x32 e foram separa-

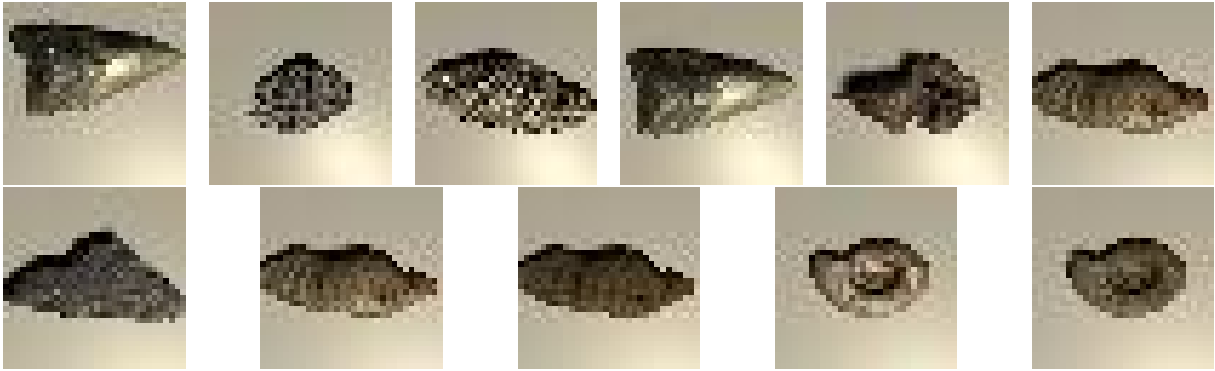


Figura 5.16 – Exemplos de fósseis usados no treinamento.

das em 3 grupos, primeiro foi separado 10 tipos de fósseis que foram divididos aleatoriamente na proporção de 70% e 30%, o primeiro sendo o grupo de treino e o segundo o grupo com 30% era o grupo de teste. Além disso foi separadas todas as imagens de um tipo de fóssil para ser um segundo grupo de testes, com o propósito de testar a rede com um fóssil nunca antes visto de nenhuma maneira durante o treinamento. Para testar a hipótese foram utilizadas redes com algumas configurações, a rede pode usar a imagem e o parâmetro de altura para determinar os ângulos, usar apenas os parâmetros de imagem para determinar a altura e os ângulos, além disso há também a possibilidade da rede utilizar imagens coloridas ou imagens em escala em cinza, totalizando 4 configurações de rede.

- O primeiro teste realizado foi utilizando imagens coloridas utilizando a imagem e a altura para determinar os ângulos da câmera sendo obtido um erro quadrático médio de 0.0012 para o grupo de teste com os mesmos objetos e com um erro de 25.0661 para os testes com fósseis desconhecidos.
- Em seguida foi realizado o teste utilizando apenas a imagem colorida para determinar a altura e os ângulos da câmera e a distância da câmera, sendo obtido um erro de 16.9495 com o grupo de testes de fósseis conhecidos e um erro quadrático médio de 12.0154 para o grupo de fósseis desconhecidos
- O primeiro teste com imagens em escala em cinza foi utilizado o modelo que utiliza a imagem e a distância para determinar os ângulos da câmera e demonstrou um erro quadrático médio de 0.0008 já o erro do grupo de fósseis desconhecidos foi de 25.5840
- O último teste utilizou imagens em escala em cinza e determinava a altura e os ângulos da câmera, esse teste apresentou um erro de 0.0006 no grupo de teste e um erro de 19.6265 no grupo de fósseis desconhecidos

Como demonstrado acima a rede demonstrou uma precisão muito grande em certas circunstâncias mesmo com pouca disponibilidade de dados para o experimento. O conjunto de

dados gerados e testados tinha o tamanho em memória de 14.3 MB. Os detalhes de código e documentação são apresentados nos apêndices B e C.

5.3.8 Conclusões

A precisão indicou que o processo de detectar posicionamento de pequenos desvios usando imagens é prático para medidas precisas de forma e volume. Verificou-se ainda que a rede consegue generalidade de comportamento com quantidade relativamente pequena de exemplos. Foram usados poucos fósseis e o comportamento de rastreamento se aplicou sobre formas diferentes. Nesse sentido, foram obtidas as propriedades esperadas de aproximador convexo para a rede neural e o seu uso como intermediário de um observador baseado em modelo é uma técnica que pode encontrar diversas aplicações.

6 CONCLUSÕES E ESTUDOS FUTUROS

6.1 A combinação de modelos e redes neurais

O argumento prático mais importante proposto neste trabalho é o de que há grande espaço para o uso de redes neurais como técnica de inteligência artificial em uma área onde ainda não é contemplada, o tratamento de dados de instrumentação e construção de observadores ou mesmo melhoria de observadores baseados em modelos.

Foram contemplados três estudos de caso:

- EC 1 Estimativa de ângulos de rotação e velocidades angulares inerciais usando sensores de aparelhos celulares.
- EC 2 Localização de falhas em painéis fotovoltaicos usando imagens obtidas de drones.
- EC 3 Obteção de mapas de superfície para posterior reconstrução tridimensional a partir de fotos em distâncias focais diferentes de uma mesma amostra em um microscópio.

6.2 Critérios empíricos de separação

Em todos os estudos de caso (ECs), procurou-se dividir o trabalho em uma parte determinística e conhecida que deveria ser resolvida com informações dinâmicas que podem ser lidas de sensores e com algumas que não estão acessíveis pelo mesmo caminho.

- EC 1 Conheciam-se estimativas para ângulos de rotação a partir do magnetômetro, supondo a constância do campo magnético e duas estimativas para velocidade angular, uma delas com magnetômetro e outra com o giroscópio.
- EC 2 Conheciam-se a posição tridimensional do drone e a sua inclinação, permitindo inferir a direção de obtenção da imagem a partir da localização de sua câmera.
- EC 3 Conhecia-se a posição vertical da tomada de imagem da câmera, porém com um erro limitado pelo motor de posicionamento e pela possível angulação da lente.

Formula-se algum tipo de observador baseado em modelo para preencher as informações indisponíveis.

- EC 1 Conheciam-se estimativas para ângulos de rotação a partir do magnetômetro, supondo a constância do campo magnético e duas estimativas para velocidade angular, uma delas com magnetômetro e outra com o giroscópio.

- EC 2 Conheciam-se a posição tridimensional do drone e a sua inclinação, permitindo inferir a direção de obtenção da imagem a partir da localização de sua câmera.
- EC 3 Conheciam-se a posição vertical da tomada de imagem da câmera, porém com um erro limitado pelo motor de posicionamento e pela possível angulação da lente.

Nessa forma, o modelo não é completamente determinístico devido à natureza dos variáveis disponíveis para alimentá-lo dinamicamente. Os dados para usar o modelo se dividem entre aqueles medidos diretamente e aqueles que estão disponíveis em informações não estruturadas como imagens. Então, divide-se a tarefa em uma parte que é um cálculo determinístico e outro não determinístico que produz medidas que podem ser usadas em um modelo desse tipo.

- EC 1 A relação intrínseca entre as variáveis dinâmicas por estarem sujeitas ao mesmo corpo rígido e suas discrepâncias serem supostas como diferentes entre os diferentes sensores. Os sensores podem possuir dinâmicas não modeladas que são absorvidas por um modelo de rede neural através do treinamento.
- EC 2 A localização dos painéis em uma imagem pode ser entendida um problema de segmentação quando se fala de detectar sua presença ou como um problema de pontos característicos, quando se consideram seus extremos. A localização de falhas em painéis é um problema de detecção de pontos característicos. São abordagens típicas de redes neurais covolucionais no sentido de que suas formas são definidas de forma imprecisa por elementos de imagens que são treinados pela rede.
- EC 3 A angulação da captura de uma imagem é visível na variação de contornos em um mapa de direções entre uma imagem e outra. A distância também possui uma relação com as dimensões relativas dos contornos. Similarmente, essas relações são treinadas em uma sequência de imagens.

Assim, o observador baseado em modelo serve para contemplar dinâmicas previsíveis e bem conhecidas e uma parte de suas leituras precisa ser estimada usando uma técnica heurística que absorve variações de menor dimensão, porém cujo comportamento não é conhecido quantitativamente, apenas qualitativamente.

6.3 Redes neurais como aproximadores

Em todos os casos de estudo e na argumentação proposta, as redes neurais são entendidas como aproximadores de comportamentos dinâmicos ou geométricos que podem ter razões determinísticas não modeladas, porém que não são considerados relevantes para o problema e que acontecem em ordem de gradeza abaixo da considerada para as medidas relevantes que se deseja obter.

Dessa forma, medidas de posicionamento e inclinação se tornam possíveis em situações antes previamente não usadas, contornando problemas específicos de cada aplicação.

- EC 1 Os sensores dos aparelhos celulares possuem dinâmicas complexas e que variam com parâmetros específicos de cada unidade fabricada, ainda que qualitativamente semelhantes. Tais sensores apresentam histerese, sensibilidade aos campos elétricos e magnéticos e, principalmente, variações devidas ao fato de serem consultados em tempos de amostragem diferentes e conflitantes pelo processador do aparelho.
- EC 2 Os painéis embora muito semelhantes em aparência para o olho humano, são vagamente diferentes e podem ser vistos em iluminação e ambientes diferentes. O ambiente em que se situação podem apresentar forma geométricas semelhantes impossibilitando o uso direto de técnicas conhecidas de processamento de imagens para detectar características de interesse.
- EC 3 Os formatos diferentes das amostras em um microscópio impossibilitam, assim como no caso anterior, o uso de técnicas de processamento de imagens. Embora seja o caso de que a percepção humana facilmente possa verificar variação de distância e inclinação entre duas imagens e inclusive estimar direção e valores aproximados de tais movimentos.

Em grande medida, o processo de treinamento de redes neurais, que funciona como um otimizador convexo, é um caminho indireto e sub-ótimo para se aproximar da solução desse problemas que poderiam ser formulados como problemas de otimização convexa mono-objetivo. Não se usa essa abordagem direta de otimização devido ao número proibitivo de variáveis em qualquer desses problemas mesmo em versões simplificadas.

Argumentos qualitativos também podem ser levantados, no sentido de que redes neurais apresentam o comportamento de autossimilaridade, como é visível em aplicações de segmentação e reconhecimento de imagens de forma quantitativa e lúdica. A autossimilaridade em diversas escalas, ou em uma rede, entre as diversas camadas com números de pesos diferentes, é uma característica de fractalidade de um conjunto de dados. Essa linha de abordagem teórica é uma inspiração para o campo da inteligência artificial que busca estabelecer conjuntos de regras para as arquiteturas mais apropriadas para cada problema. Muito embora essa área ainda esteja oferecendo apenas resultados iniciais, os elementos de convexidade e fractalidade são apontados como importantes nesses estudos qualitativos e teóricos.

6.4 Trabalhos e aplicações possíveis

Resume-se o tema deste trabalho como a combinação de inteligência artificial usando sensores produzindo modelos de posicionamento dinâmico. Usando as técnicas desenvolvidas e prosseguindo em aplicações semelhantes podem-se apontar algumas aplicações possíveis:

- Integração das técnicas em sensores de movimento biomecânicos para melhoria das leituras de velocidade e posição.
- Integração de leituras biomecânicas com imagens de vídeos usando redes neurais treinadas para estimativa de postura usando imagens.
- Integração dos resultados de detecção de falhas em uma plataforma de dados de imagens geradas por drones automaticamente.
- Integração de estimativas de distância focal com técnicas baseadas em luz direcional em uma sequência para obter modelos tridimensionais de boa qualidade e microscópios tridimensionais.

REFERÊNCIAS

- ABADI, M. *et al.* Tensorflow: A system for large-scale machine learning. In: **12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)**. Savannah, GA: USENIX Association, 2016. p. 265–283. ISBN 978-1-931971-33-1. Disponível em: <<https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi>>.
- ABDOLLAHI, F.; TALEBI, H. A.; PATEL, R. V. A stable neural network-based observer with application to flexible-joint manipulators. **IEEE Transactions on Neural Networks**, v. 17, n. 1, p. 118–129, 2006.
- ALSAFASFEH, M. *et al.* Unsupervised fault detection and analysis for large photovoltaic systems using drones and machine vision. **Energies**, v. 11, n. 9, 2018.
- ANDRADE, C. M. *et al.* Internal and imagined external foci of attention do not influence pirouette performance in ballet dancers. **Research Quarterly for Exercise and Sport**, Routledge, v. 00, p. 1–10, 2020. ISSN 21683824. Disponível em: <<https://doi.org/10.1080/02701367.2019.1698697>>.
- ANDRYCHOWICZ, M. *et al.* Learning dexterous in-hand manipulation. **The International Journal of Robotics Research**, v. 39, n. 1, p. 3–20, 2020.
- BERGMEIR, C. N.; SANCHEZ, J. M. B. **Neural Networks in R Using the Stuttgart Neural Network Simulator: RSNNS**. American Statistical Association, 2012. Disponível em: <<http://hdl.handle.net/10481/39548>>.
- BERTHELOT, D. *et al.* Understanding and improving interpolation in autoencoders via an adversarial regularizer. In: **International Conference on Learning Representations**. [s.n.], 2019. Disponível em: <<https://openreview.net/forum?id=S1fQSiCcYm>>.
- BESANCON, G. (Ed.). **Nonlinear Observers and Applications**. [S.l.]: Springer, 2007. ISBN 2007931052.
- BIEBERICH, E. Recurrent fractal neural networks: a strategy for the exchange of local and global information processing in the brain. **BioSystems**, v. 66, p. 145–164, 2002. Disponível em: <www.elsevier.com/locate/biosystems>.
- BILAL, A. *et al.* Do convolutional neural networks learn class hierarchy? **IEEE Transactions on Visualization and Computer Graphics**, IEEE Computer Society, v. 24, p. 152–165, 1 2018. ISSN 10772626.
- BINI, S. A. Artificial intelligence, machine learning, deep learning, and cognitive computing: What do these terms mean and how will they impact health care? **Journal of Arthroplasty**, Churchill Livingstone Inc., v. 33, p. 2358–2361, 8 2018. ISSN 15328406.
- BLANCHARD, N. *et al.* A neurobiological evaluation metric for neural network model search. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2019.
- BOSCH, P. P. J. van den; KLAUW, A. C. van der. **Modeling, identification and simulation of dynamical systems**. [S.l.]: CRC Press, 1994.

BOYLE, M. The integration of angular velocity. **Advances in Applied Clifford Algebras**, Springer Science and Business Media LLC, v. 27, n. 3, p. 2345–2374, May 2017. ISSN 1661-4909. Disponible em: <<http://dx.doi.org/10.1007/s00006-017-0793-z>>.

BULAT, A. *et al.* **Toward fast and accurate human pose estimation via soft-gated skip connections**. 2020.

BULLING, A.; BLANKE, U.; SCHIELE, B. A tutorial on human activity recognition using body-worn inertial sensors. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 46, n. 3, jan. 2014. ISSN 0360-0300. Disponible em: <<https://doi.org/10.1145/2499621>>.

CARACUEL, J. E.; DELGADO, J. C. J. Aplicaciones fotogramétricas digitales en el análisis morfométrico de fósiles. **Boletín Geológico y Minero**, v. 113, p. 85–95, 2002.

CARACUEL, S. E.; CARDENAL, J.; DELGADO-GARCI, J. A simple photogrammetric method to improve quantitative image analysis in geoscience research. **Journal of Sedimentary Research**, v. 70, p. 1337–1340, 2000.

CARNEVALE, E.; LOMBARDI, L.; ZANCHI, L. Life cycle assessment of solar energy systems: Comparison of photovoltaic and water thermal heater at domestic scale. **Energy**, v. 77, p. 434 – 446, 2014. ISSN 0360-5442. Disponible em: <<http://www.sciencedirect.com/science/article/pii/S0360544214010846>>.

CHEN, W. *et al.* Synthesizing training images for boosting human 3d pose estimation. In: **2016 Fourth International Conference on 3D Vision (3DV)**. [S.l.: s.n.], 2016. p. 479–488.

CHUNG, T. J. **Computational Fluid Dynamics**. 2. ed. [S.l.]: Cambridge University Press, 2010.

COMBES, R. T. des *et al.* On the learning dynamics of deep neural networks. p. 1–21, 2018. Disponible em: <<http://arxiv.org/abs/1809.06848>>.

CORKE, P. **Robotics, Vision and Control: Fundamental Algorithms in MATLAB**. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2013. ISBN 3642201431.

CRAIG, J. R. R.; KURDILA, A. J. **Fundamentals of Structural Dynamics**. 2. ed. [S.l.]: John Wiley & Sons, 2006.

CRUZADO-CABALLERO, P.; JIMÉNEZ-GOMIS, C.; RUIZ, C. C. Implementación de la fotogrametría para la salvaguarda del patrimonio paleontológico canario photogrammetry implementation for the protection of the canarian palaeontological heritage. **Geogaceta**, v. 66, p. 59–62, 2019.

CUCCHIELLA, F.; D'ADAMO, I.; GASTALDI, M. Photovoltaic energy systems with battery storage for residential areas: an economic analysis. **Journal of Cleaner Production**, v. 131, p. 460 – 474, 2016. ISSN 0959-6526. Disponible em: <<http://www.sciencedirect.com/science/article/pii/S0959652616304462>>.

CUESTA-VARGAS, A. I.; GALÁN-MERCANT, A.; WILLIAMS, J. M. The use of inertial sensors system for human motion analysis. **Physical Therapy Reviews**, Taylor Francis, v. 15, n. 6, p. 462–473, 2010. PMID: 23565045. Disponible em: <<https://doi.org/10.1179/1743288X11Y.0000000006>>.

- DAPONTE, P. *et al.* Metrology for drone and drone for metrology: Measurement systems on small civilian drones. In: **2015 IEEE Metrology for Aerospace (MetroAeroSpace)**. [S.l.: s.n.], 2015. p. 306–311.
- DARBY, J. *et al.* An evaluation of 3d head pose estimation using the microsoft kinect v2. **Gait Posture**, v. 48, p. 83 – 88, 2016. ISSN 0966-6362. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0966636216300479>>.
- DILLON, J. V. *et al.* Tensorflow distributions. **CoRR**, abs/1711.10604, 2017. Disponível em: <<http://arxiv.org/abs/1711.10604>>.
- DOBREV, D. A definition of artificial intelligence. 10 2012. Disponível em: <<http://arxiv.org/abs/1210.1568>>.
- DOUGLAS, M. K.; BERARDI, V. L. Revisiting squared-error and cross-entropy functions for training neural network classifiers. **Neural Computation and Applications**, Springer-Verlag, n. 3, p. 310–318, 2005.
- Flayols, T. *et al.* Experimental evaluation of simple estimators for humanoid robots. In: **2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)**. [S.l.: s.n.], 2017. p. 889–895.
- FRANKLIN, G. F.; POWELL, J. D.; EMAMI-NAEINI, A. **Feedback control of dynamic systems**. [S.l.]: Pearson, 2014. ISBN 978-0-13-349659-8.
- FRIESE, K.-I.; HERRLICH, M.; WOLTER, F.-E. Using game engines for visualization in scientific applications. In: CIANCARINI, P. *et al.* (Ed.). **New Frontiers for Entertainment Computing**. Boston, MA: Springer US, 2008. p. 11–22. ISBN 978-0-387-09701-5.
- FUJIMOTO, R. *et al.* (Ed.). **Research Challenges in Modeling and Simulation for Engineering Complex Systems**. [S.l.]: Springer, 2017. (Simulation Foundations, Methods and Applications).
- GAO, J. *et al.* Multiscale entropy analysis of biological signals: a fundamental bi-scaling law. **Frontiers in Computational Neuroscience**, v. 9, p. 64, 2015. ISSN 1662-5188. Disponível em: <<https://www.frontiersin.org/article/10.3389/fncom.2015.00064>>.
- GARRIDO, J. M. **Introduction to Dynamical Systems**. [S.l.: s.n.], 2018. 275-305 p. ISBN 0387946772.
- GECGEL, O. *et al.* Gearbox fault diagnostics using deep learning with simulated data. In: **2019 IEEE International Conference on Prognostics and Health Management (ICPHM)**. [S.l.: s.n.], 2019. p. 1–8.
- GOLDSBOROUGH, P. A tour of tensorflow. **CoRR**, abs/1610.01178, 2016. Disponível em: <<http://arxiv.org/abs/1610.01178>>.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. <<http://www.deeplearningbook.org>>.
- GRABSKA-GRADZIŃSKA, I.; NOWAK, L.; GRABSKA, E. Automatic story generation based on graph model using godot engine. In: RUTKOWSKI, L. *et al.* (Ed.). **Artificial Intelligence and Soft Computing**. Cham: Springer International Publishing, 2020. p. 397–405. ISBN 978-3-030-61401-0.

GRAVES, A.; MOHAMED, A. rahman; HINTON, G. E. Speech recognition with deep recurrent neural networks. **CoRR**, abs/1303.5778, 2013. Disponível em: <<http://dblp.uni-trier.de/db/journals/corr/corr1303.html#abs-1303-5778>>.

GREGORY, J. **Game Engine Architecture**. 3. ed. [S.l.]: CRC Press, 2018.

GUANA, V.; STROULIA, E.; NGUYEN, V. Building a game engine: A tale of modern model-driven engineering. In: **2015 IEEE/ACM 4th International Workshop on Games and Software Engineering**. [S.l.: s.n.], 2015. p. 15–21.

HAEBERLE, H. S. *et al.* Artificial intelligence and machine learning in lower extremity arthroplasty: A review. **Journal of Arthroplasty**, Churchill Livingstone Inc., v. 34, p. 2201–2203, 10 2019. ISSN 15328406.

HAGHIGHI, P. S.; SETON, O.; NASRAOUI, O. **An Explainable Autoencoder For Collaborative Filtering Recommendation**. 2019.

HALLADAY, K. Writing shaders in godot. In: _____. **Practical Shader Development: Vertex and Fragment Shaders for Game Developers**. Berkeley, CA: Apress, 2019. p. 351–368. ISBN 978-1-4842-4457-9. Disponível em: <https://doi.org/10.1007/978-1-4842-4457-9_18>.

HEDRICK, T. L. Software techniques for two- and three-dimensional kinematic measurements of biological and biomimetic systems. **Bioinspiration & Biomimetics**, IOP Publishing, v. 3, n. 3, p. 034001, jul 2008. Disponível em: <<https://doi.org/10.1088/1748-3182/3/3/034001>>.

HELM, J. M. *et al.* Machine learning and artificial intelligence: Definitions, applications, and future directions. **Current Reviews in Musculoskeletal Medicine**, Springer, v. 13, p. 69–76, 2 2020. ISSN 19359748. Disponível em: <<https://link.springer.com/article/10.1007/s12178-020-09600-8>>.

HENRY, C. *et al.* Automatic detection system of deteriorated pv modules using drone with thermal camera. **Applied Sciences**, v. 20, n. 11, 2020.

HOFFER, E.; AILON, N. Deep metric learning using triplet network. In: FERAGEN, A.; PELILLO, M.; LOOG, M. (Ed.). **Similarity-Based Pattern Recognition**. Cham: Springer International Publishing, 2015. p. 84–92. ISBN 978-3-319-24261-3.

IMAIZUMI, M.; FUKUMIZU, K. **Deep Neural Networks Learn Non-Smooth Functions Effectively**. 2019.

JANA, A. K. **Chemical process modelling and computer simulation**. 3. ed. [S.l.]: PHI Learning, 2018.

JIANG, H. A new perspective on machine learning : How to do perfect supervised learning. 2000.

JIANG, H. Why learning of large-scale neural networks behaves like convex optimization. 2019. Disponível em: <<http://arxiv.org/abs/1903.02140>>.

JULIANI, A. *et al.* Unity: A general platform for intelligent agents. **CoRR**, abs/1809.02627, 2018. Disponível em: <<http://arxiv.org/abs/1809.02627>>.

KETKAR, N. Introduction to PyTorch. In: _____. **Deep Learning with Python: A Hands-on Introduction**. Berkeley, CA: Apress, 2017. p. 195–208. ISBN 978-1-4842-2766-4. Disponível em: <https://doi.org/10.1007/978-1-4842-2766-4_12>.

KIM, Y. H.; LEWIS, F. L.; ABDALLAH, C. T. A dynamic recurrent neural-network-based adaptive observer for a class of nonlinear systems. **Automatica**, v. 33, n. 8, p. 1539 – 1543, 1997. ISSN 0005-1098. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0005109897000654>>.

KLUEVER, C. A. **Dynamic Systems Modeling, Simulation, and Control**. [S.l.]: John Wiley & Sons, 2014.

KOCIJAN, J. *et al.* Dynamic systems identification with gaussian processes. **Mathematical and Computer Modelling of Dynamical Systems**, Taylor Francis, v. 11, n. 4, p. 411–424, 2005.

KOLAH, A. *et al.* Design of a marker-based human motion tracking system. **Biomedical Signal Processing and Control**, v. 2, n. 1, p. 59 – 67, 2007. ISSN 1746-8094. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1746809407000195>>.

KUCERA, E.; HAFFNER, O.; LESKOVSKÝ, R. Interactive and virtual/mixed reality applications for mechatronics education developed in unity engine. In: **2018 Cybernetics Informatics (K I)**. [S.l.: s.n.], 2018. p. 1–5.

LANDI, G.; ZAMPINI, A. **Linear Algebra and Analytic Geometry for Physical Sciences**. [S.l.]: Springer, 2018. ISBN 978-3-319-78360-4.

LANE, N. D. *et al.* A survey of mobile phone sensing. **IEEE Communications Magazine**, v. 48, n. 9, p. 140–150, 2010.

LEE, M. *et al.* Validity of shoe-type inertial measurement units for parkinson’s disease patients during treadmill walking. **Journal of NeuroEngineering and Rehabilitation**, v. 15, 05 2018.

LELIS, G. *et al.* Combining multifocal microscopy and photometric stereo with a low cost motorized positioner. In: . [S.l.: s.n.], 2017.

LEVINE, S. *et al.* Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. **International Journal of Robotics Research**, v. 37, p. 421–436, 2018. ISSN 17413176.

LEWIS, M.; JACOBSON, J. Game engine in scientific research. **Communications of the ACM**, v. 45, n. 1, 2002. Disponível em: <<https://www.cse.unr.edu/~sushil/class/gas/papers/GameAlp27-lewis.pdf>>.

LI, Y. *et al.* Convergent learning: Do different neural networks learn the same representations? In: . [S.l.: s.n.], 2015. v. 44, p. 196–212.

LI, Z. Robust control of pm spherical stepper motor based on neural networks. In: . [S.l.: s.n.], 2009. v. 56, p. 2945–2954. ISSN 02780046.

LINTRAMI, T. **Unity 2017 Game Development Essentials**. [S.l.]: Packt Publishing, 2018.

LIU, S. *et al.* A novel distance metric: Generalized relative entropy. **Entropy**, Multidisciplinary Digital Publishing Institute, v. 19, p. 269, 6 2017. ISSN 1099-4300. Disponível em: <<https://www.mdpi.com/1099-4300/19/6/269>>.

LJUNG, L. System identification. In: _____. **Wiley Encyclopedia of Electrical and Electronics Engineering**. American Cancer Society, 2017. p. 1–19. ISBN 9780471346081. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/047134608X.W1046.pub2>>.

LJUNG, L.; CHEN, T.; MU, B. A shift in paradigm for system identification. **International Journal of Control**, Taylor Francis, v. 93, n. 2, p. 173–180, 2020.

LUENBERGER, D. An introduction to observers. **IEEE Transactions on Automatic Control**, v. 16, n. 6, p. 596–602, 1971.

MA, L. *et al.* Modeling and calibration of high-order joint-dependent kinematic errors for industrial robots. **Robotics and Computer-Integrated Manufacturing**, v. 50, p. 153–167, 2018. ISSN 0736-5845. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0736584517301965>>.

MARCUS, G. Deep learning: A critical appraisal. 2018.

MCGINLEY, J. L. *et al.* The reliability of three-dimensional kinematic gait measurements: A systematic review. **Gait Posture**, v. 29, n. 3, p. 360–369, 2009. ISSN 0966-6362. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0966636208002646>>.

MESSAY, T.; ORDÓÑEZ, R.; MARCIL, E. Computationally efficient and robust kinematic calibration methodologies and their application to industrial robots. **Robotics and Computer-Integrated Manufacturing**, v. 37, p. 33–48, 2016. ISSN 0736-5845. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S073658451500071X>>.

MICHALSKI JAIME G. CARBONELL, T. M. M. R. S. (Ed.). **Machine learning: An artificial intelligence approach**. [S.l.]: Springer-Verlag, 1984.

NADKARNI, P. M.; OHNO-MACHADO, L.; CHAPMAN, W. W. Natural language processing: an introduction. **Journal of the American Medical Informatics Association**, v. 18, n. 5, p. 544–551, 09 2011. ISSN 1067-5027. Disponível em: <<https://doi.org/10.1136/amiajnl-2011-000464>>.

NAGABANDI, A. *et al.* Deep dynamics models for learning dexterous manipulation. In: KAELBLING, L. P.; KRAGIC, D.; SUGIURA, K. (Ed.). **Proceedings of the Conference on Robot Learning**. [S.l.: s.n.], 2020. (Proceedings of Machine Learning Research, v. 100), p. 1101–1112.

NAYLOR, C. D. On the prospects for a (deep) learning health care system. **JAMA - Journal of the American Medical Association**, American Medical Association, v. 320, p. 1099–1100, 9 2018. ISSN 15383598.

NIKLAUS, S.; MAI, L.; LIU, F. Video frame interpolation via adaptive convolution. 3 2017. Disponível em: <<http://arxiv.org/abs/1703.07514>>.

OPENAI *et al.* Learning dexterous in-hand manipulation. 8 2018. Disponível em: <<http://arxiv.org/abs/1808.00177>>.

PANCH, T.; SZOLOVITS, P.; ATUN, R. Artificial intelligence, machine learning and health systems. **Journal of Global Health**, University of Edinburgh, v. 8, 2018. ISSN 20472986. Disponível em: <[pmc/articles/PMC6199467/https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6199467/](https://pubmed.ncbi.nlm.nih.gov/PMC6199467/)>.

PARK, D.; RAMANAN, D. Articulated pose estimation with tiny synthetic videos. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops**. [S.l.: s.n.], 2015.

PASZKE, A. *et al.* Pytorch: An imperative style, high-performance deep learning library. In: WALLACH, H. *et al.* (Ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2019. v. 32, p. 8026–8037. Disponível em: <<https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf>>.

PINTO, V. A. *et al.* Attention focus does not influence performance of sit-to-stand in young and older adults. **Journal of Motor Behavior**, Routledge, v. 0, p. 1–9, 2020. ISSN 19401027. Disponível em: <<https://doi.org/10.1080/00222895.2020.1723477>>.

POPOVIC, M. B. **Biomechanics and Robotics**. [S.l.]: CRC Press, 2013. ISBN 9781482242737.

PRIYADARSHNI; SOHAL, J. Improvement of artificial neural network based character recognition system, using scilab. **Optik**, v. 127, n. 22, p. 10510 – 10518, 2016. ISSN 0030-4026. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0030402616305575>>.

QIU, W.; YUILLE, A. Unrealcv: Connecting computer vision to unreal engine. In: HUA, G.; JÉGOU, H. (Ed.). **Computer Vision – ECCV 2016 Workshops**. Cham: Springer International Publishing, 2016. p. 909–916. ISBN 978-3-319-49409-8.

RAUGEI, M.; FRANKL, P. Life cycle impacts and costs of photovoltaic systems: Current state of the art and future outlooks. **Energy**, v. 34, n. 3, p. 392 – 399, 2009. ISSN 0360-5442. WESC 2006 Advances in Energy Studies. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0360544209000061>>.

ROSA, G. H. D.

Otimização Meta-Heurística para Regularização de Modelos de Aprendizado em Profundidade, 2018.

SEJNOWSKI, T. J.; POGGIO, T. a. **Dynamical Systems in Neuroscience**. [s.n.], 2007. v. 25. 227-256 p. ISBN 978-0262514200. Disponível em: <<http://books.google.com/books?hl=en&lr=&id=kVjM6DFk-twC&oi=fnd&pg=PR15&dq=Dynamical+Systems+in+Neuroscience&ots=KRExnXb9si&sig=eN72JzIWk6-LfvNDSFETcxn0vyo>>.

SHI, F. *et al.* **Circus ANYmal: A Quadruped Learning Dexterous Manipulation with Its Limbs**. 2020.

SICILIANO, B.; KHATIB, O. **Springer Handbook of Robotics**. Berlin, Heidelberg: Springer-Verlag, 2007. ISBN 354023957X.

SORBELLI, F. B. *et al.* On the accuracy of localizing terrestrial objects using drones. In: **2018 IEEE International Conference on Communications (ICC)**. [S.l.: s.n.], 2018. p. 1–7.

- STANKEVICH, E.; PARAMONOV, I.; TIMOFEEV, I. Mobile phone sensors in health applications. In: **2012 12th Conference of Open Innovations Association (FRUCT)**. [S.l.: s.n.], 2012. p. 1–6.
- SULTANA, F.; SUFIAN, A.; DUTTA, P. Evolution of image segmentation using deep convolutional neural network: A survey. 1 2020. Disponível em: <<http://arxiv.org/abs/2001.04074><http://dx.doi.org/10.1016/j.knosys.2020.106062>>.
- SZELISKI, R. **Computer Vision: Algorithms and Applications**. 1st. ed. Berlin, Heidelberg: Springer-Verlag, 2010. ISBN 1848829345.
- TAN, C. *et al.* A survey on deep transfer learning. In: KŮRKOVÁ, V. *et al.* (Ed.). **Artificial Neural Networks and Machine Learning – ICANN 2018**. Cham: Springer International Publishing, 2018. p. 270–279. ISBN 978-3-030-01424-7.
- THEOCHARIS, J.; PETRIDIS, V. Neural network observer for induction motor control. **IEEE Control Systems Magazine**, v. 14, n. 2, p. 26–37, 1994.
- THORN, A. **Mastering Unity Scripting**. [S.l.]: Packt Publishing, 2015.
- THRUN, S.; BURGARD, W.; FOX, D. **Probabilistic robotics**. Cambridge, Mass.: MIT Press, 2005. ISBN 0262201623 9780262201629.
- TOPOL, E. J. High-performance medicine: the convergence of human and artificial intelligence. **Nature Medicine**, Nature Publishing Group, v. 25, p. 44–56, 1 2019. ISSN 1546170X.
- TRINH, H.; FERNANDO, T. **Functional observers for dynamical systems**. [S.l.]: Springer, 2012. (Lecture Notes on Control and Information Systems).
- ULLMAN, T. D. *et al.* Mind games: Game engines as an architecture for intuitive physics. **Trends in Cognitive Sciences**, v. 21, n. 9, p. 649 – 665, 2017. ISSN 1364-6613. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1364661317301134>>.
- VEDALDI, A.; LENC, K. Matconvnet: Convolutional neural networks for matlab. In: **Proceedings of the 23rd ACM International Conference on Multimedia**. New York, NY, USA: Association for Computing Machinery, 2015. (MM '15), p. 689–692. ISBN 9781450334594. Disponível em: <<https://doi.org/10.1145/2733373.2807412>>.
- VISVIKIS, D. *et al.* Artificial intelligence, machine (deep) learning and radio(geno)mics: definitions and nuclear medicine imaging applications. 2019. Disponível em: <<https://link.springer.com/article/10.1007/s00259-019-04373-w>>.
- WANG, J. *et al.* Cnn-rnn: A unified framework for multi-label image classification. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2016.
- WANG, S.; QIAN, Y.; YU, K. Focal kl-divergence based dilated convolutional neural networks for co-channel speaker identification. In: **2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.: s.n.], 2018. p. 5339–5343.
- Wu, L. *et al.* A minimal poe-based model for robotic kinematic calibration with only position measurements. **IEEE Transactions on Automation Science and Engineering**, v. 12, n. 2, p. 758–763, 2015.

XIE, M. *et al.* A robust data interpolation based on a back propagation artificial neural network operator for incomplete acquisition in wireless sensor networks. **Journal of Sensors**, Hindawi Limited, v. 2018, 2018. ISSN 16877268.

Yi, J. *et al.* Kinematic modeling and analysis of skid-steered mobile robots with applications to low-cost inertial-measurement-unit-based motion estimation. **IEEE Transactions on Robotics**, v. 25, n. 5, p. 1087–1097, 2009.

YONG, J. **Optimization Theory: A Concise Introduction**. [S.l.: s.n.], 2018. ISBN 978-981-323-764-3.

YOSINSKI, J. *et al.* Understanding neural networks through deep visualization. 6 2015. Disponível em: <<http://arxiv.org/abs/1506.06579>>.

ZADEH, L. On the identification problem. **IRE Transactions on Circuit Theory**, v. 3, n. 4, p. 277–281, 1956.

ZAHEDI, A. Solar photovoltaic (pv) energy; latest developments in the building integrated and hybrid pv systems. **Renewable Energy**, v. 31, n. 5, p. 711 – 718, 2006. ISSN 0960-1481. SOUTH/SOUTH. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0960148105001990>>.

ZHANG, Z.; SABUNCU, M. Generalized cross entropy loss for training deep neural networks with noisy labels. In: BENGIO, S. *et al.* (Ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2018. v. 31, p. 8778–8788. Disponível em: <<https://proceedings.neurips.cc/paper/2018/file/f2925f97bc13ad2852a7a551802feca0-Paper.pdf>>.

ANEXO A – CÓDIGO E DOCUMENTAÇÃO PARA DADOS DE SENSORES INERCIAIS

O trabalho descrito conceitualmente na seção 5.1 pode ser analisado com detalhes neste anexo. São apresentados os programas para processamento de sinais, treinamento e aplicação das redes neurais, cálculos de modelo cinemático e cálculos estatísticos usados no trabalho publicado. Apresentam-se também programas de validação das leituras obtidas com celular quando comparadas com um sistema de aquisição de movimento.

```

1  % validação de dados medidos para o eixo do celular em relação ao ...
    eixo do
2  % tronco no experimento de pirueta no bale
3  clearvars, clc
4
5  % inícios e fins dos movimentos de pirueta (3 repetições)
6  onset = [4.507 5.590 5.016];
7  termination = [9.541 10.309 8.085];
8  trials = [1 5 7];
9
10 for k = 1:3;
11     trial = trials(k);
12     clear('C','t','av','avr','at','atr')
13     st = num2str(trial,'%04d');
14     file = ['TMR0988_' st '.tsv'];
15     C = dlmread(file,'\t',11,0);
16
17
18     sf = 120;
19     dt = 1/sf;
20
21     N = size(C,1);
22     t = (0:(N-1))*dt;
23
24     plot(t,C(:,4:6))
25
26     f = 0:0.1:40;
27     Cpsd = pwelch(C,hanning(round(N/10)),round(N/20),f,sf);
28     Cpsd = max(Cpsd');
29     plot(f,20*log10(Cpsd))
30     title('PSD estimated with Welch Method')
31     xlabel('Frequency [Hz]')
32     ylabel('Power density [dB]')

```



```

33
34 % marcadores para o teste de pirueta
35 % 1 - LIPS - isquio esquerdo
36 % 2 - LV2 - lombar 2
37 % 3 - LAC - ombro esquerdo
38 % 4 - IJ - pescoço frente
39 % 5 - RIPS - isquio direito
40 % 6 - T1 - cervical 2
41 % 7 - nada
42 % 8 - LIAS - cintura esquerda
43 % 9 - LV1 - lombar 1
44 % 10 - RAC - ombro direito
45 % 11 - LV3 - lombar 3
46 % 12 - PX - peito/esterno
47 % 13 - C7 - cervical 1
48 % 14 - RIAS - cintura direita
49 % 15 - T8 = MAI - meio costas
50
51 LIPS = C(:,1:3);
52 LV2 = C(:,4:6);
53 LAC = C(:,7:9);
54 IJ = C(:,10:12);
55 RIPS = C(:,13:15);
56 T1 = C(:,16:18);
57 % 7 - nada
58 LIAS = C(:,22:24);
59 LV1 = C(:,25:27);
60 RAC = C(:,28:30);
61 LV3 = C(:,31:33);
62 PX = C(:,34:36);
63 C7 = C(:,37:39);
64 RIAS = C(:,40:42);
65 T8 = C(:,43:45);
66
67 % thorax axis - Y axis according to the ISB recommendation
68 IP = (PX+T8)*0.5;
69 SP = (IJ+C7)*0.5;
70 YTq = SP-IP;
71
72 % smartphone axis
73 IP = (PX+T8)*0.5;
74 SP = (IJ+C7+.5*LAC+.5*RAC)/3;
75 YTs = SP-IP;
76
77
78 de = YTq;
79 aux = vrrotvec(de(1,:), [0 0 1]);

```

```

80 at(1) = aux(end);
81 for i = 2:N
82     aux = vrrotvec(de(i,:), [0 0 1]);
83     at(i) = aux(end);
84     dde(i,:) = de(i, :)-de(i-1, :);
85 end
86
87 [b,a]=butter(10,6/(120/2));
88 dde = filtfilt(b,a,dde);
89
90 for i = 2:N
91     aux = vrrotvec(dde(i,:), [0 0 1]);
92     av(i) = aux(end);
93 end
94 av = filtfilt(b,a,av);
95
96 der = YTs;
97 aux = vrrotvec(der(1,:), [0 0 1]);
98 atr(1) = aux(end);
99 for i = 2:N
100     aux = vrrotvec(der(i,:), [0 0 1]);
101     atr(i) = aux(end);
102     dder(i,:) = der(i, :)-der(i-1, :);
103 end
104 dder = filtfilt(b,a,dder);
105
106 for i = 2:N
107     aux = vrrotvec(dder(i,:), [0 0 1]);
108     avr(i) = aux(end);
109 end
110 avr = filtfilt(b,a,avr);
111
112 [~,ti] = min(abs(onset(k)-t));
113 [~,tf] = min(abs(termination(k)-t));
114
115 t = t(ti:tf);
116 at = at(ti:tf);
117 atr = atr(ti:tf);
118 av = av(ti:tf);
119 avr = avr(ti:tf);
120
121 figure(1)
122 plot(t, at*180/pi, t, atr*180/pi)
123 xlabel('Time [s]')
124 ylabel('Angle of trunk [degrees]')
125 legend('Qualisys', 'Smartphone')
126 title('Comparison of smartphone sensors and motion tracking - ...

```

```

        filtered')
127 saveas(gcf,['angleTrunk' num2str(trial) '.png'])
128 rms((at-atr)./at)
129
130 figure(2)
131 plot(t,av*180/pi,t,avr*180/pi)
132 xlabel('Time [s]')
133 ylabel('Angle of rotation [degrees]')
134 legend('Qualisys','Smartphone')
135 title('Comparison of smartphone sensors and motion tracking - ...
        filtered')
136 saveas(gcf,['angleRotation' num2str(trial) '.png'])
137 rms((av-avr)./av)
138
139 At{k} = at;
140 Atr{k} = atr;
141 Av{k} = av;
142 Avr{k} = avr;
143
144 end
145
146 M = 200;
147 for k = 1:3
148     N = length(At{k});
149     x = 1:N;
150     xq = linspace(x(1),x(end),M);
151     Mat(k,:) = interp1(x,At{k},xq);
152 end
153 Mat = mean(Mat);
154
155 for k = 1:3
156     N = length(Atr{k});
157     x = 1:N;
158     xq = linspace(x(1),x(end),M);
159     Matr(k,:) = interp1(x,Atr{k},xq);
160 end
161 Matr = mean(Matr);
162
163 for k = 1:3
164     N = length(Av{k});
165     x = 1:N;
166     xq = linspace(x(1),x(end),M);
167     Mav(k,:) = interp1(x,Av{k},xq);
168 end
169 Mav = mean(Mav);
170
171 for k = 1:3

```

```

172     N = length(Avr{k});
173     x = 1:N;
174     xq = linspace(x(1),x(end),M);
175     Mavr(k,:) = interp1(x,Avr{k},xq);
176 end
177 Mavr = mean(Mavr);
178
179
180 t = linspace(t(1),t(end),M);
181 figure(1)
182 plot(t,Mat*180/pi,t,Matr*180/pi)
183 xlabel('Time [s]')
184 ylabel('Angle of trunk [degress]')
185 legend('Qualisys','Smartphone')
186 title('Comparison of smartphone sensors and motion tracking - filtered')
187 saveas(gcf,['angleTrunkAverage.png'])
188 rms((at-atr)./at)
189 histogram((at-atr)./at,20,'Normalization','Probability')
190 title('Relative difference of trunk angle between Qualysis and ...
        Smartphone')
191 xlabel('Difference')
192 ylabel('Frequency')
193 saveas(gcf,['angleTrunkDifferenceHistogram.png'])
194
195 figure(3)
196 plot(t,Mav*180/pi,t,Mavr*180/pi)
197 xlabel('Time [s]')
198 ylabel('Angle of rotation [degrees]')
199 legend('Qualisys','Smartphone')
200 title('Comparison of smartphone sensors and motion tracking - filtered')
201 saveas(gcf,['angleRotationAverage.png'])
202 rms((av-avr)./av)
203 histogram((at-atr)./at,20,'Normalization','Probability')
204 title('Relative difference of rotation angle between Qualysis and ...
        Smartphone')
205 xlabel('Difference')
206 ylabel('Frequency')
207 saveas(gcf,['angleRotationDifferenceHistogram.png'])
208
209 % Bland-Altman
210 figure(4)
211 m = (log2(Mat)+log2(Matr))*0.5;
212 d = log2(Mat)-log2(Matr);
213 plot(t,m,t,m+d,t,m-d)
214
215 % relative difference in average
216 disp('Relative difference in average')

```

```

217 disp('Trunk angle')
218 180/pi*(mean(Mat)-mean(Matr))
219 (mean(Mat)-mean(Matr))/mean(Mat)
220 disp('Rotation angle')
221 180/pi*(mean(Mav)-mean(Mavr))
222 (mean(Mav)-mean(Mavr))./mean(Mav)
223
224 % save comparation files
225 T = table;
226 T.Time = t';
227 T.QualalysisAngleTR = 180/pi*Mat';
228 T.SmartphoneAngleTR = 180/pi*Matr';
229 T.QualalysisAngleRA = 180/pi*Mav';
230 T.SmartphoneAngleRA = 180/pi*Mavr';
231 writetable(T, 'comparationQualalysisSmartphone.csv')

```

```

1 % script para ler dados dos arquivos salvos pelo celular
2 % no aplicativo Sensor Kinectics Pro
3 % os dados são filtrados e reamostrados para uma frequência uniforme
4 % valores estatísticos são salvos para análise
5
6 clearvars
7 clc
8 % condições dos experimentos com caneca
9 % VN - caneca vazia, velocidade normal
10 % CN - caneca cheia velocidade normal
11 % CR - caneca cheia, velocidade rápida
12 condicoes = {'VN', 'CN', 'CR'};
13 % jovens e idosos
14 grupos = {'OLD', 'YOUNG'};
15 % foco interno e foco externo
16 focos = {'FE', 'FI'};
17 cond = 1;
18 f = 1;
19 problemas = [];
20 tt=[];
21
22
23 % sujeitos usados para cada grupo e cada foco
24 % sujeitos{grupo,foco}
25 sujeitos{1,1} = [500 501 502 503 504 ...
26     505 506 513 516 517 ...
27     519 521 523 525 527 ...
28     529 531 533 535 537 ...
29     539 543 547 549 551 ...
30     555 557 561];

```

```
31
32 sujeitos{1,2} = [507 508 509 510 511 ...
33     512 514 515 518 520 ...
34     522 524 526 528 530 ...
35     532 534 536 538 540 ...
36     542 544 545 548 550 ...
37     554 556 558 560];
38
39 sujeitos{2,1} = [105 119 120 121 122 124 125 ...
40     126 127 128 129 130 ...
41     131 132 133 134 135 136 ...
42     142 143 144 146 147 ...
43     152 153 155 156 157 158 ...
44     159 162 163 166];
45
46 sujeitos{2,2} = [100 101 102 103 106 ...
47     107 108 110 111 112 ...
48     113 114 115 116 117 ...
49     118 137 139 140 141 ...
50     145 149 150 151 154 ...
51     160 161 164 165 167 ...
52     168 169];
53
54 % pasta de localização dos dados
55 pasta = 'Dados juntos/STS';
56 % pasta = 'Dados juntos/WALK';
57
58 media_acc=[];
59 desvio_acc=[];
60 var_acc=[];
61 suave_acc=[];
62
63 if pasta(end) == 'S'
64     caracter = 's';
65 else
66     caracter = 'c';
67 end
68
69 for g = 1:2
70     for f = 1:2
71         pastaAtual = [pasta '/' grupos{g} '/' focos{f}];
72         for cond = 1:3
73             Tmed_ag=[];
74             Tmed_gg=[];
75             Tmed_mg=[];
76             Tstd_a=[];
77             Tstd_g=[];
```

```

78     Tstd_m=[];
79     Tvar_a=[];
80     Tvar_g=[];
81     Tvar_m=[];
82     Tsuave_a=[];
83
84     for suj = 1:length(sujeitos{g,f})
85         sujeito = sujeitos{g,f}(suj);
86
87         try
88             arq_mag = [num2str(sujeito) condicoes{cond}...
89                 caractere '_mag.csv'];
90             arq_acc = [num2str(sujeito) condicoes{cond}...
91                 caractere '_acc.csv'];
92             arq_gyr = [num2str(sujeito) condicoes{cond}...
93                 caractere '_gyr.csv'];
94
95             dm = csvread([pastaAtual '/' arq_mag],1,0);
96             da = csvread([pastaAtual '/' arq_acc],1,0);
97             dg = csvread([pastaAtual '/' arq_gyr],1,0);
98
99             Fs=40; %taxa de amostragem de 40Hz, menor do que
100             % todas as taxas e maior do que o dobro da frequencia
101             % do sinal mais rápido
102
103             %reamostra o sinal para uma taxa de amostragem
104             % constante por interpolação e um filtro ...
105             anti-aliasing
106             [dmr, tmr] = resample(dm(:,2:4), dm(:,1), Fs);
107             [dar, tar] = resample(da(:,2:4), da(:,1), Fs);
108             [dgr, tgr] = resample(dg(:,2:4), dg(:,1), Fs);
109
110             %corta todos os sinais no mesmo tempo,
111             % para todos os dados acabarem juntos
112             lmin = min([length(tar) length(tgr) length(tmr)]);
113             tar=tar(1:lmin);
114             dar=dar(1:lmin,1:end);
115             tgr=tgr(1:lmin);
116             dgr=dgr(1:lmin,1:end);
117             tmr=tmr(1:lmin);
118             dmr=dmr(1:lmin,1:end);
119             t=tmr;
120             tf=t(end);
121
122             %Filtra os sinais com um filtro butterworth de 10Hz
123             dmf = filtragem_nova(dmr,Fs);

```

```

124     daf = filtragem_nova(dar,Fs);
125     dgf = filtragem_nova(dgr,Fs);
126
127
128     clear da dar dg dgr dm dmr lmin tg ta tg itmin
129
130     %média do sinal de cada eixo de cada variável
131     med_ag=mean(daf);
132     med_gg=mean(dgf);
133     med_mg=mean(dmf);
134
135     %desvio padrão do sinal de cada eixo de cada variável
136     std_a=std(daf);
137     std_g=std(dgf);
138     std_m=std(dmf);
139
140     %variancia do sinal de cada eixo de cada variável
141     var_a=var(daf);
142     var_g=var(dgf);
143     var_m=var(dmf);
144
145     %suavidade
146     suave_a = suave(t,daf);
147
148     save([pasta '/Resultados/' 'sinais_' ...
149         num2str(sujeito) '_' grupos{g} '_' focos{f} ...
150         '_' ...
151         condicoes{cond} '.mat'],...
152         'dmf','daf','dgf','t');
153
154     Tmed_ag=[Tmed_ag;g f cond sujeito med_ag];
155     Tmed_gg=[Tmed_gg;g f cond sujeito med_gg];
156     Tmed_mg=[Tmed_mg;g f cond sujeito med_mg];
157     Tstd_a=[Tstd_a;g f cond sujeito std_a];
158     Tstd_g=[Tstd_g;g f cond sujeito std_g];
159     Tstd_m=[Tstd_m;g f cond sujeito std_m];
160     Tvar_a=[Tvar_a;g f cond sujeito var_a];
161     Tvar_g=[Tvar_g;g f cond sujeito var_g];
162     Tvar_m=[Tvar_m;g f cond sujeito var_m];
163     Tsuave_a=[Tsuave_a;g f cond sujeito suave_a];
164
165     media_acc=[media_acc;g f cond sujeito med_ag];
166     desvio_acc=[desvio_acc;g f cond sujeito std_a];
167     var_acc=[var_acc;g f cond sujeito var_a];
168     suave_acc=[suave_acc;g f cond sujeito suave_a];
169

```



```

170         tt=[tt; g f cond sujeito tf];
171
172         catch e
173             disp(e)
174
175             if g==1
176                 grupo_erro = 'OLD__';
177             else
178                 grupo_erro = 'YOUNG';
179             end
180             problemas = [problemas; [num2str(sujeito) '_'...
181                 grupo_erro '_' ...
182                 focos{f} '_' condicoes{cond}]];
183         end
184         save([pasta '/Resultados/' 'variaveis_' ...
185             grupos{g} '_' focos{f} '_' ...
186             condicoes{cond} '.mat'],...
187             'Tmed_ag','Tmed_gg','Tmed_mg',...
188             'Tstd_a','Tstd_g','Tstd_m','Tvar_a',...
189             'Tvar_g','Tvar_m', 'Tsuave_a');
190     end
191 end
192 end
193 end
194 save([pasta '/Resultados/' 'problemas.mat'],'problemas');
195 save([pasta '/Resultados/' 'estatisticas.mat'],'media_acc',...
196     'desvio_acc', 'var_acc', 'suave_acc');

```

```

1 function [Accn, Omn, Thetan, ti] = interterco(Theta,Om,Acc,tr)
2
3 N = size(Theta,1);
4 Ni = round(N/3);
5 ind = Ni+1:2*Ni;
6 ti = tr(ind);
7
8 Accn = Acc(ind,:);
9 Thetan = Theta(ind,:);
10 Omn = Om(ind,:);

```

```

1 function [Theta,Om,tr] = variaveis_dinamicas(m,g,Nw,t)
2 % Cálculo das variáveis dinâmicas usando os sensores
3 % do magnetômetro e do giroscópio.
4 %
5 % [Theta,Om,tr] = variaveis_dinamicas(m,g,Nw,t)

```

```

6 % m – leitura do magnetômetro de 3 eixos a cada instante de t
7 % g – leitura do giroscópio de 3 eixos a cada instante de t
8 % Nw – tamanho da janela de instantes de tempo usados
9 % na estimativa de cada ponto
10 % t – vetor de tempo
11
12 % estimaiva do ângulo de rotação pelo magnetômetro
13 tm = anguloMagnetometro(m);
14 % estimativa da velocidade angular por duas abordagens
15 [Wg1,Wg2] = estimate_equation(tm,m,g,t);
16 % uso dos ângulos e velocidaes estimadas para obter
17 % estimativa robusta de todos considerando a coerência interna
18 % dos dados
19 [Theta,Om,tr] = estimate_pose_encoder(mt,Wg1,Wg2,Nw,t);

```

```

1 function [Theta,Om,tr] = variaveis_dinamicas(m,g,Nw,t)
2 % Cálculo das variáveis dinâmicas usando os sensores
3 % do magnetômetro e do giroscópio.
4 %
5 % [Theta,Om,tr] = variaveis_dinamicas(m,g,Nw,t)
6 % m – leitura do magnetômetro de 3 eixos a cada instante de t
7 % g – leitura do giroscópio de 3 eixos a cada instante de t
8 % Nw – tamanho da janela de instantes de tempo usados
9 % na estimativa de cada ponto
10 % t – vetor de tempo
11
12 % estimaiva do ângulo de rotação pelo magnetômetro
13 tm = anguloMagnetometro(m);
14 % estimativa da velocidade angular por duas abordagens
15 [Wg1,Wg2] = estimate_equation(tm,m,g,t);
16 % uso dos ângulos e velocidaes estimadas para obter
17 % estimativa robusta de todos considerando a coerência interna
18 % dos dados
19 [Theta,Om,tr] = estimate_pose_encoder(mt,Wg1,Wg2,Nw,t);

```

```

1 % cálculo das variáveis dinâmicas dos experimentos de
2 % caminhada e sentado para de pé com caneca
3
4 % pastas com os dados processados
5 folder{1} = 'Dados juntos/STS/Resultados/';
6 folder{2} = 'Dados juntos/WALK/Resultados/';
7 exps = {'STS', 'WALK'};
8 % tamanho da janela de dados usada do cálculo da variáveis
9 Nw = 60;

```

```

10
11 for indfold = 1:length(folder)
12
13     fold = folder{indfold};
14     arqs = dir([fold 'sinais_*.mat']);
15     nomes = {arqs.name};
16     for i = 1:length(nomes)
17         % carregar arquivo com os dados filtrados
18         load([fold nomes{i}])
19         % dmf – dados filtrados e reamostrados do magnetômetro;
20         % dgf – dados filtrados e reamostrados do giroscópio
21         % daf – dados filtrados e reamostrados do acelerômetro
22         % t – valores de tempo dos dados filtrados
23
24         % cálculo das variáveis dinâmicas usando
25         [Theta,Om,tr] = variaveis_dinamicas(dmf,dgf,Nw,t);
26         % corrigir a aceleração local para o sistema de coordenadas
27         % inercial usando os ângulos calculados
28         Acc = corrigir_acelera(daf,Theta);
29         % selecionar o terço intermediário dos dados para eliminar
30         % início e fim de movimento
31         [Acc, Om, Theta, ti] = interterco(Theta,Om,Acc,tr);
32         % salvar as variáveis dinâmicas em arquivo
33         save([fold 'vd_' nomes{i}], 'Acc', 'Theta', 'Om', 'tr');
34     end
35
36 end

```

```

1 function Acc = corrigir_acelera(daf,Theta)
2
3 N = size(Theta,1);
4 Acc = zeros(N,3);
5 for j = 1:N
6     ang = Theta(j,:);
7     Acc(j,:) = (eul2rotm(ang)*daf(j,:).').';
8 end

```

```

1 function [Theta,Om,tr] = estimate_pose_encoder(mt,Wg1,Wg2,Nw,t)
2 % Estima a rotação 3D e a velocidade angular 3D usando
3 % uma estimativa prévia dos ângulos de Euler e duas estimativas
4 % das velocidades angulares.
5 % utiliza-se a hipótese de coerência interna das variáveis e uma
6 % rede neural do tipo autoencoder sobre uma janela de Nw instantes
7 % de tempo.

```

```

8 %
9 % [Theta,Om,tr] = estimate_pose_encoder(mt,Wg1,Wg2,Nw,t)
10 % mt - (NX3) ângulos de Euler iniciais para cada tempo.
11 % Wg1 - (NX3) estimativa de velocidade angular para cada tempo.
12 % Wg2 - (NX3) estimativa de velocidade angular para cada tempo.
13 % Nw - instantes de tempo usados no ajuste.
14 % t - (N) vetor de tempo.
15
16 % matriz de dados para treinar a rede neural
17 X = [];
18 i = 1;
19 Nt = size(Wg1,1);
20 while i+Nw+1<=Nt % percorrer o conjunto dos pontos
21     fim = min(Nt+i-1,size(Wg1,1));
22     % empilhando os dados na forma de vetor em blocos de 3*Nw
23     % 3 dimensões de dados e Nw pontos do tempo
24     % são empilhadas a estimativa de ângulo e as duas formas da
25     % estimativa de velocidade angular, 3*3*Nw números
26     wg1p = reshape(Wg1(i:i+Nw-1,:),1,3*Nw);
27     wg2p = reshape(Wg2(i:i+Nw-1,:),1,3*Nw);
28     mtp = reshape(mt(i:i+Nw-1,:),1,3*Nw);
29     X = [X [wg1p.'; wg2p.'; mtp.'];
30     i = i+1;
31 end
32
33 % uma rede do tipo autoencoder é treinada com 600 épocas
34 autoenc = trainAutoencoder(X,'MaxEpochs',600);
35 % dados estimados
36 XR = predict(autoenc,X);
37
38 % reconstrução das variáveis dinâmicas por blocos
39 % estimadas pelo autoencoder
40 for i = 1:size(XR,2)
41     % primeira estimativa de velocidade angular
42     XR1 = XR(1:3*Nw,i);
43     % segunda estimativa de velocidade angular
44     XR2 = XR(3*Nw+1:6*Nw,i);
45     % média para obtenção de um valor estimado
46     WR = 0.5*XR1+0.5*XR2;
47     % estimativa dos ângulos de rotação
48     TR = XR(6*Nw+1:9*Nw,i);
49     % ajuste das dimensões
50     TR = reshape(TR,Nw,3);
51     WR = reshape(WR,Nw,3);
52     %tr = t((i-1)*Nw+1:i*Nw);
53     % um ponto é estimado para as três dimensões das duas variáveis
54     % dinâmicas, usando o primeiro ponto da estimativa do autoencoder

```

```

55     Om(i,:) = WR(1,:); % ângulos
56     Theta(i,:) = TR(1,:); % velocidade angular
57 end
58
59 % os últimos pontos da estimativa são aproveitados
60 % do caso última sequência de pontos, pois os pontos não serão
61 % múltiplos de Nw no caso geral
62 Om = [Om; WR(2:end,:)];
63 Theta = [Theta; TR(2:end,:)];
64 No = size(Om,1);
65 % o vetor de tempo é corrigido para o tamanho estimado
66 tr = t(1:No);

```

```

1 function [Wg1,Wg2] = estimate_equation(tm,m,g,t)
2 % Estimar as duas formas de calcular a velocidade angular
3 % usando os sensores. Esses valores serão usados para ajustar
4 % medidas mais robustas.
5 % - A primeira medida é estimada usando o ângulo do magnetômetros.
6 % Determinam-se ângulos de Euler para o ângulo do magnetômetro corrente.
7 % A partir deles aplica-se a matriz de rotação correspondente
8 % sobre a leitura do giroscópio: Wg1.
9 % - A segunda medida é estimada pelo magnetômetro. A taxa de variação
10 % instantânea do magnetômetro deve ser uma rotação pura pela constância
11 % do campo magnético da terra. Assim, essa variação é a velocidade
12 % angular inercial: Wg2.
13 %
14 % [Wg1,Wg2] = estimate_equation(tm,m,g,t)
15 % tm - (NX3) ângulos de Euler para cada instante de tempo de t. ...
    Estimados pelo
16 % magnetômetro.
17 % m - (NX3) leitura do magnetômetro de 3 eixos para cada instante de t.
18 % g - (NX3) leitura do giroscópio de três eixos para cada instante de t.
19 % t - (N) vetor de tempo
20 %
21 % Wg1 - (NX3) primeira estimativa para a velocidade angular
22 % Wg2 - (NX3) segunda estimativa para a velocidade angular
23
24 % intervalo constante de tempo
25 dt = t(2)-t(1);
26 % número de pontos
27 N = size(tm,1);
28 % inicialização das matrizes
29 Wg1 = zeros(N,3);
30 Wg2 = Wg1;
31
32 for i = 2:N

```

```

33     % primeira estimativa
34     % ângulos de Euler em um instante
35     ang = tm(i,:);
36     % leitura do giroscópio
37     w = g(i,:).';
38     % correção para o sistema inercial pela matriz de rotação
39     wg = eul2rotm(ang)*w;
40     Wg1(i,:) = wg.';
41
42     % segunda estimativa
43     % variação instantânea do magnetômetro
44     dm = (m(i,:).'-m(i-1,:).')/dt;
45     % divisão pelo módulo por se tratar de rotação pura
46     dm = dm/norm(m(i,:));
47     Wg2(i,:) = dm;
48 end

```

```

1 function eul = anguloMagnetometro(dmf)
2 % Estimativa inicial de ângulos de Euler ao longo do tempo
3 % usando a leitura filtrado do magnetômetros de três eixos.
4
5 % Média de alguns pontos para definir a direção inicial do campo ...
   magnético.
6 ni = 10;
7 m0 = mean(dmf(3:ni,:));
8
9 [N,~] = size(dmf);
10 eul = zeros(N,3);
11 for i = 1:N % cada instante de tempo
12     m = dmf(i,:);
13     % rotação 3D entre o campo inicial e o campo presente
14     r = vrrotvec(m0,m);
15     % conversão da rotação em eixo e ângulo
16     % para uma matriz de rotação
17     M = vrrotvec2mat(r);
18     % conversão da matriz de rotação para os ângulos de Euler.
19     eul(i,:) = rotm2eul(M);
20 end

```

ANEXO B – CÓDIGO E DOCUMENTAÇÃO PARA GERAÇÃO DE IMAGENS DE MICROSCÓPIO

Os códigos usados no ambiente de desenvolvimento Unity 3D são apresentados a seguir.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5
6 public class HiResScreenShots : MonoBehaviour {
7     public int resWidth = 1024;
8     public int resHeight = 720;
9
10    public int frame = 0;
11
12    public bool errors = true;
13
14    public float ys = -1.5f;
15    public float ye = -1f;
16
17    public float dy = 0.01f;
18
19    public Vector3 initial;
20
21    public string ScreenShotName(int width, int height) {
22
23        Vector3 p = this.transform.position;
24        Vector3 r = this.transform.eulerAngles;
25
26        return string.Format("{0}/reconstruction_errors/"+...
27        "x_{1}_y_{2}_z_{3}_tx_{4}_ty_{5}_tz_{6}.png",
28            Application.dataPath,
29            p.x, p.y, p.z, r.x, r.y, r.z);
30    }
31
32
33    void takePicture() {
34        RenderTexture rt = new RenderTexture(resWidth, ...
35            resHeight, 24);
36        GetComponent<Camera>().targetTexture = rt;
37        Texture2D screenShot = new Texture2D(resWidth, ...
```

```
        resHeight, TextureFormat.RGB24, false);
37 GetComponent<Camera>().Render();
38 RenderTexture.active = rt;
39 screenshot.ReadPixels(new Rect(0, 0, resWidth, ...
        resHeight), 0, 0);
40 GetComponent<Camera>().targetTexture = null;
41 RenderTexture.active = null;
42 Destroy(rt);
43 byte[] bytes = screenshot.EncodeToPNG();
44 string filename = ScreenShotName(resWidth, resHeight);
45 System.IO.File.WriteAllBytes(filename, bytes);
46
47
48
49 }
50
51 public void showCameraPosition() {
52     Debug.Log(this.transform.position);
53 }
54 void Start(){
55     frame = 0;
56     Vector3 ip = this.transform.position;
57     this.initial = new Vector3(this.transform.eulerAngles.x, ...
58     this.transform.eulerAngles.y, this.transform.eulerAngles.z);
59
60     this.transform.position = new Vector3(ip.x, this.y, ip.z);
61     this.errors = true;
62
63 }
64
65
66 void Update() {
67
68
69
70     frame++;
71
72     if(frame%100==0 && this.transform.position.y <= this.ye) {
73
74         float x = this.transform.position.x;
75         float y = this.transform.position.y;
76         float z = this.transform.position.z;
77         float rx = this.initial.x;
78         float ry = this.initial.y;
79         float rz = this.initial.z;
80
81         takePicture();
```



```
82
83     if(frame%1000==0 && this.errors)
84         y+=this.dy;
85     if(!errors)
86         y+=this.dy;
87
88     this.transform.position = new Vector3(x,y,z);
89
90     if(this.errors) {
91         float ex = Random.Range(-5f,5f);
92         float ey = Random.Range(-5f,5f);
93         float ez = Random.Range(-5f,5f);
94         this.transform.eulerAngles = new ...
95             Vector3(rx+ex,ry+ey,rz+ez);
96         //Debug.Log(this.initial);
97     }
98 }
99
100 }
101
102 }
```

ANEXO C – CÓDIGO E DOCUMENTAÇÃO PARA TREINAMENTO DE REDE PARA MICROSCÓPIO

```
1 #!/usr/bin/env python
2 # coding: utf-8
3
4 # # Estimativa de ângulos de posição de imagem
5 #
6 # Estimar ângulação imperfeita de câmera com imagens de fósseis.
7 # O objetivo é corrigir imperfeições de angulação para melhor ...
   reconstrução tridimensional posterior.
8
9 # ## Preparação
10 #
11 # – Pacotes usados
12
13 # In[1]:
14
15 import os
16 import pandas as pd
17 import tensorflow as tf
18 import tensorflow.keras as ks
19 import matplotlib.pyplot as plt
20 from PIL import Image, ImageOps
21 import numpy as np
22
23 from sklearn import preprocessing
24
25 # – Carregando os datasets com os endereços das imagens e os dados de ...
   posicionamento.
26
27 # In[2]:
28
29 ## Pasta Reduced
30
31 all_files = ['reduced/img32_1/img_df32.csv',
32             'reduced/img32_2/img_df32.csv',
33             'reduced/img32_3/img_df32.csv',
34             'reduced/img32_4/img_df32.csv',
35             'reduced/img32_5/img_df32.csv',
36             'reduced/img32_6/img_df32.csv',
37             'reduced/img32_7/img_df32.csv',
38             'reduced/img32_8/img_df32.csv',
```

```
39         'reduced/img32_9/img_df32.csv',
40         'reduced/img32_11/img_df32.csv']
41
42 li = []
43
44
45 for filename in all_files:
46     df = pd.read_csv(filename, index_col=None, header=0)
47     li.append(df)
48
49 T = pd.concat(li, axis=0, ignore_index=True)
50
51 # - Converter imagem para matriz.
52
53 # In[3]:
54
55 file = T['filename'][0]
56 im = Image.open(file)
57
58 imf = ks.preprocessing.image.img_to_array(im)
59 img_shape = imf.shape
60
61
62 # In[4]:
63
64 T.iloc[0, :-1].values
65
66
67 # In[5]:
68
69
70 (1, img_shape[:])
71
72
73 # ### Transformação das imagens:
74 #
75 # Tipicamente o keras somente recebe Numpy arrays então:
76 #
77 # * Alterar o formato para Numpy Array
78 #
79 # * Normalizar
80
81 # In[6]:
82
83
84 # Objeto "normalizador"
85 min_max_scaler = preprocessing.MinMaxScaler()
```

```

86
87
88 #input_images = np.empty((0,img_shape[0],img_shape[1], img_shape[2] ))
89 #input_poses = T.iloc[:,1:4].values
90 #input_y = T.iloc[:,0].values
91 #input_y = np.expand_dims(input_y,axis=1)
92 input_images = np.empty((0,img_shape[0],img_shape[1], img_shape[2] ))
93 input_poses = min_max_scaler.fit_transform(T.iloc[:,1:4].values)
94 input_y = T.iloc[:,0].values
95 input_y = min_max_scaler.fit_transform(np.expand_dims(input_y,axis=1))
96
97 # Diferença das Imagens
98 for t in T['filename']:
99
100     file = t
101     im = Image.open(file)
102     imf = ks.preprocessing.image.img_to_array(im)/255
103     imf = np.expand_dims(imf,axis=0)
104     input_images = np.concatenate((input_images,imf),axis=0)
105
106
107 # ##### Criando a diferença de imagens
108 #
109 # Não está em uso. Notificado como irrelevante na prática para texa ...
110 #     de aproximação.
111 #
112 # Hipótese: a rede absorver uma expectatica instrumental de valores ...
113 #     iniciais. Como a reconstrução final é feita por diferenças, não ...
114 #     há necessidade.
115 #
116 # In[7]:
117
118 norms_poses = np.linalg.norm(input_poses,axis=1)
119
120 imin = np.argmin(norms_poses)
121
122 for i in np.arange(input_images.shape[0]):
123
124     input_images[i] -= input_images[imin]
125
126 # ### Criação dos Datasets de treino e teste
127 #
128 # Separação dos datasets de treino e teste.
129 # Sorteio aleatório de imagens a se utilizar como treinamento ou teste.
130
131 # In[8]:

```

```
130
131 N = input_y.shape[0]
132 frac = 0.7
133 frac = int(np rint (frac*N))
134 ind = np.arange(N)
135 np.random.shuffle(ind)
136 ind_train = ind[:frac]
137 ind_test = ind[frac:]
138
139
140 # In[9]:
141
142 v = []
143 for i in ind_train:
144     v.append([i, 'Train'])
145 for i in ind_test:
146     v.append([i, 'Test'])
147 df_train_test_index = pd.DataFrame(v, columns = ['Index', 'Group'])
148
149
150 # In[10]:
151
152 df_train_test_index.loc[df_train_test_index['Group']=='Test']
153 df_train_test_index.to_csv('train_test_index.csv', index=False)
154
155
156 # In[11]:
157
158
159 input_images_train = input_images[ind_train]
160 input_y_train = input_y[ind_train]
161 input_poses_train = input_poses[ind_train]
162
163 input_images_test = input_images[ind_test]
164 input_y_test = input_y[ind_test]
165 input_poses_test = input_poses[ind_test]
166
167
168 # ## Modelagem
169 #
170 # Definir o modelo de aproximação e a estrutura da rede neural para ...
171     capturar a aproximação.
172
173 # ### Regressão: Definir o ângulo usando uma imagem e a altura
174 #
175 # Modelo combinando uma LeNet com a adição de Y nas camadas densas.
176 #
```

```
176 # Importação de pacotes para definir a estrutura de camadas.
177
178 # In[12]:
179
180
181 from tensorflow.keras.models import Model
182 from tensorflow.keras.models import Sequential
183 from tensorflow.keras.layers import BatchNormalization
184 from tensorflow.keras.layers import AveragePooling2D
185 from tensorflow.keras.layers import MaxPooling2D
186 from tensorflow.keras.layers import Conv2D
187 from tensorflow.keras.layers import Activation
188 from tensorflow.keras.layers import Dropout
189 from tensorflow.keras.layers import Flatten
190 from tensorflow.keras.layers import Input
191 from tensorflow.keras.layers import Dense
192 from tensorflow.keras.layers import concatenate
193
194
195 # In[13]:
196
197
198 entry_size = input_images_train[0].shape
199 input_size = input_y_train[0].shape
200 inputA = Input(shape=entry_size)
201 inputB = Input(shape=input_size)
202
203 x = Conv2D(filters = 6, kernel_size =(5,5), activation='relu', ...
           strides=(1), padding='valid', input_shape= entry_size[1:])(inputA)
204 x = MaxPooling2D(pool_size=(5, 5), strides=(2), padding='same')(x)
205 x = Conv2D(filters = 16, kernel_size =(5,5), activation='relu', ...
           strides=(1), padding='valid', input_shape= entry_size[1:])(x)
206 x = MaxPooling2D(pool_size=(5, 5), strides=(2), padding='same')(x)
207
208 x = Flatten() (x)
209 x = Dense(200, activation="relu") (x)
210 x = Dense(100, activation="relu") (x)
211 #x = Dense(3, activation="relu") (x)
212 x = Model(inputs=inputA, outputs=x)
213
214 layers_input = [input_size, 2]
215 y = inputB
216
217
218 # Combined Layer
219 combined = concatenate([x.output, y])
220 z = Dense(50, activation="relu")(combined)
```

```

221 z = Dense(25, activation="relu")(z)
222 z = Dense(12, activation="relu")(z)
223 z = Dense(6, activation="relu")(z)
224
225 #z = Dense(3, activation="linear")(z)
226 z = Dense(3)(z)
227
228 # Generate Model
229 model = Model(inputs= [x.input, y], outputs=z)
230
231
232 # Espera-se uma estrutura conveza de erro, portanto a estratégia de ...
    otimização é baseadas em um critério também convexo.
233
234 # In[14]:
235
236
237 model.compile( loss = 'mean_squared_error', optimizer='adam', ...
    metrics=['mse'])
238
239 model.summary()
240
241 # Definição dos parâmetros de treinamento e treinamento usando os ...
    dados organizados.
242
243 # In[15]:
244
245 batch_size = 64
246 max_epochs = 500
247
248 h = model.fit(x = [input_images_train, input_y_train], y = ...
    input_poses_train, batch_size= batch_size, epochs= max_epochs, ...
    verbose=1)
249
250 print('\n \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\ DONE \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\')
251
252 # #### Teste de erro quadrático médio
253
254 # In[16]:
255
256 eval_test1 = model.evaluate([input_images_train, input_y_train], ...
    input_poses_train, verbose=0)
257 print("Erro médio do treino: Perda {0:.4f}, mse ...
    {1:.4f}".format(eval_test1[0], eval_test1[1]))
258
259 # In[17]:
260

```

```
261 eval_test2 = model.evaluate([input_images_test, input_y_test], ...
    input_poses_test, verbose=0)
262 print("Erro médio do teste: Perda {0:.4f}, mse ...
    {1:.4f}".format(eval_test2[0], eval_test2[1]))
263
264 # In[18]:
265
266 input_poses_train.max()
267
268 # In[19]:
269
270 input_poses_train.min()
271
272 # In[20]:
273
274 #model.save("img_y_to_p.h5") ### Keras 2.2 old
275 tf.keras.models.save_model(
276     model, "img_y_to_p.h5", overwrite=True, include_optimizer=True, ...
    save_format=None,
277     signatures=None, options=None
278 )
279
280 # In[21]:
281
282 predictions = model.predict([input_images_test, input_y_test])
283 #predictions = model.predict(X_lstm_fill_reg_week)
284 #plt.plot(predictions, "r")
285 #plt.plot(input_poses_test)
286 #plt.grid(True)
287 #plt.legend(['Previsão', 'Real'], loc='upper left')
288 #plt.show()
289
290
291 # #### Comparação gráfica
292 #
293 # Gráficos que comparam os dados de ângulos estimados com os ...
    fornecidos para os três ângulos disponíveis.
294
295 # In[22]:
296
297 n = predictions.shape[0]
298 for i in range(3):
299     plt.subplot(3,1,i+1)
300     plt.plot(np.arange(n), predictions[:,i], np.arange(n), ...
    input_poses_test[:,i])
301
302
303 input_poses_test.shape
```



```

304
305 # ## Regressão: Definir o ângulo e a altura usando a imagem
306 #
307 #
308 # Aqui, para fins de comparação, usa-se a imagem para definir não ...
    apenas os ângulos e a altura. Embora haja uma ambiguidade, a rede ...
    absorve uma ideia de parâmetros médios como referência.
309 #
310 # A arquitetura LeNet foi usada para regressão.
311 #
312 # Os mesmos dados foram usados, agora com as alturas também como ...
    saída da rede.
313
314 # In[23]:
315
316 input_y_poses_train = np.append(input_y_train, input_poses_train, ...
    axis = 1)
317 input_y_poses_test = np.append(input_y_test, input_poses_test, axis = 1)
318
319 np.append(input_y_test, input_poses_test, axis = 1).shape
320 tf.keras.backend.clear_session
321
322 # In[24]:
323
324 entry_size = input_images_train[0].shape
325 inputA = Input(shape=entry_size)
326
327
328 x = Conv2D(filters = 6, kernel_size =(5,5), activation='relu', ...
    strides=(1), padding='valid', input_shape= entry_size[1:])(inputA)
329 x = MaxPooling2D(pool_size=(5, 5), strides=(2), padding='same')(x)
330 x = Conv2D(filters = 16, kernel_size =(5,5), activation='relu', ...
    strides=(1), padding='valid', input_shape= entry_size[1:])(x)
331 x = MaxPooling2D(pool_size=(5, 5), strides=(2), padding='same')(x)
332
333 x = Flatten()(x)
334 x = Dense(200, activation="relu")(x)
335 x = Dense(100, activation="relu")(x)
336 x = Dense(50, activation="relu")(x)
337 x = Dense(25, activation="relu")(x)
338 x = Dense(12, activation="relu")(x)
339 #x = Dense(4, activation="relu")(x)
340 z = Dense(4)(x)
341 model2 = Model(inputs=[inputA], outputs=z)
342
343 model2.compile( loss = 'mean_squared_error', optimizer='adam', ...
    metrics=['mse'])

```

```

344
345 model2.summary()
346
347
348 # Parâmetros de treinamento e treinamento da rede.
349
350 # In[25]:
351
352
353 batch_size = 64
354 max_epochs = 500
355
356 h = model2.fit(x = input_images_train, y = input_y_poses_train, ...
                 batch_size= batch_size, epochs= max_epochs, verbose=1)
357
358 print('\n \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
           DONE      ///////////////////////////////////////////////////')
359
360
361 # In[26]:
362
363 eval_test1 = model2.evaluate(input_images_train, input_y_poses_train, ...
                              verbose=0)
364 print("Erro médio do treino: Perda {0:.4f}, mse ...
         {1:.4f}".format(eval_test1[0], eval_test1[1]))
365
366 # In[27]:
367
368 eval_test2 = model2.evaluate(input_images_test, input_y_poses_test, ...
                              verbose=0)
369 print("Erro médio do teste: Perda {0:.4f}, mse ...
         {1:.4f}".format(eval_test2[0], eval_test2[1]))
370
371
372 # In[28]:
373
374
375 #model2.save("img_to_p_y.h5")
376 tf.keras.models.save_model(
377     model2, "img_to_y_p.h5", overwrite=True, include_optimizer=True, ...
        save_format=None,
378     signatures=None, options=None
379 )

```

```

1 #!/usr/bin/env python
2 # coding: utf-8
3

```

```
4 # # Teste com imagens de fossil não treinado
5 #
6 # As redes treinadas para obter ângulos de imagens tomadas de fósseis ...
   são treinadas novamente para obter os mesmos parâmetros sobre as ...
   imagens de um fóssei não treinado.
7
8 # - Carregar os pacotes.
9
10 # In[1]:
11
12
13 import os
14 import pandas as pd
15 import tensorflow as tf
16 import tensorflow.keras as ks
17 import matplotlib.pyplot as plt
18 from PIL import Image, ImageOps
19 import numpy as np
20
21 from tensorflow.keras.models import Model
22 from tensorflow.keras.models import Sequential
23 from tensorflow.keras.layers import BatchNormalization
24 from tensorflow.keras.layers import AveragePooling2D
25 from tensorflow.keras.layers import MaxPooling2D
26 from tensorflow.keras.layers import Conv2D
27 from tensorflow.keras.layers import Activation
28 from tensorflow.keras.layers import Dropout
29 from tensorflow.keras.layers import Flatten
30 from tensorflow.keras.layers import Input
31 from tensorflow.keras.layers import Dense
32 from tensorflow.keras.layers import concatenate
33
34 from sklearn import preprocessing
35
36
37 # ##### Carregar a lista dos índices utilizados para treino e para ...
   teste:
38
39 # In[2]:
40
41
42 all_files = ['reduced/img32_1/img_df32.csv',
43             'reduced/img32_2/img_df32.csv',
44             'reduced/img32_3/img_df32.csv',
45             'reduced/img32_4/img_df32.csv',
46             'reduced/img32_5/img_df32.csv',
47             'reduced/img32_6/img_df32.csv',
```

```

48         'reduced/img32_7/img_df32.csv',
49         'reduced/img32_8/img_df32.csv',
50         'reduced/img32_9/img_df32.csv',
51         'reduced/img32_11/img_df32.csv',]
52
53 li = []
54
55 for filename in all_files:
56     df = pd.read_csv(filename, index_col=None, header=0)
57     li.append(df)
58
59 T = pd.concat(li, axis=0, ignore_index=True)
60 file = T['filename'][0]
61 im = Image.open(file)
62 imf = ks.preprocessing.image.img_to_array(im)
63 img_shape = imf.shape
64
65
66 # In[3]:
67
68
69 #input_images = np.empty((0,img_shape[0],img_shape[1],img_shape[2]))
70 #input_poses = T.iloc[:,1:4].values
71 #input_y = T.iloc[:,0].values
72 #input_y = np.expand_dims(input_y,axis=1)
73
74 min_max_scaler = preprocessing.MinMaxScaler()
75
76 input_images = np.empty((0,img_shape[0],img_shape[1], img_shape[2] ))
77 input_poses = min_max_scaler.fit_transform(T.iloc[:,1:4].values)
78 input_y = T.iloc[:,0].values
79 input_y = min_max_scaler.fit_transform(np.expand_dims(input_y,axis=1))
80
81 for t in T['filename']:
82
83     file = t
84     im = Image.open(file)
85
86     imf = ks.preprocessing.image.img_to_array(im)/255
87     imf = np.expand_dims(imf,axis=0)
88     input_images = np.concatenate((input_images,imf),axis=0)
89
90
91 # In[4]:
92
93
94 norms_poses = np.linalg.norm(input_poses,axis=1)

```

```
95
96 imin = np.argmin(norms_poses)
97
98 for i in np.arange(input_images.shape[0]):
99
100     input_images[i] -= input_images[imin]
101
102
103 # In[5]:
104
105
106 index_list = pd.read_csv('train_test_index.csv', index_col=None, ...
107     header=0)
108
109 # In[6]:
110
111
112 ind_train = index_list['Index'].loc[index_list['Group'] == ...
113     'Train'].tolist()
114 ind_test = index_list['Index'].loc[index_list['Group'] == ...
115     'Test'].tolist()
116
117 # In[7]:
118
119 #input_images_train = input_images[ind_train]
120 #input_y_train = input_y[ind_train]
121 #input_poses_train = input_poses[ind_train]
122
123 input_images_test = input_images[ind_test]
124 input_y_test = input_y[ind_test]
125 input_poses_test = input_poses[ind_test]
126
127
128 # ##### Group 11 Load
129
130 # In[8]:
131
132
133 group_11 = df = pd.read_csv('reduced/img32_10/img_df32.csv', ...
134     index_col=None, header=0)
135 #group_11 = df = pd.read_csv('reduced/img32_11/img_df32.csv', ...
136     index_col=None, header=0)
137 imf = ks.preprocessing.image.img_to_array(im)
138 img_shape = imf.shape
```

```
137
138
139 # In[9]:
140
141
142 group_11_images = np.empty((0, img_shape[0], img_shape[1], img_shape[2]))
143 group_11_poses = group_11.iloc[:, 1:4].values
144 group_y = group_11.iloc[:, 0].values
145 group_11_y = np.expand_dims(group_y, axis=1)
146
147 for t in group_11['filename']:
148     file = t
149     im = Image.open(file)
150     imf = ks.preprocessing.image.img_to_array(im)/255
151     imf = np.expand_dims(imf, axis=0)
152     group_11_images = np.concatenate((group_11_images, imf), axis=0)
153
154
155
156 # In[10]:
157
158
159 norms_poses = np.linalg.norm(group_11_poses, axis=1)
160
161 imin = np.argmin(norms_poses)
162
163 for i in np.arange(group_11_images.shape[0]):
164
165     group_11_images[i] -= group_11_images[imin]
166
167
168 # In[11]:
169
170
171 model1 = tf.keras.models.load_model('img_y_to_p.h5')
172 model2 = tf.keras.models.load_model('img_to_p_y.h5')
173
174
175 # In[12]:
176
177
178 eval_test1 = model1.evaluate([input_images_test, input_y_test], ...
179                             input_poses_test, verbose=0)
180
181 print("Erro médio do teste: Perda {0:.4f}, mse ...
182       {1:.4f}".format(eval_test1[0], eval_test1[1]))
183
184 eval_test1 = model1.evaluate([group_11_images, group_11_y], ...
```

```
        group_11_poses, verbose=0)
182 print("Erro médio do grupo 11: Perda {0:.4f}, mse ...
        {1:.4f}".format(eval_test1[0], eval_test1[1]))
183
184
185 # In[13]:
186
187
188 group_11_y_poses = np.append(group_11_y, group_11_poses, axis = 1)
189 input_y_poses_test = np.append(input_y_test, input_poses_test, axis = 1)
190
191
192 # In[14]:
193
194 eval_test2 = model2.evaluate(input_images_test, input_y_poses_test, ...
        verbose=0)
195 print("Erro médio do teste: Perda {0:.4f}, mse ...
        {1:.4f}".format(eval_test2[0], eval_test2[1]))
196
197 eval_test2 = model2.evaluate(group_11_images, group_11_y_poses, ...
        verbose=0)
198 print("Erro médio do grupo 11: Perda {0:.4f}, mse ...
        {1:.4f}".format(eval_test2[0], eval_test2[1]))
199
200
201 # In[15]:
202
203
204 predictions = model1.predict([group_11_images, group_11_y])
205
206
207 # In[17]:
208
209
210 n = predictions.shape[0]
211 for i in range(3):
212     plt.subplot(3,1,i+1)
213     plt.plot(np.arange(n), predictions[:,i], np.arange(n), ...
214             group_11_poses[:,i])
```

ANEXO D – CÓDIGO E DOCUMENTAÇÃO PARA GERAÇÃO DE IMAGENS DE FALHAS EM PANEIS

Os códigos usados no ambiente de desenvolvimento Unity 3D são apresentados a seguir.

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 /** class that generates and places faults in a panel object*/
6 public class PlaceFault : MonoBehaviour
7 {
8     // initial positial for all faults
9     private Vector3 initialPosition;
10    // initial scale for all the faultls
11    private Vector3 initialScale;
12    // up direction in the panel reference frame
13    private Vector3 dUp;
14    // down direction in the panel reference frame
15    private Vector3 dDown;
16    // left direction in the panel reference frame
17    private Vector3 dLeft;
18    // right direction in the panel reference frame
19    private Vector3 dRight;
20
21    // coordinates of the four corners of the panel
22    private Vector3 posPanel1, posPanel2, posPanel3, posPanel4;
23
24    // positions of the faults – up to 5
25    private Vector3[] faultsPositions = new Vector3[5];
26
27    // intensities of the faults
28    private float[] faultsIntensity = new float[5];
29
30    // GameObject that are the faults in the World
31    public GameObject[] s;
32
33    // reference to the panel as GameObject
34    public GameObject panel;
35
36    public Vector3[] panelCoordinates() {
37        Vector3 [] pc = new Vector3[4];

```



```
38
39     pc[0] = posPanel1;
40     pc[1] = posPanel2;
41     pc[2] = posPanel3;
42     pc[3] = posPanel4;
43
44     return pc;
45
46 }
47
48 public Vector3[] faultsCoordinates() {
49     return faultsPositions;
50 }
51
52 public float[] faultsSize() {
53     return faultsIntensity;
54 }
55
56 /** generates the faults in random position inside the panel */
57 public void stemFaults() {
58
59     // dimensions of the panel in its own frame of reference
60     float deltaX = 1.45f;
61     float deltaY = 1.8f;
62
63     // random numbers to generate position in x and y
64     float rx;
65     float ry;
66
67     // random numbers to generate scale of each fault
68     float sx;
69     float sy;
70
71     // random number of faults from 0 to 5
72     int nFaults = Random.Range(0, s.Length);
73
74     // all faults start again when this function is called
75     Vector3 p0 = this.initialPosition;
76     Vector3 s0 = this.initialScale;
77
78     // 4 corners of the panel initialised
79     posPanel1 = p0;
80     posPanel2 = p0 + deltaY*s[0].transform.up;
81     posPanel3 = p0 + deltaX*s[0].transform.right;
82     posPanel4 = p0 + ...
            deltaY*s[0].transform.up+deltaX*s[0].transform.right;
83
```

```

84
85     // positions of each fault
86     for(int i = 0; i< nFaults; i++) {
87
88         // fault shown
89         s[i].SetActive(true);
90
91         rx = Random.Range(0f,1f);
92         ry = Random.Range(0f,1f);
93         sx = Random.Range(0.8f,2f);
94         sy = Random.Range(0.8f,2f);
95
96         Vector3 pos = p0;
97         Vector3 scale = s0;
98
99         // position and scale defined
100        s[i].transform.position = pos + ...
            ry*deltaY*s[i].transform.up+rx*deltaX*s[i].transform.right;
101        s[i].transform.localScale = new ...
            Vector3(sx*scale.x,sy*scale.y,scale.z);
102
103        Vector3 fp = s[i].transform.position;
104
105        // record fault position and intensity – intensity taken ...
            as product of two dimensions
106        faultsPositions[i] = fp;
107        faultsIntensity[i] = sx*sy;
108
109    }
110
111    // hide fault objects chosen not to ben shown
112    for(int i = nFaults; i< s.Length; i++) {
113
114        s[i].SetActive(false);
115        // all information for these faults is zero
116        faultsPositions[i] = new Vector3(0f,0f,0f);
117        faultsIntensity[i] = 0;
118
119    }
120
121 }
122
123 void Start()
124 {
125     // initial position and scale made available for when the ...
        function stemFaults
126     // is called again, restarting the fault

```

```

127         this.initialPosition = s[0].transform.position;
128         this.initialScale = s[0].transform.localScale;
129
130
131     }
132
133 }

```

```

1  using UnityEngine;
2  using System.Collections;
3  using System.IO;
4  using System.Globalization;
5  using System.Threading;
6
7  /*** Initialises panel faults, moves camera position and orientation, ...
8  generates images.
9  */
9  public class HiResScreenshots : MonoBehaviour {
10
11     public int resWidth = 1614;
12     public int resHeight = 908; // image resolution
13
14     public static string delim = ", "; // delimiter for de labels' files
15
16     public PlaceFault[] panels; // reference to each photovoltaic panel
17
18     StreamWriter wfpw, wfps, wffw, wffs, camPos; // file objects to ...
19         labels' files
20
21     StreamReader camMove; // file object to camera's positions and ...
22         orientations
23
24     int imageCount = 0; // image identifier and counter
25     int cameraPositionCount = 0; // index of position for camera in ...
26         the file of camera positions
27
28     public Camera cam; // reference to camera object
29     float [] pcam; // data for camera's position and orientation
30
31     public int Ncam = 10; // number of images taken for each camera ...
32         position (each eith new fault generation)
33     public int Nimg = 1; // number of screen updates before an image ...
34         capture
35     public int Ncampos = 20; // image taken every Ncampos camera ...
36         positions

```

```

32
33     public bool capturePhotos = true; // controls if images are ...
           captured into files or not
34
35
36     /** generate file name for image */
37     public static string ScreenShotName(int imgid) {
38         return string.Format("{0}/screenshots/screen_{1}.png",
39                               Application.dataPath,
40                               imgid);
41     }
42
43     /** string with image id and componentes of a Vector3 each line
44     Used to generate a line of labels' file.*/
45     public static string[] vectors2Lines(int id, Vector3[] v) {
46
47         string[] lines = new string[v.Length];
48         string sid = id.ToString();
49
50         for(int i = 0; i<v.Length; i++) {
51             lines[i] = id.ToString() + ", ";
52             lines[i] += v[i].x.ToString() + ", " + v[i].y.ToString() ...
                    + ", " + v[i].z.ToString();
53         }
54
55         return lines;
56     }
57
58     /** String with image id, panel id, components of a Vector3 and a ...
           float for each line.
59     Panel id and image id are repeated. Used to mark coordinates of ...
           faults and its intensity. */
60     public static string shortData2Lines(int id, Vector3[] v, ...
           float[] s, int panel) {
61         string lines = "";
62
63
64         for(int i = 0; i<v.Length; i++) {
65             if(s[i]>0) {
66                 lines += id.ToString() + delim + panel.ToString() + ...
                           delim;
67                 lines += v[i].x.ToString() + delim + ...
                           v[i].y.ToString() + delim + v[i].z.ToString();
68                 lines += delim + s[i].ToString();
69                 if(i<v.Length-1)
70                     lines += "\n";
71             } else {

```

```

72         lines = lines.TrimEnd('\n');
73     }
74 }
75     return lines;
76 }
77
78 /** String with image id, panel id, components of several ...
79     Vector3s (4).
80 Panel id and image id are repeated. Used to mark coordinates of ...
81 the 4 extremities
82 of the panel. */
83 public static string panel2Line(int id, Vector3[] p, int panel) {
84     string line = "";
85     string sid = id.ToString();
86
87     line = id.ToString() + delim + panel.ToString();
88     for(int i = 0; i<p.Length; i++) {
89         line += delim;
90         line += p[i].x.ToString() + delim + p[i].y.ToString() + ...
91         delim + p[i].z.ToString();
92     }
93
94     return line;
95 }
96
97 /** Generates an image from the Main Camera with chosen ...
98     resolution and saves it to a file. */
99 void takePicture2() {
100
101     // The Render Texture in RenderTexture.active is the one
102     // that will be read by ReadPixels.
103     RenderTexture currentRT = RenderTexture.active;
104     RenderTexture.active = this.cam.targetTexture;
105
106     // Render the camera's view.
107     this.cam.Render();
108
109     // Make a new texture and read the active Render Texture into it.
110     Texture2D image = new Texture2D(resWidth, resHeight);
111     image.ReadPixels(new Rect(0, 0, resWidth, resHeight), 0, 0);
112     image.Apply();
113
114     // Replace the original active Render Texture.
115     RenderTexture.active = currentRT;
116
117     byte[] bytes = image.EncodeToPNG();
118     string filename = ScreenShotName(imageCount);

```

```

115     System.IO.File.WriteAllBytes(filename, bytes);
116 }
117
118     /** Generates an image from the Main Camera with chosen ...
119         resolution and saves it to a file. */
119 void takePicture3() {
120     RenderTexture rt = new RenderTexture(resWidth, resHeight, 24);
121     this.cam.targetTexture = rt;
122     Texture2D screenShot = new Texture2D(resWidth, resHeight, ...
123         TextureFormat.RGB24, false);
124
124     this.cam.Render();
125     RenderTexture.active = rt;
126     screenShot.ReadPixels(new Rect(0, 0, resWidth, resHeight), 0, 0);
127     this.cam.targetTexture = null;
128     RenderTexture.active = null;
129     Destroy(rt);
130
131     byte[] bytes = screenShot.EncodeToPNG();
132     string filename = ScreenShotName(imageCount);
133     System.IO.File.WriteAllBytes(filename, bytes);
134 }
135
136 /** Generates an image from the Main Camera with chosen ...
137     resolution and saves it to a file. */
137 void takePicture() {
138
139     Screen.SetResolution(resWidth, resHeight, true);
140     string filename = ScreenShotName(imageCount);
141     ScreenCapture.CaptureScreenshot(filename);
142
143 }
144
145
146 /** Piles a array of Vector3 into an array of floats. */
147 float[] vector3tofloats(Vector3[] v) {
148     int k = 0;
149     float [] data = new float[3*v.Length];
150     for(int i = 0; i<v.Length; i++) {
151         data[k++] = v[i].x;
152         data[k++] = v[i].y;
153         data[k++] = v[i].z;
154     }
155     return data;
156 }
157
158 /** String with numbers separated by comma converted to float ...

```

```

    array. */
159 float[] lineToFloats(string line) {
160     string[] numbers = line.Split(',');
161     float[] v = new float[numbers.Length];
162
163     for(int i = 0; i<numbers.Length; i++) {
164         v[i] = float.Parse(numbers[i]);
165     }
166
167     return v;
168
169 }
170
171 /** Float array to string to save a line in file. */
172 string floatsToLine(float[] a) {
173     string s = "";
174     for(int i = 0; i<a.Length-1; i++) {
175         s += a[i].ToString() + delim;
176     }
177
178     s += a[a.Length-1].ToString();
179
180     return s;
181
182 }
183
184 /** Takes a 6 float vector and uses it as position and ...
    orientation. Applies them to the camera state. */
185 void moveCamera(float [] p) {
186     Vector3 pos = new Vector3(p[0],p[1],p[2]);
187     Vector3 rot = new Vector3(p[3],p[4],p[5]);
188
189     this.cam.transform.position = pos;
190     this.cam.transform.eulerAngles = rot;
191
192 }
193
194 /** Converts a list of Vector3 taken as World Points to Screen Points
195 relatively to the Main Camera. */
196 public Vector3[] screenPoints(Vector3[] pp) {
197     Vector3[] spp = new Vector3[pp.Length];
198
199     float ax = 1f*resWidth/Screen.width;
200     float ay = 1f*resHeight/Screen.height;
201
202     for(int i = 0; i<pp.Length; i++) {
203         spp[i] = this.cam.WorldToScreenPoint(pp[i]);

```

```
204         spp[i].x = spp[i].x*ax;
205         spp[i].y = spp[i].y*ay;
206     }
207
208     return spp;
209 }
210
211 /** write data in file streamers to files and close them. */
212 void closeFiles() {
213     wffw.Flush();
214     wffw.Close();
215
216     wfpw.Flush();
217     wfpw.Close();
218
219     wffs.Flush();
220     wffs.Close();
221
222     wfps.Flush();
223     wfps.Close();
224
225     camMove.Close();
226
227     camPos.Flush();
228     camPos.Close();
229 }
230
231 /** Procedures to end the simulation. */
232 void endSimulation() {
233     // write data in file streamers to files and close them.
234     closeFiles();
235
236     // finish simulator
237     #if UNITY_EDITOR // runing in editor
238         UnityEditor.EditorApplication.isPlaying = false;
239     #else // compiled simulator
240         Application.Quit();
241     #endif
242 }
243
244 /** initial camera position */
245 void initialCamera() {
246     pcam = new float[6];
247     pcam[0] = this.cam.transform.position[0];
248     pcam[1] = this.cam.transform.position[1];
249     pcam[2] = this.cam.transform.position[2];
250     pcam[3] = this.cam.transform.eulerAngles[0];
```



```
251     pcam[4] = this.cam.transform.eulerAngles[1];
252     pcam[5] = this.cam.transform.eulerAngles[2];
253 }
254
255 void Update()
256 {
257
258     if(Input.GetKeyDown(KeyCode.Escape)) { // end of simulation
259         endSimulation();
260
261     }
262
263     if(Time.frameCount % Nimg == 0) { // rate of frames for image ...
264         generation
265
266         for(int i = 0; i<panels.Length; i++) { // each ...
267             photovoltaic panel
268             panels[i].stemFaults(); // generate faults in random ...
269             locations
270
271             float[] fi = panels[i].faultsSize(); // faults intensity
272
273             Vector3[] pc = panels[i].panelCoordinates(); // four ...
274             corners of panels
275             Vector3[] fc = panels[i].faultsCoordinates(); // ...
276             location of faults
277
278             string panpos = panel2Line(imageCount,pc,i); // ...
279             string to write in file
280             string faultpos = ...
281             shortData2Lines(imageCount,fc,fi,i); // string to ...
282             write in file
283
284             //write to file streamers
285             if(panpos.Length>1)
286                 wfpw.WriteLine(panpos);
287             if(faultpos.Length>1)
288                 wffw.WriteLine(faultpos);
289
290             // convert points to screen points
291             pc = screenPoints(pc);
292             fc = screenPoints(fc);
293
294             // convert and save to file streamers
295             panpos = panel2Line(imageCount,pc,i);
296             faultpos = shortData2Lines(imageCount,fc,fi,i);
297
298         }
299     }
```

```
290         if(panpos.Length>1)
291             wfps.WriteLine(panpos);
292         if(faultpos.Length>1)
293             wffs.WriteLine(faultpos);
294
295
296
297     }
298
299     // save image from camera with all the faults placed and ...
300     // locations saved
301     if(capturePhotos)
302         takePicture3();
303
304     string lineCam = floatsToLine(pcam);
305     // save camera position for each image
306     camPos.WriteLine(imageCount.ToString() + delim + lineCam);
307     // image count and identifier in the files
308     imageCount++;
309 }
310
311 if(Time.frameCount % Ncam*Nimg == 0) { // rate of frames for ...
312     // camera positioning
313     if(!camMove.EndOfStream) {
314
315         string line = "";
316         for(int ij = 0; ij<Ncampos; ij++) {
317             // read line for a new camera position
318             if(!camMove.EndOfStream) {
319                 line = camMove.ReadLine();
320             } else {
321                 endSimulation();
322             }
323         }
324
325         if(line.Length>0) {
326             // convert to numbers
327             pcam = lineToFloats(line);
328             // moves camera
329             moveCamera(pcam);
330
331             cameraPositionCount++;
332         }
333     } else { // end of camera positions
334         endSimulation();
```

```
335         }
336
337     }
338
339
340 }
341
342 void Start() {
343     // file streamers
344
345     // corners of panels in world coordinates
346     wfpw = new ...
347         StreamWriter(Application.dataPath+"/labels/panels3D.csv");
348     // corners of panels in screen coordinates
349     wfps = new ...
350         StreamWriter(Application.dataPath+"/labels/panelsScreen.csv");
351     // faults location and intensities in world coordinates
352     wffw = new ...
353         StreamWriter(Application.dataPath+"/labels/faults3D.csv");
354     // faults location and intensities in screen coordinates
355     wffs = new ...
356         StreamWriter(Application.dataPath+"/labels/faultsScreen.csv");
357
358     // choose point (.) as decimal separator
359     Thread.CurrentThread.CurrentCulture = ...
360         CultureInfo.CreateSpecificCulture("en-GB");
361
362     // file with coordinates for camera movement
363     camMove = new ...
364         StreamReader(Application.dataPath+"/camInterp.csv");
365     // initializes camera position
366     initialCamera();
367
368     // number of image taken and associated camera positions and ...
369     // orientations
370     camPos = new ...
371         StreamWriter(Application.dataPath+"/labels/camImage.csv");
372
373     // no physics is need, so no time flow
374     Time.timeScale = 0f;
375
376 }
377 }
```

```
374 /** extension to have a function that concatenates arrays */
375 public static class Extension {
376     public static T[] Concat<T>(this T[] first, params T[] second) {
377         if(first == null) {
378             return second;
379         }
380         if(second == null) {
381             return first;
382         }
383
384         T[] result = new T[first.Length+second.Length];
385         first.CopyTo(result,0);
386         second.CopyTo(result, first.Length);
387
388         return result;
389     }
390 }
```

ANEXO E – CÓDIGO E DOCUMENTAÇÃO PARA DETECÇÃO DE FALHAS EM PANEIS

Geração de máscaras para o treinamento de redes neurais usando os arquivos de dados gerados em simulação.

```
1  #!/usr/bin/env python
2  # coding: utf-8
3
4  # # Generating masks
5  #
6  # This file genertes boolean masks to identify panels and faults in ...
   the imagens generated by the Unity 3D simulation. The masks for ...
   faults are generated separated from the masks for the panels.
7  #
8  # There as masks separated in files (on mask each each) and also a ...
   file that contains all the masks for each kind. Files are saved ...
   in HDF5 format.
9
10 # ## Modules
11 #
12 # - `os` - to read files
13 # - `numpy` - math functions and matrices
14 # - `shapely` - define contours to panel's masks
15 # - `PIL` - image reading
16 # - `h5py`- HDF5 files, good for matrices
17
18 # In[1]:
19
20
21 import os
22 import numpy as np
23 from shapely.geometry import Point, Polygon
24 from PIL import Image
25 import h5py as h5
26
27
28 # ## Masks for panels
29 #
30 #
31 # There is a `csv` file with the screeb coordinates for the four ...
   corners of the panels for every panel in every image.
32 #
```

```

33 # The columns of the `csv` file are the following:
34 # 1. Image id
35 # 2. Panel id
36 # 3. x coordinate of first corner
37 # 4. y coordinate of first corner
38 # 5. x coordinate of second corner
39 # 6. y coordinate of second corner
40 # 7. x coordinate of third corner
41 # 8. y coordinate of third corner
42 # 9. x coordinate of fourth corner
43 # 10. y coordinate of fourth corner
44 #
45 # The file is read and the numbers are taken as points in a plane. Then ...
    the `shapely` module is used when creating a convex polygon that ...
    represents the panel. Screen coordinates inside the polygon are ...
    marked `true` and outside, `false`.
46 #
47
48 # In[9]:
49
50
51 # Using the matrix with all the lines
52 # of the original csv file,
53 # only the lines that refer to a panel
54 # in an image are taken.
55 def linesForPanelImage(panel, image, allLinesMatrix):
56     pc = allLinesMatrix
57
58     linesImg = pc[:,0]==image # selects the image
59     linesPi = pc[:,1]==panel # selects the panel
60     lines = linesPi * linesImg # logic intersection
61
62     return pc[lines,:]
63
64
65 # In[10]:
66
67
68 # creates a list of plane points that are the corners
69 # of the panels in an image.
70 # Selects the columns of the matrix with the
71 # x and y coordinates.
72 def cornersOfPanel(dataMatrix):
73
74     # collects the points in the data matrix
75     c = []
76     for i in range(np panels):

```

```

77         c.append(dataMatrix[:, (2+i*3):(4+i*3)].squeeze())
78
79     # assemble list of points
80     # as objects of appropriate class
81     # of shapely module
82     p = []
83     for pi in c:
84         p.append(Point(pi[0],pi[1]))
85
86     # creates a polygon with the corners and
87     # applies convex hull to determine correct
88     # order of points
89     pg = Polygon(p)
90     pg = pg.convex_hull
91
92     return pg
93
94
95 # In[11]:
96
97
98 # iterates over all images and all panels
99 # collecting the polygons that
100 # define the panels
101 def cornersOfAllPanels(pointsMatrix, image_id, npanels=4):
102     pg = []
103     for pi in range(npanels):
104         panelData = linesForPanelImage(pi, image_id, pointsMatrix)
105         pg.append(cornersOfPanel(panelData))
106
107     return pg
108
109
110 # In[72]:
111
112
113 # uses the polygons in the image coordinates
114 # to generate a boolean mask
115 def generateMask(pg,width,height):
116
117     mask = np.zeros([height,width], dtype=bool)
118
119     for i in range(width):
120         for j in range(height):
121             # every pixel is taken as point
122             p = Point(i,j)
123             for k in range(npanels):

```

```
124         # function 'within' in 'shapely'
125         # tests if a point is
126         # inside a polygon
127         if p.within(pg[k]):
128             mask[j,i] = True
129
130     return np.flipud(mask)
131
132
133 # ### Main procedure
134 #
135 # Generate masks for panels.
136
137 # In[76]:
138
139
140 # folder with images and csv files
141 folder = '../..data'
142 files = os.listdir(folder)
143 panelsCorners = folder+'/panelsScreen.csv'
144
145 # number of panels
146 npanels = 4
147
148 # select a image just to read dimensions
149 files = os.listdir(folder)
150 file = files[10]
151 img = Image.open(folder+'/'+file)
152 width = img.width
153 height = img.height
154
155 # folder where to store masks
156 maskfolder = '../..masks/'
157
158 # file with all masks
159 ft = h5.File(maskfolder+'masks.h5', 'w')
160 # read file
161 pc = np.loadtxt(panelsCorners, delimiter=',')
162
163 for file in files:
164     # select image files saved as 'screen_id.png'
165     if 'screen_' in file:
166
167         image_id = int(file.split('.')[0].split('_')[1])
168
169         # polygons that surround panels in images
170         pg = cornersOfAllPanels(pc, image_id, npanels)
```



```
171     # masks
172     mask = generateMask(pg,width,height)
173
174     # save mask as png image
175     img = Image.fromarray((mask * 255).astype('uint8'),
176                          mode='L')
177     img.save(masksfolder+'m_'+file)
178
179     # add a mask to the file with all masks
180     ft.create_dataset(file[:-4],mask.shape,data=mask)
181
182     # create, save data and close a HDF5 file
183     # with the mask for a single image
184     fi = h5.File(masksfolder+file[:-4]+'.h5','w')
185     fi.create_dataset('data',mask.shape,data=mask)
186     fi.close()
187
188 # close file with all masks
189 ft.close()
190
191
192 # ## Masks for faults
193 #
194 # The masks for the faults are generated as circles around the ...
195 #   faults' center. They are not dimensionless, since the are ellipses ...
196 #   whose area is proportional to the number called severity. Details ...
197 #   for their generation are in the Unity simulation that creates the ...
198 #   images.
199 #
200 # The `csv` files for the faults have columns as follows (each line ...
201 #   contains the data for a fault):
202 #
203 # 1. Image id
204 # 2. Panel id
205 # 3. x screen coordinate
206 # 4. y screen coordinate
207 # 5. severity
208 #
209 # The procedure in this case is simpler and thus not divided into ...
210 #   several methods.
211 #
212 # In[86]:
213
214 # folder with images and csv files
215 folder = '../..data'
```

```

212 files = os.listdir(folder)
213 file = files[10]
214 img = Image.open(folder+'/'+file)
215 width = img.width
216 height = img.height
217
218 # folder where to store the masks
219 maskfolder = '../..masks/'
220
221 # file for the masks of all images
222 ft = h5.File(maskfolder+'faults_masks.h5', 'w')
223
224 # csv file with faults locations and severity
225 faultsCorners = folder+'/faultsScreen.csv'
226 fc = np.loadtxt(faultsCorners, delimiter=',')
227
228 # creating a pair of matrices containing
229 # x and y coordinates of pixels
230 # they are used in the calculation
231 # of a circle around a pixel
232 x = np.arange(width)
233 y = np.arange(height)
234 Y,X = np.meshgrid(y,x)
235
236 # iterating over the image files
237 for file in files:
238     if 'screen_' in file:
239
240         mask = np.zeros([height,width], dtype=bool)
241         # identifies image id as in the file name
242         # screen_id.png
243         iimg = int(file.split('.')[0].split('_')[1])
244         # faults of a particular image
245         faultsIndex = fc[:,0]==iimg
246         position = fc[faultsIndex,2:4]
247         severity = fc[faultsIndex,-1]
248
249         # for every fault, calculate a circle
250         # around it proportional to the severity
251         for pos,sev in zip(position,severity):
252
253             # the circle around the fault
254             # must not escape the borders
255             # of the image
256             Xd = np.clip(X-pos[0],0,width-1)
257             Yd = np.clip(Y-pos[1],0,height-1)
258             # circle inequation

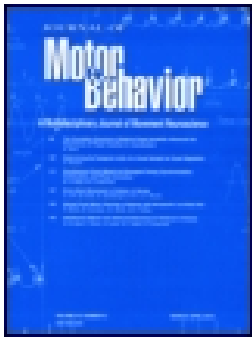
```

```
259         insideFault = Xd*Xd+Yd*Yd<10*sev*sev
260
261         # logical 'or' to include another
262         # fault in the mask for current image
263         mask = np.logical_or(mask, insideFault)
264
265         # save mask as png image
266         img = Image.fromarray((mask * 255).astype('uint8'),
267                               mode='L')
268         img.save(masksfolder+'fm_'+file)
269
270         # add current mask to the file
271         ft.create_dataset(file[:-4],mask.shape,data=mask)
272
273         # create, stores and closes
274         # mask for current image in a separate file
275         fi = h5.File(masksfolder+'fm_'+file[:-4]+'.h5', 'w')
276         fi.create_dataset('data',mask.shape,data=mask)
277         fi.close()
278
279 # close file for all images
280 ft.close()
```

ANEXO F – PUBLICAÇÕES RELACIONADAS COM ESTA TESE

As publicações científicas, algumas já publicadas e outras a serem publicadas podem ser vistas nos anexos:

- Estudo de foco no ato de sentar e levantar usando medidas de ângulos obtidos com sensores inerciais de aparelhos de celular. (Submetido.)
- Estudo de foco na pirueta do balé usando medidas de ângulos obtidos com sensores inerciais de aparelhos de celular. (Publicado.)
- Estudo sobre o cálculo de uma medida de controle motor no ato de sentar e levantar usando dados de sistemas de aquisição de movimento que foram tratados usando redes neurais do tipo autoencoder. (Publicado.)
- Otimização do procedimento de reconstrução 3D de um microscópio pela otimização da medida de posição. (Correção.)



Attention Focus Does Not Influence Performance of Sit-to-Stand in Young and Older Adults

Valéria A Pinto, Alice B Campolina, Alysson F Mazoni, Daniela J S Mattos & Daniela V Vaz

To cite this article: Valéria A Pinto, Alice B Campolina, Alysson F Mazoni, Daniela J S Mattos & Daniela V Vaz (2020): Attention Focus Does Not Influence Performance of Sit-to-Stand in Young and Older Adults, Journal of Motor Behavior, DOI: [10.1080/00222895.2020.1723477](https://doi.org/10.1080/00222895.2020.1723477)

To link to this article: <https://doi.org/10.1080/00222895.2020.1723477>



Published online: 11 Feb 2020.



[Submit your article to this journal](#)



Article views: 8



[View related articles](#)



[View Crossmark data](#)

ARTICLE

Attention Focus Does Not Influence Performance of Sit-to-Stand in Young and Older Adults

Valéria A Pinto¹ , Alice B Campolina¹ , Alysson F Mazoni² , Daniela J S Mattos³ , Daniela V Vaz⁴ 

¹Rehabilitation Sciences Graduate Program, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil. ²Physical Education Graduate Program, Universidade Estadual de Campinas, Campinas, Brazil. ³Washington University School of Medicine in Saint Louis, Saint Louis, Missouri, United States. ⁴Department of Physical Therapy, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil.

ABSTRACT. An external focus of attention can improve performance, but there is little research on effects for the elderly in every day, well-learned mobility tasks. 57 older and 59 young adults performed the sit-to-stand and stand-to-sit while holding a cup, at three difficulty levels (cup empty or full, at normal or fast speed). Half were instructed to focus internally (on their movements) and half externally (on the cup). The effects of focus, age, and difficulty level were tested for movement time, mean inclination of the cup, inclination variability, and smoothness with $2 \times 2 \times 3$ ANOVAs. Significant effects of difficulty were consistent across variables ($p < 0.05$). An effect of focus was present only for the inclination variability of the stand-to-sit ($p < 0.03$), favoring an internal focus (less variability). The age \times focus interaction was significant for mean cup inclination, but *post hoc* tests failed to reveal any significant differences. The results of this study, together with the literature, suggest that an external focus may not benefit the performance of young or older adults in general mobility activities of daily living. The prevalent assumption that an external focus is always beneficial for performance needs further empirical testing.

Keywords: Elderly, focus of attention, mobility

Introduction

Attention can improve motor performance (Wulf, Shea, & Lewthwaite, 2010). Attention can be directed to the effects of movement on the environment (external focus, EF) or to movement itself (internal focus, IF). Extensive literature indicates that an EF produces better performance and learning on a variety of tasks (for a review, see Wulf, 2013). Benefits of an external over an IF of attention are seen in movement efficiency (e.g., muscular activity, force production, cardiovascular responses, etc). In particular, increased muscle fiber recruitment, increased force production, and more effective movement coordination under an EF can potentially increase movement speed (Fasoli, Trombly, Tickle-Degnen, & Verfaellie, 2002; Porter, Nolan, Ostrowski, & Wulf, 2010; Totsika & Wulf, 2003). Positive effects are also observed for movement effectiveness (e.g., accuracy, stability, etc., Wulf, 2013). The reason, according to the “constrained action hypothesis” (Wulf, McNevin, & Shea, 2001; Wulf, Shea, & Park, 2001), is that an IF may induce conscious control that interferes with automatic coordination, causing

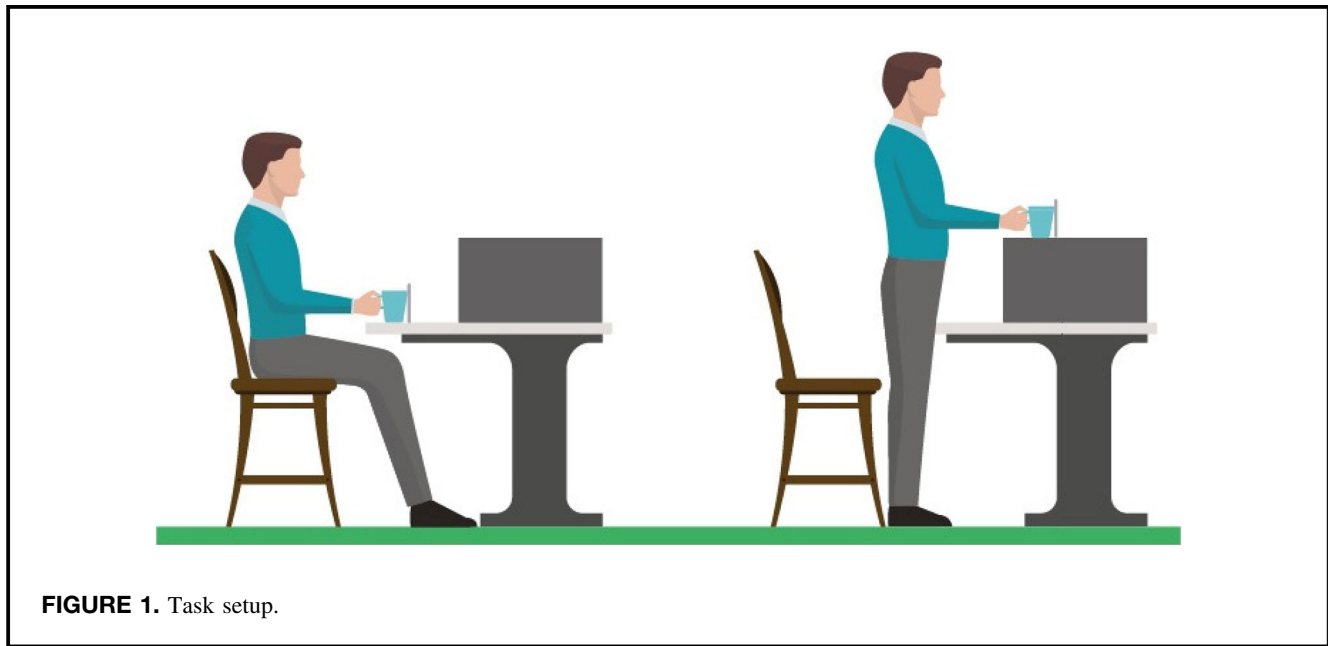
performance to suffer. An EF, on the contrary, would favor unconscious, fast and reflexive processes, resulting in greater movement fluidity.

According to the “constrained action hypothesis,” the beneficial effects of an EF are especially salient in difficult tasks, when individuals would attempt to consciously intervene in body movement more frequently (Landers, Wulf, Wallmann, & Guadagnoli, 2005; Wulf, Töllner, & Shea, 2007). Therefore, to guarantee sufficiently challenging tasks, most research has focused on inexperienced individuals performing novel sports-related tasks (Wulf, 2007, 2013), while the activities of daily living have received less attention. Would an EF improve the performance of well-learned activities of daily living, such as sit-to-stand and stand-to-sit?

The sit-to-stand and stand-to-sit are fundamental for independence and become more difficult with age. Sit-to-stand and stand-to-sit require greater hip joint moments than stair climbing or walking (Rodosky, Andriacchi, & Andersson, 1989). Additionally, good control of balance is required to deal with the rapid shift of body mass between the seat and the feet (Riley, Schenkman, Mann, & Hodge, 1991). With age-related decreases in muscle strength and balance control, the sit-to-stand and stand-to-sit become more difficult, and many older adults perform the task close to their maximal abilities (Hughes, Myers, & Schenkman, 1996). Deterioration of sit-to-stand and stand-to-sit performance in older adults is a key indicator of decreased mobility and increased risk of falls (Buatois et al., 2008).

Very frequently, the sit-to-stand and stand-to-sit are performed in association with manual tasks that pose additional control challenges, such as holding a cup full of liquid (Muhaidat, Kerr, Evans, Pilling, & Skelton, 2014). For example, one may be sitting at a table in a cafe, stand up and walk away with a coffee cup. Acceleration and orientation of the cup must be controlled to avoid spilling (Togo, Kagawa, & Uno, 2012) and movement time may increase to accommodate precise stabilization of the cup. The time difference between

Correspondence address: Daniela Virginia Vaz, Department of Physical Therapy, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil. E-mail: danielavaz@gmail.com



the Timed Up and Go test (which involves Si-St and St-Si) with and without holding a cup appears to be a valid marker of frailty and fall risk (Muhaidat et al., 2014; Tang, Yang, Peng, & Chen, 2015; Togo et al., 2012). From an experimental point of view, holding a cup also makes for a useful experimental model because it creates a natural external referent to which attention may or may not be directed during sit-to-stand and stand-to-sit, depending on instructions. Previous studies have shown that an EF on a supra-postural task goal increases movement effectiveness (McNevin & Wulf, 2002; Wulf, Mercer, McNevin, & Guadagnoli, 2004; Wulf, Weigelt, Poulter, & McNevin, 2003).

It is possible that a simple behavioral intervention with EF instructions during sit-to-stand and stand-to-sit would be especially beneficial for the elderly. Older adults are presumably more inclined to consciously control their movements in challenging tasks (Woollacott & Shumway-Cook, 2002). If the “constrained action hypothesis” is correct, conscious attention to body movement (IF) impairs automaticity and fluidity of movement. EF instructions, in contrast, may increase movement fluency, regularity, and speed (Kal, Van Der Kamp, & Houdijk, 2013). Given the fundamental importance of sit-to-stand and stand-to-sit for independence, positive effects of EF instructions could generate interesting clinical applications in rehabilitation treatments for the elderly.

This study, therefore, investigated whether attention focus instruction can have any impact on the well-learned daily life activity of sit-to-stand and stand-to-sit holding a cup, for young and older adults, in three different task difficulty levels. We hypothesized that focus instructions would interact with age and difficulty level,

being especially beneficial in more difficult conditions and for older people, at the level of movement outcome. We expected that EF instructions would produce greater movement efficiency, that is, shorter movement times. We also expected EF instructions would produce greater movement effectiveness with respect to the overall goal of keeping the cup vertical and stable during the transfer to avoid spilling. More specifically, we expected that for older people in the most difficult condition, the EF would lead to faster sit-to-stand and stand-to-sit transitions, and to cup trajectories with less inclination (more accuracy), and increased smoothness and less variability (more stability).

Method

Participants

The inclusion criteria for this study were: (1) age between 18 and 40 years or over 60; (2) no musculoskeletal symptoms affecting sit-to-stand and stand-to-sit; (3) no cognitive disorder affecting the ability to follow instructions. Participants that felt pain or discomfort during the task (2 older adults) or who were unwilling to complete it (1 young and 4 older adults) were excluded. A total of 59 healthy young adults (44 females) and 57 healthy older adults (41 females) signed consent for participation (approved by the Institution’s Ethics Committee) and completed the study.

Task and Apparatus

Participants stood up and sat down from a chair (0.47 m high) holding and transferring a cup (with a

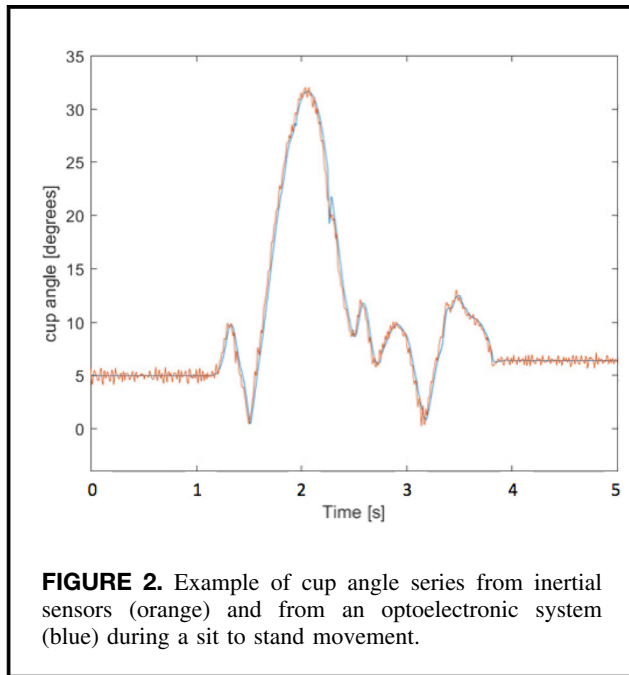


FIGURE 2. Example of cup angle series from inertial sensors (orange) and from an optoelectronic system (blue) during a sit to stand movement.

smartphone attached) between two surfaces of different heights (0.79 and 1.07 m) [Figure 1](#)). A Motorola smartphone (Android XT1058) with Sensor Kinetics Pro (Innoventions, Inc.) with a magnetometer, a gyroscope, and linear acceleration sensors was used to record the data.

Procedures

Data collection of daily life tasks in ecologically valid situations has been greatly facilitated by the development of valid and reliable smartphone technology (Boonstra et al., 2006; Galán-Mercant, Barón-López, Labajos-Manzanares, & Cuesta-Vargas, 2014; Nishiguchi et al., 2012). We used an android-based application and sensors after comparison with data from an optoelectronic system (10 cameras, Oqus Qualisys, Sweden), a gold standard for kinematic analysis. Four retro-reflective markers were placed on the smartphone. One participant performed five repetitions of the task in each of three different task difficulty levels. We expected that dependent measures averaged over five repetitions would be representative of typical performance in each experimental condition. Angle time series collected simultaneously from the two systems were compared. [Figure 2](#) shows an example of a cup angle series from the two systems in a sit-to-stand movement. The relative difference between the two series, averaged over time, with the Qualisys as a reference, varied from 0.26 to 0.29%. These tests indicated the validity of sensor data.

In line with a clinical trial rationale, participants were assigned to one of two intervention groups in counterbalanced order, as they enrolled for the study: EF

instructions (29 young and 27 older adults) or IF instructions (30 young and 30 older adults). All participants sat on a chair ([Figure 1](#)) and were instructed to grab the cup with their non-dominant hand (according to self-reported handedness) and transfer it from the lower to the higher surface as they rose from the chair, or transfer it from the higher to the lower surface as they sat down, always looking straight ahead. The EF group was instructed to “think all the time about the cup and the liquid inside the cup.” The IF group was instructed to “think about your own arm and the coordination of your movements”.

Participants performed three blocks of five trials each, under three difficulty levels: (1) empty cup at normal speed (EN); (2) full cup at normal speed (FN); and (3) full cup at a fast speed (FF). Normal and fast speed were self-chosen for each participant. For normal speed, participants were told to perform the task as they usually do in daily life. For fast speed, they were told to perform the task as fast as they could without spilling liquid. Colored adhesive tape was used to mark and maintain a standard level of liquid in the cup (1 cm below the rim). In case of spilling, the trial was discarded, the liquid was refilled to the mark and the participant was asked to repeat the trial. Focus instructions were reinforced before each condition.

Participants then answered three questions: (1) *what did you focus on while performing the task?*; (2) *were you able to follow the instruction of attention focus?*; and (3) *on a scale of 0 to 10, how well did you follow the instruction?*

Data Reduction

Given the requirements of smartphones’ operating systems, the main issue with their inertial sensors is the variability of acquisition rate (30–90 Hz). After spectral density analysis showed no relevant power above 10 Hz, linear interpolation was used to achieve a fixed common sampling frequency of 30 Hz for all three sensors. Data was then filtered with a low pass Butterworth filter of order 3 and cutoff frequency of 10 Hz. An automated Matlab (MathWorks Inc.) routine aided by visual analysis of the accelerometer time series determined time-stamps for the start and end of each sit-to-stand and stand-to-sit. Movement time was defined in seconds.

The angle (radians) of the cup with respect to the global vertical was calculated. The magnetometer was used to mark a three-dimensional vector whose variation from an initial position is taken as an inclination (the cup and smartphone were vertical while resting on a table before beginning and after the end of the movement). The inclination was then projected to the vertical axis to calculate the smartphone angle (parallel to the cup). Magnetometer signals are noisy so data from the other sensors are used to improve it. The magnetometer signal is interpolated to optimally reduce the error of its

TABLE 1. Significance values (p) for ANOVAs including participants who reported attention content appropriate to instructions (91).

	Movement time		Inclination average		Inclination variability		Smoothness	
	Sit to Stand	Stand to Sit	Sit to Stand	Stand to Sit	Sit to Stand	Stand to Sit	Sit to Stand	Stand to Sit
Age	0.648	0.788	0.335	0.349	0.951	0.121	0.578	0.578
Focus	0.746	0.636	0.745	0.990	0.224	0.002	0.891	0.891
Difficulty	0.001	0.001	0.003	0.002	0.027	0.014	0.001	0.001
Age*Focus	0.425	0.247	0.042	0.029	0.632	0.847	0.086	0.086
Difficulty*Age	0.040	0.054	0.961	0.943	0.681	0.760	0.809	0.809
Difficulty *Focus	0.705	0.220	0.995	0.979	0.751	0.536	0.632	0.632
Difficulty*Age*Focus	0.334	0.544	0.714	0.788	0.481	0.356	0.941	0.941

Statistically significant values are in bold.

derivatives compared to the gyroscope and accelerometer. The resulting signal is an estimate of the cup angle. The average and standard deviation of the cup angle over time, for the duration of a sit-to-stand and stand-to-sit, were obtained for each trial.

Smoothness is a measure of the shape of a movement time series. While jerky and irregular movements have low smoothness, steady, regular, and fluent movements are smoother. Smoothness was calculated with the negative spectral arc-length measure, as defined by Balasubramanian, Melendez-Calderon, and Burdet (2012). For each cup angle speed profile $v(t)$, $t \in [0, T]$ and duration T , we generated its Fourier magnitude spectrum. Then negative of the arc length is calculated as

$$\eta_{sal} \triangleq - \int_0^{\omega_c} \sqrt{\left(\frac{1}{\omega_c}\right)^2 + \left(\frac{d\hat{V}(\omega)}{d\omega}\right)^2} d\omega$$

$$\hat{V}(\omega) \triangleq \frac{V(\omega)}{V(0)}$$

where $V(\omega)$ is the Fourier magnitude spectrum of $v(t)$, and $[0, \omega_c]$ is the frequency band occupied by the cup movement. Greater values of this measure indicate smoother movements.

Statistical Analysis

Means and standard deviations (mean \pm SD) were used as descriptive statistics. Participants' mean age was compared between IF and EF groups with independent samples t -tests. A chi-square test was used to compare the frequency of males and females between IF and EF groups. The two-proportion z test was used to test whether the frequency of discarded trials (due to spilling) was different between IF and EF groups. Adherence to instructions was compared across groups with Fisher's exact tests for categorical answers (question 2) and a 2 (age) \times 2 (focus) ANOVA for score-based answers (question 3). Sit-to-stand and stand-to-sit performance variables were analyzed separately. The dependent variables of interest were the average and standard deviation

of cup angle over time, smoothness and movement time. Data were analyzed with a 2 (Age) \times 2 (Focus) \times 3 (Difficulty level) analysis of variance (ANOVA), with repeated measures on the last factor. All statistics were calculated using the Statistical Package for the Social Sciences Version 21.0 (SPSS for Windows, Chicago, IL). Statistical significance was set at $p < 0.05$.

Results

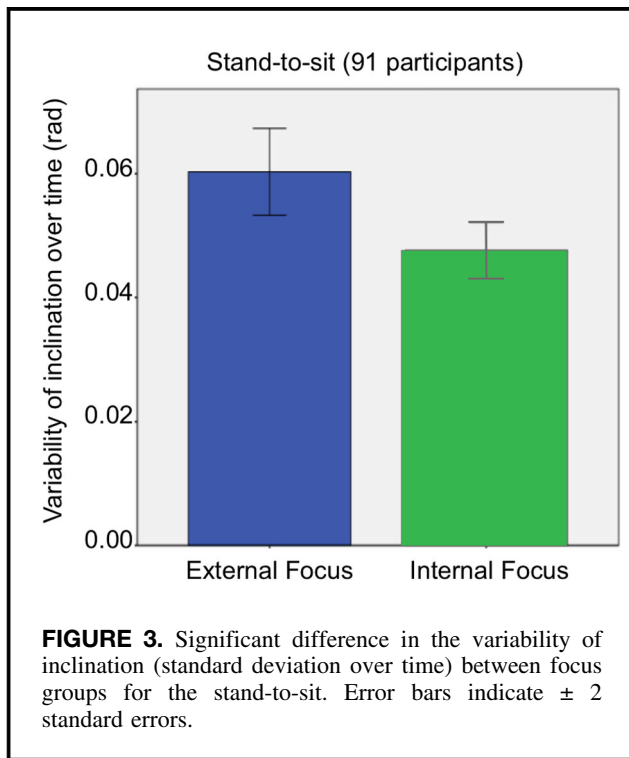
Participant Characteristics in the Two Attention Instruction Groups

A total of 116 participants (57 right-handed in IF and 50 right-handed in EF, 3 left-handed in IF and 6 left-handed in EF) took part in this study. The frequency of females and males was not statistically different ($p = 0.823$) among young participants in the IF (22 females, 8 males) compared to the EF (22 females, 7 males) group, or among old participants ($p = 0.152$) in the IF (24 females, 6 males) compared to the EF (17 females, 10 males) group. Mean age also did not differ ($p = 0.199$) between young participants in IF (24.90 ± 3.26) and EF (23.72 ± 3.68) groups (overall mean: 24.32 ± 3.50) or old participants ($p = 0.532$) in IF (68.37 ± 5.60) and EF (69.37 ± 6.46) groups (overall mean: 68.84 ± 5.99).

Ability to Follow Instructions

For the question "were you able to follow the instruction of attention focus?" the proportion of "Yes" responses among old participants for EF (96.3%) and IF (89.7%) were not statistically different ($p = 0.612$). The proportion of "Yes" responses among young participants for EF (96.6%) and IF (100%) were also not statistically different ($p = 0.491$).

For the question "on a scale of 0 to 10, how well did you follow the instruction?" the average scores for the older adults under EF and IF instructions were respectively, 8.61 ± 1.09 and 8.62 ± 1.30 . The average scores



for the young adults under EF and IF instructions were respectively, 8.41 ± 0.92 and 8.05 ± 1.10 . Age, Focus, and the Age \times Focus interaction were not significant ($p > 0.063$).

The content of answers to “*what did you focus on while performing the task?*” revealed, however, that many individuals had difficulty to focus on actual internal content. A total of 8 of the 30 older adults (26.6%) and 13 of the 30 young adults (43.3%) in the IF group gave answers indicating content inappropriate to received instruction. For example, some participants answered that they had “focused on not spilling,” or “on looking straight ahead instead of looking at the cup.” In contrast, 2 of the 27 older adults (7.40%) and 2 of the 29 young adults (6.89%) in the EF group gave answers indicating content inappropriate to received instruction.

Thus, we ran statistical ANOVAs of the effects of EF and IF on performance only for the 91 participants whose answers ensured they had used attention content that was appropriate to their respective instructions. Table 1 shows all ANOVA p values.

Performance (Movement Outcome Measures)

Table 1 shows that the main effect of Difficulty was significant for all variables. The effects of Difficulty were clear in movement time, which was significantly different ($F_{2, 174} = 62.616$, $p = 0.001$, partial $\eta^2 = 0.419$ for sit-to-stand and $F_{2, 174} = 52.518$, $p = 0.001$, partial $\eta^2 = 0.376$ for stand-to-sit) between the three difficulty levels: empty cup at normal speed (3.707 ± 0.113 for sit-

to-stand and 4.046 ± 0.121 for stand-to-sit); full cup at normal speed (4.545 ± 0.129 for sit-to-stand and 4.913 ± 0.150 for stand-to-sit); and full cup at a fast speed (3.778 ± 0.103 for sit-to-stand and 4.002 ± 0.123 for stand-to-sit).

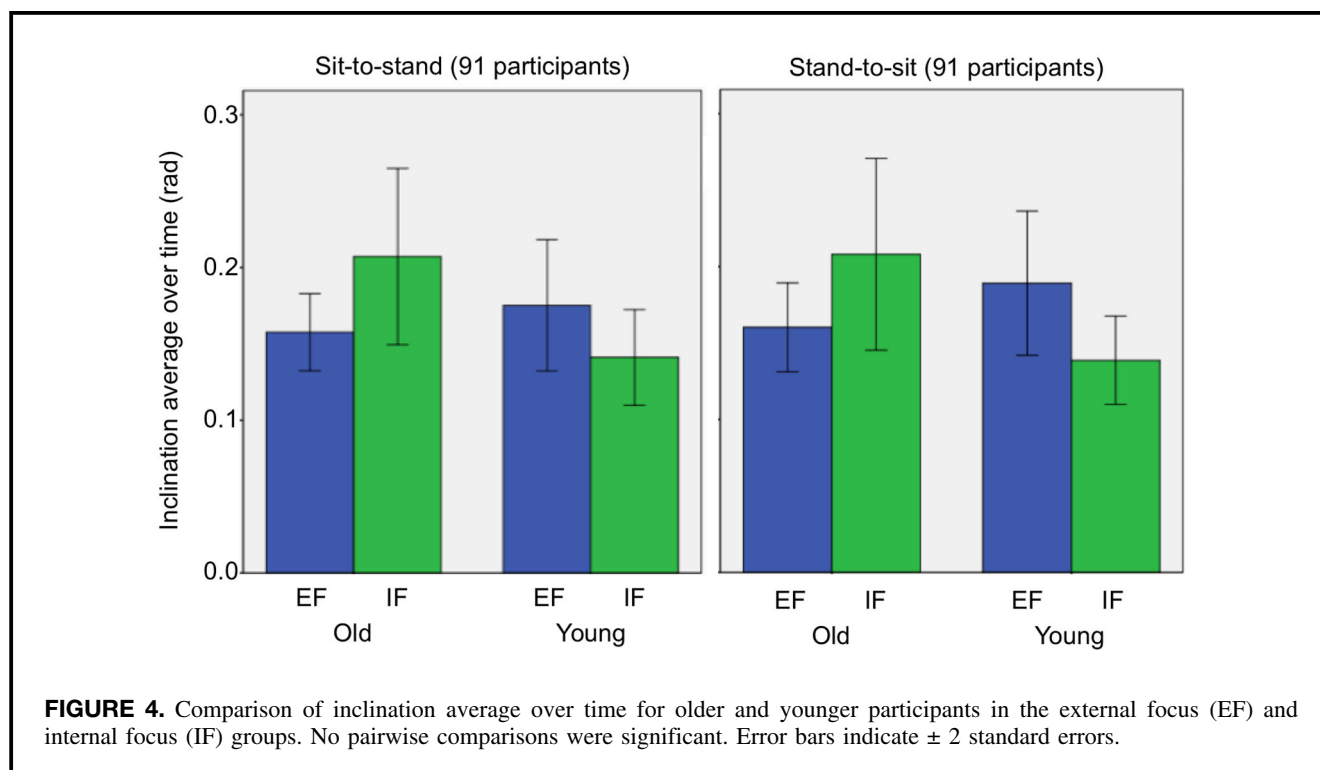
However, there were no significant differences in movement time ($F_{2, 174} = 0.106$, $p = 0.746$, partial $\eta^2 = 0.001$ for sit-to-stand and $F_{2, 174} = 0.226$, $p = 0.636$, partial $\eta^2 = 0.003$ for stand-to-sit) between the IF (4.044 ± 0.160 for sit-to-stand and 4.377 ± 0.180 for stand-to-sit) and EF groups (3.976 ± 0.137 for sit-to-stand and 4.264 ± 0.154 for stand-to-sit). No significant interaction effects involving Focus were significant ($p \geq 0.220$).

Movement time also did not differ significantly ($F_{2, 174} = 0.210$, $p = 0.648$, partial $\eta^2 = 0.002$ for sit-to-stand and $F_{2, 174} = 0.073$, $p = 0.788$, partial $\eta^2 = 0.001$ for stand-to-sit) between young participants (3.962 ± 0.153 for sit-to-stand, 4.352 ± 0.172 for stand-to-sit) and old participants (4.058 ± 0.145 for sit-to-stand, 4.288 ± 0.163 for stand-to-sit). There were significant Difficulty \times Age interaction effects for movement time ($F_{2, 174} = 3.284$, $p = 0.040$, partial $\eta^2 = 0.036$ for sit-to-stand and $F_{2, 174} = 2.974$, $p = 0.054$, partial $\eta^2 = 0.033$ for stand-to-sit), suggesting that difficulty may affect movement time for young and old participants differently. Given that the main effect of difficulty was quite consistent across variables, and that the focus of our analysis was on Focus, but not Difficulty or Age effects, these interactions were not further investigated.

A significant main effect of Focus was present only for inclination variability of the stand-to-sit, ($F_{1, 87} = 10.131$, $p = 0.002$, partial $\eta^2 = 0.104$). The group average values (IF: 0.049 ± 0.003 ; EF: 0.063 ± 0.003) indicate that variability of angle was significantly higher for EF compared to IF (Figure 3).

An Age \times Focus interaction effect was significant only for the average inclination angle during the sit-to-stand ($F_{1, 87} = 4.266$, $p = 0.042$, partial $\eta^2 = 0.047$ for sit-to-stand and $F_{1, 87} = 4.945$, $p = 0.029$, partial $\eta^2 = 0.054$ for stand-to-sit). However, Bonferroni-corrected post hoc independent t tests showed no differences for sit-to-stand for the young participants ($p = 0.267$) between IF (0.141 ± 0.064) and EF (0.176 ± 0.116), or for old participants ($p = 0.127$) between IF (0.207 ± 0.135) and EF (0.158 ± 0.063). Results were similarly not significant in the stand-to-sit for young participants ($p = 0.121$) between IF (0.139 ± 0.060) and EF (0.190 ± 0.122), or for old participants ($p = 0.177$) between IF (0.208 ± 0.147) and EF (0.160 ± 0.073). These results are shown in Figure 4. No other interactions involving Focus were significant.

The frequency of discarded trials (due to spilling) did not differ ($p = 0.144$) between the groups receiving IF (12 out of 900 trials) or EF instructions (19 out of 840 trials).



Discussion

The effects of attention focus on activities of daily living are rarely investigated. Adequate sit-to-stand and stand-to-sit performances are fundamental for maintaining independence in old age. Positive effects of focus instructions could be used in rehabilitation applications to improve the performance of this task. Thus, our trial investigated whether focus instruction interventions had any impact on performance (at the level of movement outcome) of the well-learned activity of sit-to-stand and stand-to-sit while holding a cup, for young and older adults, at three difficulty levels. We hypothesized that in the most difficult condition, for older people, an EF would lead to greater movement effectiveness, that is, less cup inclination, lower variability, and increased smoothness. The results did not support our hypothesis.

We failed to find significant focus effects except for worse angle stability under EF compared to IF for the stand-to-sit. However, this effect was not consistent, as all other performance variables showed null focus effects. Our null results are surprising in view of the conclusion of a literature review indicating that the enhancements in motor performance with an EF compared to IF are well established. The review author states: “The breadth of this effect is reflected in its generalizability to different skills, levels of expertise, and populations...” (Wulf, 2013, p. 99). Our results are inconsistent with this claim. In our study, an EF did not enhance the motor performance of sit-to-stand and stand-

to-sit while holding a cup, a skill that involves body transfer and object manipulation (Gentile, 2000), regardless of difficulty level and population. What factors may explain these null results?

First, we need to point out that we controlled for adherence to instructions. Self-reported adherence scores were similar across conditions and groups. However, several individuals (21%) reported focusing on content inconsistent with the instructions they had received. Our analysis included only individuals with appropriate attention content. Therefore the lack of focus effects cannot be attributed to inadequate adherence to instruction.

Second, our results are consistent with many recent studies involving day-to-day posture and mobility skills. Despite some previous research showing benefits of an EF for these kinds of skills (Chiviacowsky, Wulf, & Wally, 2010; McNevin, Weir, & Quinn, 2013; Richer, Saunders, Polskaia, & Lajoie, 2017), several studies report null effects for focus instructions for posture and mobility skills (De Bruin, Swanenburg, Betschon, & Murer, 2009; Landers, Hatlevig, Davis, Richards, & Rosenlof, 2016; Mak, Young, Chan, & Wong, 2018; Melker Worms et al., 2017; Richer, Polskaia, & Lajoie, 2017; Yogeve-Seligmann, Sprecher, & Kodesh, 2017).

Richer, Polskaia, and Lajoie (2017) found no difference between IF and EF for control of quiet stance in older adults. For gait performance, no effects on walking stability or balance recovery after gait perturbations were found for older adults (Melker Worms et al., 2017). Yogeve-Seligmann et al. (2017) reported that gait

variability could not be improved by focusing on keeping steps consistent or focusing on pacing gait to the rhythm of a metronome. Both focus instructions actually increased the variability of some spatiotemporal gait parameters. Mak et al. (2018) found that although IF appears to compromise gait stability, EF instructions did not improve gait stability compared to a control condition in older adults. Benefits of an EF were again not found in a randomized controlled trial on the learning of balance skills for the healthy elderly (De Bruin et al., 2009) or patients with Parkinson's Disease (Landers et al., 2016). No studies examining the effects of attention focus on the performance of the sit-to-stand and stand-to-sit were found. Our study appears to be the first on the topic, and our results are consistent with many experiments involving activities of daily living.

In the attention focus literature, the lack of benefits of EF instructions has been attributed to different factors. Researchers have argued that the benefits of an EF do not apply to movement tasks (i) that do not involve implements and have no clearly intended environmental effect (Melker Worms et al., 2017); (ii) that are too easy (Landers et al., 2016; Wulf, 2008); or (iii) that were learned in early childhood without declarative knowledge (Melker Worms et al., 2017). We will argue below that the first two reasons are not pertinent to our study, with the third reason being the most probable explanation for our results.

The first argument is that the benefits of an EF would not apply to movement tasks that do not involve action on specific objects. Usually, during sit-to-stand and stand-to-sit, the individual does not intend to produce any specific effects on external objects. In such tasks, an EF may in fact not benefit performance (see, e.g., Lawrence, Gottwald, Hardy, & Khan, 2011). In this study, however, we associated an object-manipulation goal to the sit-to-stand and stand-to-sit. This ensured a natural external reference to which attention could naturally be directed, depending on instructions. Our performance variables specifically reflect effectiveness to control the environmental effects of movement: the cup average angle, its stability, and smoothness. Thus, we expected that the benefits of an EF would apply to the performance of our task, but no advantages of an EF were found. Also, the lack of effects on movement time suggests that sit-to-stand and stand-to-sit, as a whole, were not affected.

Second, the literature indicates that an EF is purportedly more beneficial in difficult tasks, because it would prevent attempts to consciously intervene in body movement (Landers et al., 2005; Wulf, 2008; Wulf et al., 2007). To avoid a lack of effects due to unchallenging conditions, our task had three difficulty levels. Our design is limited in that it did not include a possible intermediate difficulty condition with an empty cup at

fast speed. However, performance results show that our difficulty manipulation significantly affected all variables, for both age groups.¹

The sit-to-stand and stand-to-sit with a full cup at the fastest possible speed correspond to the most difficult real-life version of the task. With no EF benefits on movement effectiveness and movement time for this version of the task, effects in any other less challenging, ecologically valid versions are unlikely.

This brings us to the third, most probable explanation for results: possibly, general postural and mobility skills that are acquired spontaneously during normal motor development with little declarative instruction (phylogenetic skills such as the sit-to-stand) are less vulnerable to interferences of attention focus (Melker Worms et al., 2017; Young & Mark Williams, 2015). Specialized complex skills learned later in life (ontogenetic skills such as sports gestures), in contrast, are usually acquired with great amounts of explicit instruction in early practice (Masters & Maxwell, 2008). For these tasks, an IF may revert the individual back to an earlier declarative stage of learning and interfere with the automaticity of control, while an EF might prioritize relevant, goal-related information for fluent coordination (Melker Worms et al., 2017; Young & Mark Williams, 2015). We speculate that because the sit-to-stand is a phylogenetic mobility skill, it would be less prone to the negative effects of an IF or the positive effects of an EF.

Interpretations of this study's results in the context of the available literature for general postural and mobility activities of daily living suggest that an EF of attention may not benefit the performance of healthy young and older adults in well-learned tasks. They indicate that the assumption that an EF is to be always preferred (Wulf, 2013, 2016; Wulf et al., 2007) needs further empirical testing for activities of daily living. This study is limited in that it did not assess coordination but only performance measures at the level of movement outcome. An EF might positively affect the coordination of postural and mobility tasks for example in individuals with neurological health conditions that impair automaticity of movement.

Note

1. Note that the small difference in movement time (0.089s) between EN and FF does not invalidate our classification of difficulty. Participants used similar times in these two conditions because when the cup was full, they had to slow down to avoid spilling. When the cup was empty, they felt comfortable moving faster as there weren't any negative consequences. FF is the hardest and EN is the easiest of the three conditions.

Funding

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

ORCID

Valéria A Pinto  <http://orcid.org/0000-0003-3140-510X>
 Alice B Campolina  <http://orcid.org/0000-0001-5253-1485>
 Alysson F Mazoni  <http://orcid.org/0000-0001-5265-6894>
 Daniela J S Mattos  <http://orcid.org/0000-0003-3396-1951>
 Daniela V Vaz  <http://orcid.org/0000-0003-0470-6361>

REFERENCES

- Balasubramanian, S., Melendez-Calderon, A., & Burdet, E. (2012). A robust and sensitive metric for quantifying movement smoothness. *IEEE Transactions on Biomedical Engineering*, *59*(8), 2126–2136. doi:10.1109/TBME.2011.2179545
- Boonstra, M. C., Van Der Slikke, R. M. A., Keijsers, N. L. W., Van Lummel, R. C., De Waal Malefijt, M. C., & Verdonchot, N. (2006). The accuracy of measuring the kinematics of rising from a chair with accelerometers and gyroscopes. *Journal of Biomechanics*, *39*(2), 354–358. doi:10.1016/j.jbiomech.2004.11.021
- Buatois, S., Miljkovic, D., Manckoundia, P., Gueguen, R., Miget, P., Vançon, G., ... Benetos, A. (2008). Five times sit to stand test is a predictor of recurrent falls in healthy community-living subjects aged 65 and older. *Journal of the American Geriatrics Society*, *56*(8), 1575–1577. doi:10.1111/j.1532-5415.2008.01777.x
- Chiviackowsky, S., Wulf, G., & Wally, R. (2010). An external focus of attention enhances balance learning in older adults. *Gait & Posture*, *32*(4), 572–575. doi:10.1016/j.gaitpost.2010.08.004
- De Bruin, E. D., Swanenburg, J., Betschon, E., & Murer, K. (2009). A randomised controlled trial investigating motor skill training as a function of attentional focus in old age. *BMC Geriatrics*, *9*(1), 15. doi:10.1186/1471-2318-9-15
- Fasoli, S. E., Trombly, C. A., Tickle-Degnen, L., & Verfaellie, M. H. (2002). Effect of instructions on functional reach in persons with and without cerebrovascular accident. *American Journal of Occupational Therapy*, *56*(4), 380–390. doi:10.5014/ajot.56.4.380
- Galán-Mercant, A., Barón-López, J. J., Labajos-Manzanares, M. T., & Cuesta-Vargas, A. I. (2014). Reliability and criterion-related validity with a smartphone used in timed-up-and-go test. *BioMedical Engineering Online*, *13*(1), 156. doi:10.1186/1475-925X-13-156
- Gentile, A. M. (2000). Skill acquisition: Action, movement, and neuromotor processes. In J. H. Carr, & R. H. Shepherd (Eds.), *Movement Science: Foundations for Physical Therapy in Rehabilitation* (pp. 111–187). Gaithersburg, MD: Aspen.
- Hughes, M. A., Myers, B. S., & Schenkman, M. L. (1996). The role of strength in rising from a chair in the functionally impaired elderly. *Journal of Biomechanics*, *29*(12), 1509–1513. (96)80001-7 doi:10.1016/S0021-9290(96)80001-7
- Kal, E. C., Van Der Kamp, J., & Houdijk, H. (2013). External attentional focus enhances movement automatization: A comprehensive test of the constrained action hypothesis. *Human Movement Science*, *32*(4), 527–539. doi:10.1016/j.humov.2013.04.001
- Landers, M. R., Hatlevig, R. M., Davis, A. D., Richards, A. R., & Rosenlof, L. E. (2016). Does attentional focus during balance training in people with Parkinson's disease affect outcome? A randomised controlled clinical trial. *Clinical Rehabilitation*, *30*(1), 53–63. doi:10.1177/0269215515570377
- Landers, M., Wulf, G., Wallmann, H., & Guadagnoli, M. (2005). An external focus of attention attenuates balance impairment in patients with Parkinson's disease who have a fall history. *Physiotherapy*, *91*(3), 152–158. doi:10.1016/j.physio.2004.11.010
- Lawrence, G. P., Gottwald, V. M., Hardy, J., & Khan, M. A. (2011). Internal and external focus of attention in a novice form sport. *Research Quarterly for Exercise and Sport*, *82*(3), 431–441. doi:10.1080/02701367.2011.10599775
- Mak, T. C. T., Young, W. R., Chan, D. C. L., & Wong, T. W. L. (2018). Gait stability in older adults during level-ground walking: The attentional focus approach. *The Journals of Gerontology: Series B*, *75*, 274–281. doi:10.1093/geronb/gby115
- Masters, R., & Maxwell, J. (2008). The theory of reinvestment. *International Review of Sport and Exercise Psychology*, *1*(2), 160–183. doi:10.1080/17509840802287218
- McNevin, N., Weir, P., & Quinn, T. (2013). Effects of attentional focus and age on suprapostural task performance and postural control. *Research Quarterly for Exercise and Sport*, *84*(1), 96–103. doi:10.1080/02701367.2013.762321
- McNevin, N. H., & Wulf, G. (2002). Attentional focus on supra-postural tasks affects postural control. *Human Movement Science*, *21*(2), 187–202. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/12167298>. doi:10.1016/S0167-9457(02)00095-7
- Melker Worms, J. L. A., Stins, J. F., van Wegen, E. E. H., Verschueren, S. M. P., Beek, P. J., & Loram, I. D. (2017). Effects of attentional focus on walking stability in elderly. *Gait & Posture*, *55*, 94–99. doi:10.1016/j.gaitpost.2017.03.031
- Muhaidat, J., Kerr, A., Evans, J. J., Pilling, M., & Skelton, D. A. (2014). Validity of simple gait-related dual-task tests in predicting falls in community-dwelling older adults. *Archives of Physical Medicine and Rehabilitation*, *95*(1), 58–64. doi:10.1016/j.apmr.2013.07.027
- Nishiguchi, S., Yamada, M., Nagai, K., Mori, S., Kajiwara, Y., Sonoda, T., ... Aoyama, T. (2012). Reliability and validity of gait analysis by android-based smartphone. *Telemedicine and e-Health*, *18*(4), 292–296. doi:10.1089/tmj.2011.0132
- Porter, J. M., Nolan, R. P., Ostrowski, E. J., & Wulf, G. (2010). Directing attention externally enhances agility performance: A qualitative and quantitative analysis of the efficacy of using verbal instructions to focus attention. *Frontiers in Psychology*, *1*(NOV), 216. doi:10.3389/fpsyg.2010.00216
- Richer, N., Polskaia, N., & Lajoie, Y. (2017). Continuous cognitive task promotes greater postural stability than an internal or external focus of attention in older adults. *Experimental Aging Research*, *43*(1), 21–33.
- Richer, N., Saunders, D., Polskaia, N., & Lajoie, Y. (2017). The effects of attentional focus and cognitive tasks on

- postural sway may be the result of automaticity. *Gait & Posture*, 54, 45–49. doi:10.1016/j.gaitpost.2017.02.022
- Riley, P. O., Schenkman, M. L., Mann, R. W., & Hodge, W. A. (1991). Mechanics of a constrained chair-rise. *Journal of Biomechanics*, 24(1), 77–85. (91)90328-K doi:10.1016/0021-9290(91)90328-K
- Rodosky, M. W., Andriacchi, T. P., & Andersson, G. B. J. (1989). The influence of chair height on lower limb mechanics during rising. *Journal of Orthopaedic Research*, 7(2), 266–271. doi:10.1002/jor.1100070215
- Tang, P. F., Yang, H. J., Peng, Y. C., & Chen, H. Y. (2015). Motor dual-task timed up & go test better identifies pre-frailty individuals than single-task timed up & go test. *Geriatrics & Gerontology International*, 15(2), 204–210. doi:10.1111/ggi.12258
- Togo, S., Kagawa, T., & Uno, Y. (2012). Motor synergies for dampening hand vibration during human walking. *Experimental Brain Research*, 216(1), 81–90. doi:10.1007/s00221-011-2909-3
- Totsika, V., & Wulf, G. (2003). The influence of external and internal foci of attention on transfer to novel situations and skills. *Research Quarterly for Exercise and Sport*, 74(2), 220–232. doi:10.1080/02701367.2003.10609084
- Woollacott, M., & Shumway-Cook, A. (2002). Attention and the control of posture and gait: A review of an emerging area of research. *Gait & Posture*, 16(1), 1–14. (01)00156-4 doi:10.1016/S0966-6362(01)00156-4
- Wulf, G. (2007). Attentional focus and motor learning: A review of 10 years of research (Target article). *E-Journal Bewegung und Training [E-Journal Movement and Training]*, 1, 4–14. Retrieved from http://www.sportwissenschaft.de/fileadmin/pdf/BuT/hossner_wulf.pdf
- Wulf, G. (2008). Attentional focus effects in balance acrobats. *Research Quarterly for Exercise and Sport*, 79(3), 319–325. doi:10.1080/02701367.2008.10599495
- Wulf, G. (2013). Attentional focus and motor learning: A review of 15 years. *International Review of Sport and Exercise Psychology*, 6(1), 77–104. doi:10.1080/1750984X.2012.723728
- Wulf, G. (2016). An external focus of attention is a condition sine qua non for athletes: A response to Carson, Collins, and Toner (2015). *Journal of Sports Sciences*, 34(13), 1293–1295.
- Wulf, G., McNevin, N., & Shea, C. H. (2001). The automaticity of complex motor skill learning as a function of attentional focus. *The Quarterly Journal of Experimental Psychology Section A*, 54(4), 1143–1154. doi:10.1080/713756012
- Wulf, G., Mercer, J., McNevin, N., & Guadagnoli, M. A. (2004). Reciprocal influences of attentional focus on postural and suprapostural task performance. *Journal of Motor Behavior*, 36(2), 189–199. doi:10.3200/JMBR.36.2.189-199
- Wulf, G., Shea, C., & Lewthwaite, R. (2010). Motor skill learning and performance: A review of influential factors. *Medical Education*, 44(1), 75–84. doi:10.1111/j.1365-2923.2009.03421.x
- Wulf, G., Shea, C., & Park, J. H. (2001). Attention and motor performance: Preferences for and advantages of an external focus. *Research Quarterly for Exercise and Sport*, 72(4), 335–344. doi:10.1080/02701367.2001.10608970
- Wulf, G., Töllner, T., & Shea, C. H. (2007). Attentional focus effects as a function of task difficulty. *Research Quarterly for Exercise and Sport*, 78(3), 257–264. doi:10.1080/02701367.2007.10599423
- Wulf, G., Weigelt, M., Poulter, D., & McNevin, N. (2003). Attentional focus on suprapostural tasks affects balance learning. *The Quarterly Journal of Experimental Psychology Section A*, 56(7), 1191–1211. doi:10.1080/02724980343000062
- Young, W. R., & Mark Williams, A. (2015). How fear of falling can increase fall-risk in older adults: Applying psychological theory to practical observations. *Gait & Posture*, 41(1), 7–12. doi:10.1016/j.gaitpost.2014.09.006
- Yogev-Seligmann, G., Sprecher, E., & Kodesh, E. (2017). The effect of external and internal focus of attention on gait variability in older adults. *Journal of Motor Behavior*, 49(2), 179–184. doi:10.1080/00222895.2016.1169983

Received April 5, 2019

Revised October 29, 2019

Accepted October 29, 2019



Internal and Imagined External Foci of Attention Do Not Influence Pirouette Performance in Ballet Dancers

Carolina M. Andrade, Thales R. de Souza, Alysson F. Mazoni, André G. P. de Andrade & Daniela V. Vaz

To cite this article: Carolina M. Andrade, Thales R. de Souza, Alysson F. Mazoni, André G. P. de Andrade & Daniela V. Vaz (2020): Internal and Imagined External Foci of Attention Do Not Influence Pirouette Performance in Ballet Dancers, Research Quarterly for Exercise and Sport, DOI: [10.1080/02701367.2019.1698697](https://doi.org/10.1080/02701367.2019.1698697)

To link to this article: <https://doi.org/10.1080/02701367.2019.1698697>



Published online: 05 Feb 2020.



Submit your article to this journal [↗](#)



View related articles [↗](#)



View Crossmark data [↗](#)



Internal and Imagined External Foci of Attention Do Not Influence Pirouette Performance in Ballet Dancers

Carolina M. Andrade ^a, Thales R. de Souza ^a, Alysson F. Mazoni ^b, André G. P. de Andrade ^a, and Daniela V. Vaz ^a

^aUniversidade Federal de Minas Gerais; ^bUniversidade Estadual de Campinas Cidade Universitária Zeferino Vaz

ABSTRACT

Ballet dancers need to constantly improve their performance. Several studies show that an internal focus (on body movements) leads to inferior motor performance relative to an external focus of attention (on the movement effects), but the majority of dancers usually adopt an internal focus. It is not clear if the benefits of an external focus are relevant for form-based tasks performed in a ballet. **Purpose:** Test whether an imagined external focus of attention can improve ballet dancers' performance of a form-based task, the *en dehors* double *pirouette* (EDDP). **Method:** 23 experienced amateurs and 17 professional dancers between 18 and 35 years old performed three EDDP under each of three attentional focus instruction conditions (No Focus, Internal, Imagined External). Order of presentation of Internal and Imagined External focus instructions was blocked and balanced across participants. Movement quality was assessed by three experts with a customized observational scale. Movement kinematics were measured by the angle of inclination of the trunk and inclination of the rotation axis with respect to the global vertical during the EDDP. **Results:** Movement quality and trunk inclination were significantly different between amateurs and professionals, but the rotation axis inclination was not. No quantitative or qualitative effects of focus were found for the performance of the EDDP across focus conditions. **Conclusions:** There appears to be no differential effects of using an internal or imagined external focus on the immediate performance of the EDDP for amateur and professional ballet dancers. Personal and task factors might account for the lack of effects.

ARTICLE HISTORY

Received 20 September 2018
Accepted 22 November 2019

KEYWORDS

Focus of attention; dance; motor control

Classical ballet dancers are constantly seeking performance improvements. The combinations of motor skills involved in ballet are some of the most complex. Dancers must develop very refined coordination and accuracy to achieve the desired esthetics (Laws, 2002). Important questions for dancers and teachers, therefore, regard how performance can be enhanced (Guss-West & Wulf, 2016). Other than the intensive practice itself, instructions and feedback given to practitioners are possibly the most important means to promote skill learning. Significantly, the effectiveness of instructions or feedback is determined not only by their information content, but also by the way in which they direct the learner's attention (Abdollahipour, Wulf, Psotta, & Nieto, 2015). To support learning and consolidation of skill, what should dancers pay attention to when practicing or performing ballet moves? How should ballet teachers direct the dancers' attention? More generally, how can the effectiveness of instructions be enhanced?

Across different fields that involve teaching motor skills, coaches, physical therapists, and teachers often

provide their clients with instructions that promote an internal focus, that is, a focus on one's own movements. Ballet dancers, either spontaneously or as a result of those instructions, also typically focus on body movements (Guss-West & Wulf, 2016). There is extensive evidence, however, that focusing away from the body on the intended movement effects (external focus) can result in superior performance and learning relative to an internal focus on body movements (Wulf, 2013).

These appear to be not just temporary benefits, they last over time: an external focus leads to superior performance in delayed retention and transfer tests for several motor skills such as balancing, jumping and throwing (Wulf, 2007). The "constrained action hypothesis" (Wulf, McNevin, & Shea, 2001) has been put forward to explain the advantages of an external focus of attention. According to this hypothesis, on the one hand, a focus on the effects of movement on the environment would favor unconscious, fast and reflexive process, resulting in greater movement ease or fluidity. On the other hand, a focus on the movement

itself would induce conscious control that interferes with automatic coordination processes, causing performance to suffer.

Traditions and widely accepted practices in ballet emphasize internally, body-focused attention during practice. Given the emphasis of ballet is on movement form and the absence of implements, an internal focus seems natural (Peh, Chow, & Davids, 2011; see also Wrisberg, 2007). Can the practical wisdom and tradition followed by teachers be actually non-optimal for improving their students' performance? Investigations comparing the effects of internal versus external focus in form tasks (e.g., gymnastics or dance) are very scarce (Abdollahipour et al., 2015; Lawrence, Gottwald, Hardy, & Khan, 2011).

One might wonder how a ballet dancer can focus externally in the absence of implements. One possibility to create external attentional cues is to simply attach an external object such a marker or sticker to the performer. In a recent study, gymnasts performed a maximum vertical jump with a 180-degree turn with hands crossing in front of their chest while airborne. Both jump height and movement quality were significantly improved when gymnasts were asked to concentrate on the direction to which a tape marker on their chest was pointing, compared with instructions to focus on the direction to which their hands (in front of their chest) were pointing, or a control condition (Abdollahipour et al., 2015).

Another possibility is to suggest images, analogies, or metaphors that can prevent an internal focus on the details of movements and induce a focus on the effects of movements, consistently with an external focus (Guss-West & Wulf, 2016; Wulf, 2007). Images like "stretching like a star in all directions" when performing an *arabesque*, "jumping over a lake" while performing a *grand jeté* or "climbing up a corkscrew" during a *pirouette* can direct attention away movement details and on to the intended effect or outcome of the action (Guss-West & Wulf, 2016). Imagining interaction with objects like an orange to be stepped on during *elevé*, or a pencil to be kept vertical when stretching the trunk upwards can direct attention outwards, creating an imagined external focus.

Both theory and evidence open up the possibility for qualitative and quantitative improvements in fundamental and difficult ballet movements. One such movement is the *pirouette*. The *pirouette* is a rapid, complete turn of the body around the longitudinal axis on a single foot, on the toes or ball of this foot. *Pirouettes* pose a significant challenge for the maintenance of whole body stability. The rotation axis of the trunk is not fixed and changes during the turn (Lin, Chen, Su, Wu, & Lin, 2014; Lin, Su, Wu, & Lin, 2013).

Alignment of the rotation axis with the vertical line is essential for dancers to maintain postural stability during the *pirouette*. Deviations of the rotation axis can threaten stability and result in failures to turn or falls (Laws, 2002). Thus, skilled dancers exhibit significantly less inclination of the rotation axis and of the trunk from the vertical (Lin et al., 2014).

It is possible that simple changes in movement instructions could help dancers improve *pirouette* performance? The *pirouette* looks like an ideal form-based skill to test the effects of attention instructions on performance, because the advantages of focusing externally are especially pronounced when the skill is difficult or complex (Wulf, 2007; Wulf, Töllner, & Shea, 2007). The objective of this study was to investigate whether the focus of attention instructions affects *pirouette* performance of professional and amateur *ballet* dancers. We expected that performance would be improved by instructions inducing an imagined external focus, away from the body, compared to an internal focus of attention or no focus instructions.

Methods

Participants

Participants of this study were 40 ballet dancers: 23 amateurs and 17 professionals, 22 females and 18 males aged between 18 and 35 years old. The inclusion criteria were specified as follows: (1) be 18 to 35 years old, (2) have at least 4 years of classical ballet training, (3) be currently practicing classical ballet. Professionals were employed as dancers by a company, while amateurs were not. Participants who presented pain, balance dysfunction or any other health issue affecting the performance of the *pirouette* were excluded.

Procedures

Each participant read and signed an informed consent form approved by the University Institutional Review Board. To enhance ecological validity (Davids, 1988), data collection was conducted in each participant's usual practice setting. Participants were asked to perform a specific and standardized classical ballet move: the *en dehors* double *pirouette* (EDDP). This task consists of two sequential rapid full revolutions of the body with single-leg support on balls of the foot (*demi-pointe*). Dancers prepared in ballet fourth position, placing the gesture leg behind the support leg with the trunk upright. Then, they quickly flexed both legs and started the turn bringing the gesture leg to the *passé* position, with the toes of the gesture leg in front

or directly to the side of the support leg knee as it extends. They finished the movement placing the foot of the gesture leg on the ground, close to the support leg, in ballet fifth position (Lin et al., 2014, 2013). See Figure 1 for a schematic representation of an EDDP.

Participants chose their preferred support leg to perform the EDDP. They were asked to perform three EDDP with no specific focus instructions, pausing to rest for as long as needed after each trial. After these initial control trials with no focus (NF), participants received instructions to use an internal focus (IF) on the body, or instructions to imagine an external object and focus on it. Although this kind of image-based instructions to focus away from the body are consistent with an external focus (Guss-West & Wulf, 2016; Wulf, 2007), there is some controversy as to whether they should be classified as actual external (Collins, Carson, & Toner, 2016). For this reason, we chose to name this condition the imagined external focus (IEF).

IF and IEF sequencing was blocked and balanced across participants. Participants performed three EDDP (with rest pauses) under each focus condition. IF instructions were: “During the *pirouettes*, think about keeping your body as aligned as possible. Focus on your body.” IEF instructions were: “During the *pirouettes*, think that there is a taut cable pulling you to the ceiling. Focus on the cable.” These instructions were chosen so that both would be equally relevant in terms of the movement outcome: keeping the body erect, straight and stable during the revolutions on the leg. This way, possible performance differences between the two experimental conditions would be attributed to the locus of attention, and not to differences in the relevance to the desired outcome.

After all 9 trials, each participant marked on a form an answer to the question “How well were you able to

follow instructions to think about keeping your body as aligned as possible?” and “... to think that there is a taut cable pulling you to the ceiling?”, in a scale that varied from 1 (poorly) to 10 (excellently).

Instruments

Naturalness of dance moves can be disrupted by data collection procedures in novel (laboratory) environments with equipment that is unusual for dancers (Davids, 1988). To avoid such interference, this study used portable inertial sensors to allow for data collection at each dancer’s usual practice setting with their usual clothing. This procedure allowed for collecting data at dance companies and thus having more dancers volunteering for the study, given their busy practice schedules. A small smartphone equipped with inertial sensors (magnetometer, gyroscope and linear acceleration sensors) was attached with an elastic band to the lumbar segment right above the fifth lumbar vertebrae (out of the participant’s view). The use of smartphone sensors and applications been shown to be valid and reliable for obtaining quality kinematic data (Galán-Mercant & Cuesta-Vargas, 2013; Nishiguchi et al., 2012). The application Sensor Kinetics Pro for Android (Innoventions, Inc.) was used to collect the global vertical, angular velocity and linear acceleration from the sensors. A digital video camera was used to record all 360 EDDP experimental trials (9 per participant).

An optoelectronic system (8 cameras, Qualisys, Sweden), a gold-standard for kinematic analysis, was also used to produce reference data for comparison with the inertial sensor data. The smartphone and 14 retro-reflective markers were placed, according to standard recommendations (Wu et al., 2005), on a volunteer’s

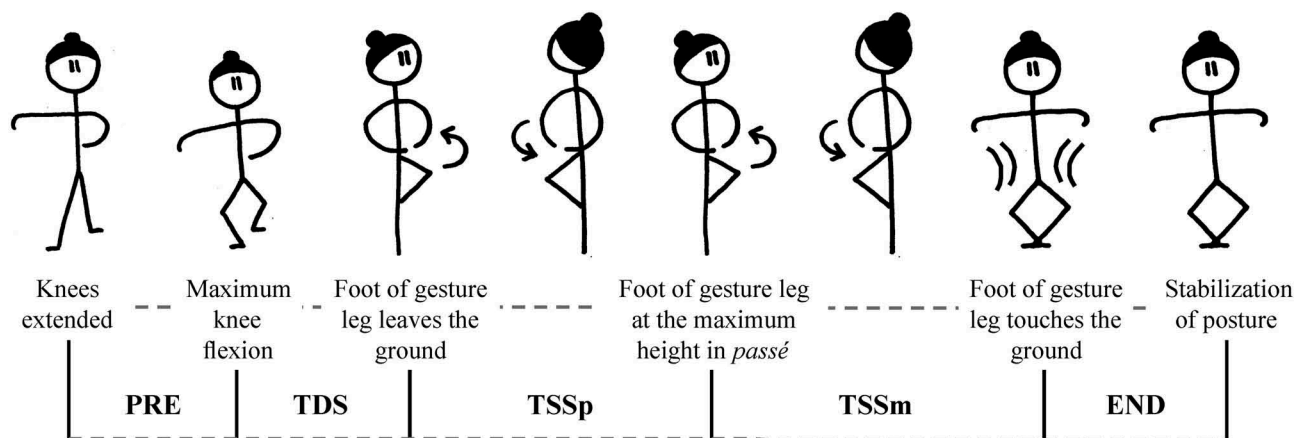


Figure 1. Phases of the *en dehors* double pirouette.

PRE = preparatory; TDS = turning with double-leg support; TSSp = turning with single-leg support in preswing; TSSm = turning with single-leg support in midswing; END = ending.

trunk. One volunteer performed three EDDP trials while data were collected simultaneously from both systems.

Data reduction

Cross-correlations between time-series derived from the inertial sensors and from Qualisys for the two kinematic variables analyzed in this study ($ANGLE_{TR}$, and $ANGLE_{RA}$, described below) were 0.96 and 0.92. The high cross-correlation values show the consistency between the two systems.

The main issue with inertial sensors in smartphones is their variable acquisition rate, due to requirements of the smartphone's operational system. Therefore, we resampled the data, given all the data recorded, in order to achieve a common sampling frequency of 30 Hz as a minimum rate. We decided on this minimum rate because the spectral density (Welch estimation method) of data from all 14 Qualisys markers showed no relevant power above 15 Hz, indicating that data collection at 30 Hz with the smartphone inertial sensors would be adequate. Digital video was recorded at a rate of 60 Hz. The procedure to synchronize the video with the inertial sensors was to knock twice on the smartphone with a finger at the start of the recording. These knocks were captured as distinct sound signals in the video and clear acceleration peaks in the sensor data with corresponding time stamps. Thus, the timestamps of both the video and smartphone could be synchronized.

Based on visual analysis of video data, the EDDP was divided into five phases: (1) preparatory (PRE), (2) turning with double-leg support (TDS), (3) turning with single-leg support in preswing (TSSp), (4) turning with single-leg support in midswing (TSSm), (5) ending phase (END) (Lin et al., 2014). See Figure 1 for the criteria used to define phases. An author with 17 years of ballet training was responsible for determining the initial and final time points for all phases of all trials.

Data were exported from the smartphone to a personal computer with Sensor Kinetics Pro. Custom made Matlab (MathWorks Inc.) routines were used to make acquisition rates uniform, filter the resampled data and then calculate angles. The linear acceleration sensor was used to identify synchronization timestamps. The time series were used to calculate, for all movement phases, the instantaneous rotation axis of the trunk (Sommer, 1992) and obtain its inclination angle ($ANGLE_{RA}$) with respect to the global vertical. The trunk inclination angle with respect to the global vertical ($ANGLE_{TR}$) was also calculated. Previous evidence indicates that these variables capture significant differences in *pirouette* performance between dancers (Lin et al., 2014, 2013).

The magnetometer was used to mark a three-dimensional vector whose variation from an initial position is taken as a rotation. The rotation was then projected to the vertical axis to calculate the smartphone angle with respect to the vertical as $ANGLE_{TR}$. The variation of calculated three-dimensional rotation vector from one instant to another is taken as an estimate of the rotation vector. The projection of this angle along the vertical is the angle of the rotation vector ($ANGLE_{RA}$). The average and root-mean-square (RMS) of both angles over time were obtained for each movement phase. The values were averaged across the 3 EDDP performed on each focus condition, and these averages were used for statistical analysis.

For analysis of movement quality, three experts in classical ballet rated all 360 EDDP videos without any information on attentional focus condition. Ordering of the videos was randomized. For each EDDP the expert rated the dancer's performance on (1) keeping balance during the turn, (2) smoothness to finish the movement, (3) appropriate closing fifth position, (4) keeping heel high during the turn (*demi-pointe*), and (5) esthetics and beauty. Each rating criterion was scored from 0 to 20, totaling a possible score of 0 to 100 for each EDDP trial. The average scores given by the experts in each focus condition were used for statistical analysis.

Statistical analysis

All variables were analyzed using the Statistical Package for the Social Sciences Version 21.0 (SPSS for Windows, Chicago, IL). Differences in the ability to adhere to focus instructions were tested with a mixed ANOVA with factors Category (2: professional or amateur) and Focus (2: IF and IEF). Agreement between raters was tested with the Intraclass Correlation Coefficient (ICC). Differences in movement quality were tested with a mixed ANOVA with factors Category (2: professional or amateur) and Focus (3: NF, IF and IEF). Differences in the average and RMS of $ANGLE_{RA}$ and $ANGLE_{TR}$ were tested with a mixed ANOVA with factors Category (2: professional or amateur), Focus (3: NF, IF or IEF), and EDDP Phase (5: PRE, TDS, TSSp, TSSm or END). Post hoc Bonferroni-corrected *t* tests were used to investigate pairwise differences. Statistical significance was set at $p < .05$.

Results

Ability to adhere to focus instructions

Professionals reported better ability to adhere to both focus instructions (IF = 8.65 ± 1.37 , IEF = 8.18 ± 1.75)

compared to amateurs ($IF = 7.57 \pm 1.60$, $IEF = 6.87 \pm 1.91$), $F(1, 38) = 6.420$, $p = .016$. Overall, both categories reported greater ability to adhere to IF (8.03 ± 1.58) than IEF instructions (7.43 ± 1.93), $F(1, 38) = 5.098$, $p = .030$. There was no interaction between Category and Focus ($p = .666$).

Movement quality

The average measure ICC was .856 with a 95% confidence interval from .828 to .880, $F(359, 718) = 6.946$, $p < .001$. Expert ratings showed superior performance for professionals (70.57 ± 10.92) compared to amateurs (50.95 ± 18.31), $F(1, 38) = 15.724$, $p < .001$. There was no main effect of Focus ($p = .925$) and no Focus x Category interaction ($p = .697$) on movement quality.

Average and root-mean-square of $ANGLE_{RA}$ and $ANGLE_{TR}$

For the average and RMS of $ANGLE_{RA}$ over time, Category was not significant ($p > .186$). Although the effect of focus was significant for $ANGLE_{RA}$ RMS, (NF = 10.76 ± 2.62 , IF = 10.97 ± 2.61 , IEF = 10.94 ± 2.62), $F(2, 76) = 4.276$, $p = .033$, contrasts showed no differences between the three focus conditions (NF x IF, $p = .088$; NF x IEF, $p = .136$; IF x IEF, $p = 1.000$). There were no interaction effects for Category x Focus ($p > .066$), Category x Phase ($p > .106$) or Focus x Phase ($p > .724$).

Table 1. $ANGLE_{TR}$ averages for each phase and category.

Phase	Category		
	Professionals	Amateurs	
PRE	8,34 \pm 2,43	10,02 \pm 1,45	$p = .019^*$
TDS	8,59 \pm 2,55	9,92 \pm 1,42	$p = .068$
TSSp	6,72 \pm 1,49	5,78 \pm 1,24	$p = .916$
TSSm	4,98 \pm 1,72	4,35 \pm 1,68	$p = .206$
END	7,16 \pm 1,79	8,50 \pm 1,57	$p = .016^*$

* Statistically significant difference between groups ($p < .05$).

The main effect of Category was significant for $ANGLE_{TR}$ average (professionals = 6.85 ± 2.57 , amateurs = 7.85 ± 2.56), $F(1, 38) = 4.792$, $p = .035$, and RMS (professionals = 7.06 ± 2.47 , amateurs = 8.17 ± 2.33), $F(1, 38) = 5.648$, $p = .023$. The Category x Phase interaction was also significant for $ANGLE_{TR}$ average ($p = .024$), with post hoc Bonferroni-corrected independent t tests showing significantly lower values for professionals compared to amateurs in the PRE and END phases (Table 1). Neither the main effect of Focus ($p > .426$) nor any interaction effect involving Focus were significant ($p > .196$).

$ANGLE_{RA}$ and $ANGLE_{TR}$ averages across Categories and Focus conditions for each movement Phase are available in Figures 2 and 3.

Discussion

This study investigated the effects of attentional focus instructions on ballet dancers' performance of a form-based skill, the EDDP. In general, professionals were

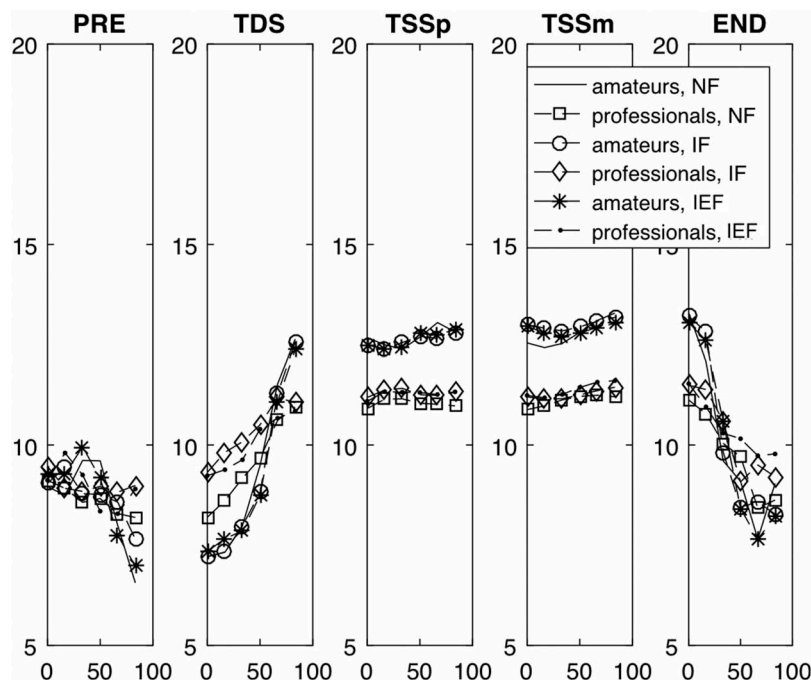


Figure 2. $ANGLE_{RA}$ average time series for each phase, category, and focus. Please see the text for the definition of phases.

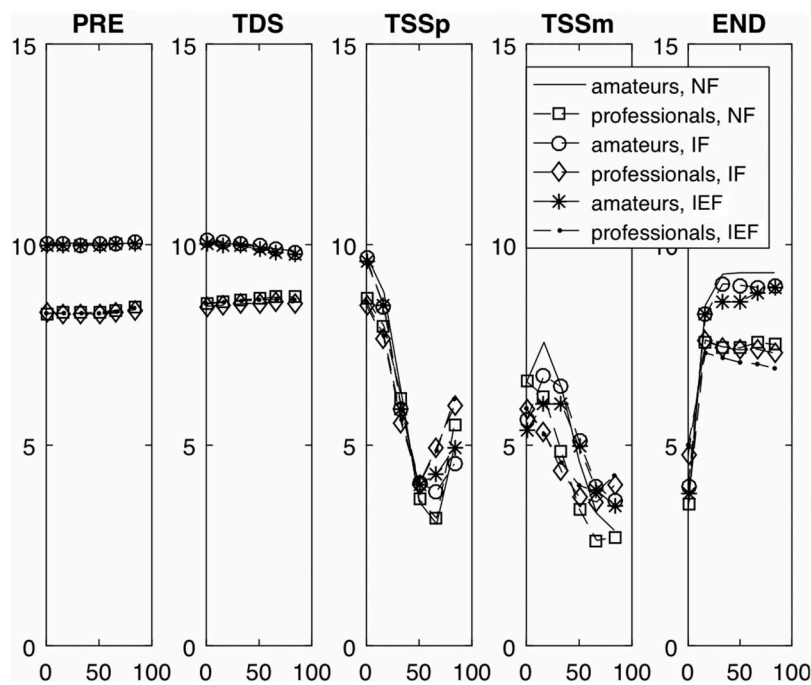


Figure 3. $ANGLE_{TR}$ average time series for each phase, category, and focus. Please see text for the definition of phases.

better able to control their attentional focus compared to amateurs. Also, both professionals and amateurs were better able to adhere to internal focus instructions than to imagined external focus instructions. This result is consistent with the evidence that ballet dancers, either spontaneously or as a result of commonly used instructions, typically focus on body movements (Guss-West & Wulf, 2016). Given the preference of ballet dancers for internally focused attention, this study set out to investigate whether EDDP performance could be improved quantitatively and qualitatively by using instructions to turn attention away from the body.

Qualitative aspects of performance were investigated. Technical and esthetic quality for each EDDP was determined by blinded expert ratings. Agreement between raters was high, showing that the quality checklist devised for this study was applied reliably by the experts. Performance of professionals was clearly superior to amateurs, as expected. These results confirm that the method to assess movement quality was sensitive to relevant performance differences. It is important to note that neither group had neither flooring nor ceiling effects on movement quality ratings. The average ratings varied from 50 to 70 out of 100, showing that there was room either for performance improvement or for deterioration as a result of the attentional focus manipulations.

Based on the often-replicated results reported in previous literature (Wulf, 2013), we expected that the

imagetic instructions eliciting a focus outside the body (functionally similar to an external focus, Guss-West & Wulf, 2016; Wulf, 2007) would lead to increased movement quality ratings compared to the IF or NF conditions, for both professionals and amateurs. Contrary to expectations, there were no effects of attentional focus on overall performance as rated by ballet experts. Attentional focus instructions, as delivered in this study, have no immediate effects on the technical and esthetic performance of the EDDP.

Quantitative aspects of EDDP performance were investigated with the kinematic variables $ANGLE_{RA}$ and $ANGLE_{TR}$. While $ANGLE_{RA}$ did not differ between professionals and amateurs, $ANGLE_{TR}$ showed consistent differences between the two categories, confirming that a lower trunk inclination angle is characteristic of improved EDDP performance. Based on the attentional focus literature, we expected that the IEF would improve kinematic aspects of movement (Lohse, Sherwood, & Healy, 2010; Munzert, Maurer, & Reiser, 2014). However, there were no significant differences between the three focus conditions for any of the kinematic variables.

This study included 40 ballet dancers, a control condition, a balanced order of presentation of IF and IEF instructions, reliable movement quality assessments, and kinematic variables previously shown to be sensitive to performance differences between ballet dancers (Lin et al., 2014, 2013). The qualitative assessments and the kinematic measures of $ANGLE_{TR}$ proved to be

sensitive to detect performance differences in the EDDP. With regards to the effects of focus, quantitative and qualitative results were consistent with each other. Together they show that an IEF is not beneficial and IF is not disruptive for EDDP performance of amateur and professional ballet dancers.

The greater majority of findings in the literature indicate superior movement performance under instructions to focus attention externally to the body. A 2013 review (Wulf, 2013) concluded: “The enhancements in motor performance and learning through the adoption of an external relative to an internal focus of attention are now well established. The breadth of this effect is reflected in its generalizability to different skills, levels of expertise, and populations, as well as its impact on both the effectiveness and efficiency of performance” (p. 99). Wulf (2007, 2013) claims that instructions relating to body movements will always prove deleterious to motor learning and performance. However, the results of this study are inconsistent with this claim.

This study is in line with other studies that have found no disadvantages of an internal focus or no benefits of an external focus for tasks that do not involve implements. Worms, Sins, van Wegen, Loram, and Beek (2017) found no effects of focus on walking stability (tested on a split-belt treadmill) in older adults. A focus on how the body feels during exercise did not disrupt movement efficiency in running (Schucker, Knopf, Strauss, & Hagemann, 2014). No benefits of an external focus were found for top-level acrobats (Cirque du Soleil) balancing on an unstable surface. For the acrobats, both the internal and external focus conditions led to worse performance than the control condition (no focus instructions) (Wulf, 2008). The same pattern of results were found for highly trained athletes, who ran faster in a control condition compared to internal and external conditions (Porter & Sims, 2013). Also, no differences were found between internal and external foci for elderly individuals learning a dynamic balance task (De Bruin, Swanenburg, Betchon, & Murer, 2009). An experiment on the learning of a gymnastics floor routine showed that internal and external foci do not affect performance (Lawrence et al., 2011).

This set of results, however, is inconsistent with those of Abdollahipour et al. (2015), who showed superior performance (gymnastics jump height and judged movement form score) with an external compared to internal or control foci. It has been argued that their choice of focus instructions for the gymnasts resulted in an unfair comparison (Collins et al., 2016), because the internal focus was irrelevant to the task

(direction in which the hands were pointing after the half turn). The external instruction, by contrast, was a clear outcome focus that directly facilitated performance (the direction in which the tape marker on the chest was pointing after the half turn). In the present study we took care to choose equally relevant, performance-related instructions, so that any differences between experimental conditions would be due to focus, and not to relevance. With this precaution, we found no differences between IEF and IF.

Different explanations can be put forth for our findings. They relate to two fundamental interacting aspects of skill: the nature of the task and personal factors. First, the *pirouette* is a form-based task where movement technique is the primary determinant of performance. Dancers need refined intersegmental coordination and accuracy to achieve the desired esthetics (Laws, 2002). The goal of the *pirouette* is to produce a specific movement pattern; thus, the movement itself is the task outcome. In this case, a focus on the movement itself may be not detrimental (Peh et al., 2011). While in tasks with implements an internal focus can direct performers' attention to sensory information that is at best tangential to goal achievement (e.g., the mechanics of muscle and joint activity) (Wrisberg, 2007), in form based-tasks like the *pirouette* an internal focus can be an “outcome-creating focus”, that is, a focus on goal achievement (Collins et al., 2016).

Second, personal factors, like familiarity with a given focus, might also be determinant of its effects. Maurer and Munzert (2013) reported that free-throw performance of skilled basketball players was superior under familiar compared with unfamiliar focus conditions, irrespective of focus being internal or external. Traditions in dance teaching practice probably influence personal preferences. Ballet training is replete with instructions that emphasize internally, body-focused attention. A survey of professional dancers showed that most spontaneously adopted internal foci, or combinations of internal and external foci (Guss-West & Wulf, 2016). This is consistent with our participants reporting higher adherence to internal focus instructions. Habitual use of internal foci has also been documented for athletes. For example, runners performing at high intensity use a predominantly internal focus of attention (Hutchinson & Tenenbaum, 2007). Golfers move back-and-forth between focussing on bodily movements and the effects of their actions (Bernier, Codron, Thienot, & Fournier, 2011). It's apparent that high-level performers are used to consciously and deliberately attempting to refine their movement technique (Toner, Montero & Moran, 2015a), so it's reasonable to expect that frequently used familiar focus

strategies become integrated into the skill and are no longer disruptive to execution (Wulf, 2008). This could explain why internally focused attention was not detrimental to performance in our study. It also opens up the possibility that internal and external focus instructions could have differential effects for novice dancers who haven't yet established focus habits.

It is also possible that images, as used in this study, might be less effective than an explicit focus on externally attached objects (as used in Abdollahipour et al., 2015). The effectiveness of imagetic instructions will depend on an individual's understanding of their meaning and his or her motor imagery ability. Although imagetic instructions may be common in ballet classes, our participants were, in general, less able to adhere to the image-based IEF than to IF. This could explain why directing attention away from the body to the movement effects was not beneficial to performance in our study.

Finally, another interpretation for our results concerns the theoretical basis of the focus effects, which have been explained with the "constrained action hypothesis". According to this hypothesis, an internal focus would induce conscious control that interferes with *automatic* coordination processes (Wulf et al., 2001). But do *pirouettes* ever become "automatic" or "proceduralized" for ballet dancers? Research in motor learning and performance has been greatly influenced by the information-processing framework of skill acquisition (e.g., Fitts & Posner, 1967). In this framework, skill learning starts with slow, deliberate, effortful controlled processing. Gradually, learners progress through the associative stage to the highest stage of skill execution, the *autonomous* or *procedural stage*, in which performance is uncontrolled, unconscious, efficient and fast. In this view, there are distinct, independent stages that a performer goes through as practice accumulates: (1) cognitive, (2) associative, (3) automatic. Supposedly, our participants, who were experienced amateurs and professionals, would be at the automatic stage of skill with regards to *pirouettes*. An internal focus would impair their performance, leading them to consciously control automatic or proceduralized movements, reverting them back to a step-by-step execution mode (Wulf, 2007).

However, there is now ample literature with alternative interpretations of expert performance, stemming especially from phenomenological accounts (Toner, et al., 2015a, 2015b). Evidence shows that expert performers may continuously cycle back and forth between performance stages (Gray, 2004) and indeed *actively seek to counteract automaticity* to avoid "arrested development" of their performance, when they engage in

deliberate practice (Ericsson, Krampe, & Tesch-Römer, 1993). Skilled performance requires the dynamic interplay of automatic and conscious processing in order to avoid errors and to meet varying contextual demands. Our dancers, therefore, may not have suffered negative effects of an internal focus because their performance is not really adequately characterized as *automatic*.

Our results show that the practical wisdom and tradition followed by teachers does not hurt the performance of the *pirouette* for amateurs and professional dancers, as would be inferred from the attentional focus literature. It is important to point out; however, that these results may not apply to learning situations. An external attentional focus might be relevant to expedite a beginner's learning of form-based skills. It makes intuitive sense that a focus on the effects of movement would help planning it (Prinz, 1997). Analogies such as the one used in this study are a good source of information for the general expected movement effect. Thus, beginners might benefit from practicing with form analogies inducing an outcome focus more than by focusing internally on the segmental motion. There appears to be, however, no differential effects of IF and IEF on the immediate performance of the *en dehors* double *pirouette* for amateur and professional ballet dancers.

This study might have been limited by the use of sensors on a single instead of multiple trunk landmarks. This may have reduced the sensitivity to detect effects of attentional focus conditions at the kinematic level. Any kinematic effects that may have been present, however, were not sufficient to produce esthetic effects according to the ratings from ballet experts. Possible imperceptible movement differences may be relevant; however, for the specific prevention of musculoskeletal injuries, and should be investigated further. Lastly, it is possible that using a classically explicit external focus directed to a physical object (that may be attached to the body) instead of image-based instructions may positively affect performance.

What does this article add?

Important questions for ballet dancers and teachers regarding how performance can be enhanced. There is extensive evidence that focusing away from the body on the intended movement effect can result in superior performance relative to an internal focus on body movements, but research on the effects of attentional focus instructions for form-based activities such as ballet is scarce. The objective of this study was to investigate whether *pirouette* performance of professional and amateur ballet dancers can be improved quantitatively and qualitatively by instructions created to induce an

imagined external focus, compared to an internal focus of attention or no focus instructions. Contrary to expectations, a imagined external focus produced no improvement on movement quality as rated by ballet experts blinded to experimental conditions. Kinematics measures of trunk verticality were also not improved. Additionally, no detrimental effects of an internal focus were found. Possibly, the benefits of using an imagined external focus of attention are limited by personal preferences and the nature of form-based tasks, when movement form itself is the most valuable outcome. External focus instructions, however, may be beneficial to expedite the initial learning phases of form-based tasks, for they provide information about the general expected movement effect. Future studies should investigate this possibility.

Funding

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) – [Finance Code 001].

ORCID

Carolina M. Andrade  <http://orcid.org/0000-0002-6756-6071>

Thales R. de Souza  <http://orcid.org/0000-0001-8081-1687>

Alysson F. Mazoni  <http://orcid.org/0000-0001-5265-6894>

André G. P. de Andrade  <http://orcid.org/0000-0003-3406-4558>

Daniela V. Vaz  <http://orcid.org/0000-0003-0470-6361>

References

- Abdollahipour, R., Wulf, G., Psotta, R., & Nieto, M. P. (2015). Performance of gymnastics skill benefits from an external focus of attention. *Journal of Sports Sciences*, 33(17), 1–7. doi:10.1080/02640414.2015.1012102
- Bernier, M., Codron, R., Thienot, E., & Fournier, J. F. (2011). The attentional focus of expert golfers in training and competition: A naturalistic investigation. *Journal of Applied Sport Psychology*, 23(3), 326–341. doi:10.1080/10413200.2011.561518
- Collins, D., Carson, H. J., & Toner, J. (2016). Letter to the editor concerning the article “performance of gymnastics skill benefits from an external focus of attention” by Abdollahipour, Wulf, Psotta, & Nieto (2015). *Journal of Sports Sciences*, 34(13), 1–5. doi:10.1080/02640414.2015.1098782
- Davids, K. (1988). Ecological validity in understanding sport performance: Some problems of definition. *Quest*, 40(2), 126–136. doi:10.1080/00336297.1988.10483894
- De Bruin, E. D., Swanenburg, J., Betchon, E., & Murer, K. (2009). A randomised controlled trial investigating motor skill training as a function of attentional focus in old age. *BioMed Central Geriatrics*, 9(1), 1–10. doi:10.1186/1471-2318-9-1
- Ericsson, K. A., Krampe, R. T., & Tesch-Römer, C. (1993). The role of deliberate practice in the acquisition of expert performance. *Psychological Review*, 100(3), 363–406. doi:10.1037/0033-295X.100.3.363
- Fitts, P. M., & Posner, M. I. (1967). *Human performance*. Belmont, CA: Brooks/Cole Publishing Company.
- Galán-Mercant, A., & Cuesta-Vargas, A. I. (2013). Differences in trunk kinematic between frail and nonfrail elderly persons during turn transition based on a smartphone inertial sensor. *BioMed Research International*, 1(2), 1–6. doi:10.1155/2013/279197
- Gray, R. (2004). Attending to the execution of a complex sensorimotor skill: Expertise differences, choking, and slumps. *Journal of Experimental Psychology: Applied*, 10(1), 42–54. doi:10.1037/1076-898X.10.1.42
- Guss-West, C., & Wulf, G. (2016). Attentional focus in classical ballet: A survey of professional dancers. *Journal of Dance Medicine & Science*, 20(1), 23–29. doi:10.12678/1089-313X.20.1.23
- Hutchinson, J. C., & Tenenbaum, G. (2007). Attention focus during physical effort: The mediating role of task intensity. *Psychology of Sport and Exercise*, 8(2), 233–245. doi:10.1016/j.psychsport.2006.03.006
- Lawrence, G. P., Gottwald, V. M., Hardy, J., & Khan, M. A. (2011). Internal and external focus of attention in a novice form sport. *Research Quarterly for Exercise and Sport*, 82(3), 431–441. doi:10.1080/02701367.2011.10599775
- Laws, K. (2002). *Physics and the art of dance: Understanding movement*. New York, NY: Oxford University Press.
- Lin, C.-W., Chen, S.-J., Su, F.-C., Wu, H.-W., & Lin, C.-F. (2014). Differences of ballet turns (pirouette) performance between experienced and novice ballet dancers. *Research Quarterly for Exercise and Sport*, 85(3), 330–340. doi:10.1080/02701367.2014.930088
- Lin, C.-W., Su, F.-C., Wu, H.-W., & Lin, C.-F. (2013). Effects of leg dominance on performance of ballet turns (pirouettes) by experienced and novice dancers. *Journal of Sports Sciences*, 31(16), 1781–1788. doi:10.1080/02640414.2013.803585
- Lohse, K. R., Sherwood, D. E., & Healy, A. F. (2010). How changing the focus of attention affects performance, kinematics, and electromyography in dart throwing. *Human Movement Science*, 29(4), 542–555. doi:10.1016/j.humov.2010.05.001
- Maurer, H., & Munzert, J. (2013). Influence of attentional focus on skilled motor performance: Performance decrement under unfamiliar focus conditions. *Human Movement Science*, 32(4), 730–740. doi:10.1016/j.humov.2013.02.001
- Munzert, J., Maurer, H., & Reiser, M. (2014). Verbal-motor attention-focusing instructions influence kinematics and performance on a golf-putting task. *Journal of Motor Behavior*, 46(5), 309–318. doi:10.1080/00222895.2014.912197
- Nishiguchi, S., Yamada, M., Nagai, K., Mori, S., Kajiwara, Y., Sonoda, T., ... Aoyama, T. (2012). Reliability and validity of gait analysis by android-based smartphone. *Telemedicine and e-Health*, 18(4), 292–296. doi:10.1089/tmj.2011.0132
- Peh, S. Y.-C., Chow, J. Y., & Davids, K. (2011). Focus of attention and its impact on movement behaviour. *Journal of Science and Medicine Sport*, 14(1), 70–78. doi:10.1016/j.jsams.2010.07.002
- Porter, J. M., & Sims, B. (2013). Altering focus of attention influences elite athletes sprinting performance. *International Journal of Coaching Science*, 7(2), 41–51.

- Prinz, W. (1997). Perception and action planning. *European Journal of Cognitive Psychology*, 9(2), 129–154. doi:10.1080/713752551
- Schucker, L., Knopf, C., Strauss, B., & Hagemann, N. (2014). An internal focus of attention is not always as bad as its reputation: How specific aspects of internally focused attention do not hinder running efficiency. *Journal of Sport & Exercise Psychology*, 36(3), 233–243. doi:10.1123/jsep.2013-0200
- Sommer, H. J. (1992). Determination of first and second order instant screw parameters from landmark trajectories. *Journal of Mechanical Design*, 114(2), 274–282. doi:10.1115/1.2916943
- Toner, J., Montero, B. G., & Moran, A. (2015a). Considering the role of cognitive control in expert performance. *Phenomenology and the Cognitive Sciences*, 14(4), 1127–1144. doi:10.1007/s11097-014-9407-6
- Toner, J., Montero, B. G., & Moran, A. (2015b). The perils of automaticity. *Review of General Psychology*, 19(4), 431–442. doi:10.1037/gpr0000054
- Worms, J. L. A. M., Sins, J. F., van Wegen, E. E. H., Loram, I. D., & Beek, P. J. (2017). Influence of focus of attention, reinvestment and fall history on elderly gait stability. *Physiological Reports*, 5(1), 1–9.
- Wrisberg, C. A. (2007). An applied sport psychological perspective on the relative merits of an external and internal focus of attention. *E-Journal Bewegung und Training*, 1, 53–54.
- Wu, G., van der Helm, F. C. T., Veeger, H. E. J., Makhsous, M., Roy, P. V., Anglin, C., ... Buchholz, B. (2005). 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*, 38, 981–992. doi:10.1016/j.jbiomech.2004.05.042
- Wulf, G. (2007). *Attention and motor skill learning*. Las Vegas, NV: Human Kinetics.
- Wulf, G. (2008). Attentional focus effects in balance acrobats. *Research Quarterly for Exercise and Sport*, 79(3), 319–325. doi:10.1080/02701367.2008.10599495
- Wulf, G. (2013). Attentional focus and motor learning: A review of 15 years. *International Review of Sport and Exercise Psychology*, 6(1), 77–104. doi:10.1080/1750984X.2012.723728
- Wulf, G., McNevin, N., & Shea, C. (2001). The automaticity of complex motor skill learning as a function of attentional focus. *The Quarterly Journal of Experimental Psychology*, 54A(4), 1143–1154. doi:10.1080/713756012
- Wulf, G., Töllner, T., & Shea, C. H. (2007). Attentional focus effects as a function of task difficulty. *Research Quarterly for Exercise and Sport*, 78(3), 257–264. doi:10.1080/02701367.2007.10599423

Improving the accuracy of a low-cost multifocal stereo microscope through microstepping and dynamic modelling

Alysson Fernandes Mazoni, Carolina Zabini, Jorge Stolfi and Joao Mauricio Rosario

Abstract—In this work we consider the acquisition of three-dimensional models of millimeter-scale objects using a low-cost equipment consisting of a microscope with automated focus plane adjustment and variable lighting. We show how to improve the accuracy of the resulting models by a combination of microstepping, feedback control, and dynamic system modeling. Tests show errors of the computed depth dimension could be routinely kept below ?? mm.

I. INTRODUCTION

THE study of microfossils in paleontology, especially for taxonomy and taphonomy, has greatly benefited from the capture and analysis of three-dimensional digital models. Morphometric data and taphonomic information are more easily extracted from these models than from the object directly [1], [2].

The application that motivated this work is the use of fossils of small marine bivalve brachiopods (from about 1 millimeter to a few centimetres across) for the interpretation of age and paleoenvironmental conditions previous and during the fossilization process. This is done by studying the fossils from various sedimentary layers and their relative frequencies. The taxonomic classification depends on very small and/or subtle details, such as the convexity of the shell and the number and shape of ridges on them (external ornamentation). To achieve this goal, we needed 3D models with relatively high accuracy, especially along the depth axis (perpendicular to the image plane).

Several papers have been published on the acquisition of 3D models of fossils [2]–[7] and outcrops [8], [9]. The best method for each application depends on details such as the size of the fossil specimens, how well they are differentiated from the matrix, and the resolution needed for the analysis. Several types of microscopes are commercially available for this purpose, mostly using structured light or laser ranging [10], [11]. These instruments have high resolution and accuracy, but are very expensive and depend on proprietary closed-source software. Do-it-yourself (DIY) devices, controlled by open software, can be built at much lower cost, and several designs are available [16–21].

A. F. Mazoni and J. M. Rosario are with the Faculty of Mechanical Engineering, University of Campinas, Campinas, Sao Paulo, Brazil.

J. Stolfi is with the Institute of Computing, University of Campinas, Sao Paulo, Brazil.

Carolina Zabini is with the Institute of Geosciences, University of Campinas, Campinas, Sao Paulo, Brazil.

In this paper we describe such a 3D capture system, that we designed and built for the above application. The system consists of a motorized microscope and ancillary control electronics and software. The device is assembled from simple 3D-printed parts and inexpensive commercial components, including a low-cost digital microscope camera, smooth and threaded steel rods, light-emitting diodes (LEDs), cables, and stepper motors. The software, created by the authors, is available under the free open source model. It automatically acquires a stack of 2D images of each specimen, varying the position of the focus plane and (optionally) the lighting conditions. The captured images can then be combined by multifocus stereo software [?]?? to yield the desired 3D model.

The original contributions of this paper include the use of a rotational position encoder and a microstepping feedback loop, based on a linear dynamic model of the system, to achieve high positioning accuracy of the camera. With this device we have obtained fossil models with better than 0.?? mm accuracy along the depth axis. The multiple programmable light sources in our device afford better image quality, and could be exploited by photometric stereo algorithms to obtain even higher accuracy ??.

A. Principles of 3D model capture

The main approach we use to acquire the 3D models is called *multifocal stereo*. It exploits the fact that a microscope can produce a sharp-focused image only for objects that are located in a very narrow range of distances from the plane of the objective lens. Objects that are before or beyond that range appear blurred on the image. This phenomenon is a serious flaw for single-image capture, but by comparing the local sharpness of several images taken with varying object-lens distance one can determine the depth (Z coordinate) of each point on the visible surface of the object [12].

Another approach that could be used with our device is *photometric stereo*. This method uses multiple photos of the object, all taken from the same direction and distance but with the object illuminated by light from different directions. By comparing the apparent brightness of each pixel in the different images, one can determine the slope of the object's surface at that pixel—that is, the gradient of the Z coordinate as a function of the X and Y image coordinates. This data then can be integrated to obtain the Z coordinate [13]–[16].

Figure 1(top) shows a single normal-view image of one of the fossils that are the object of our research, still embedded

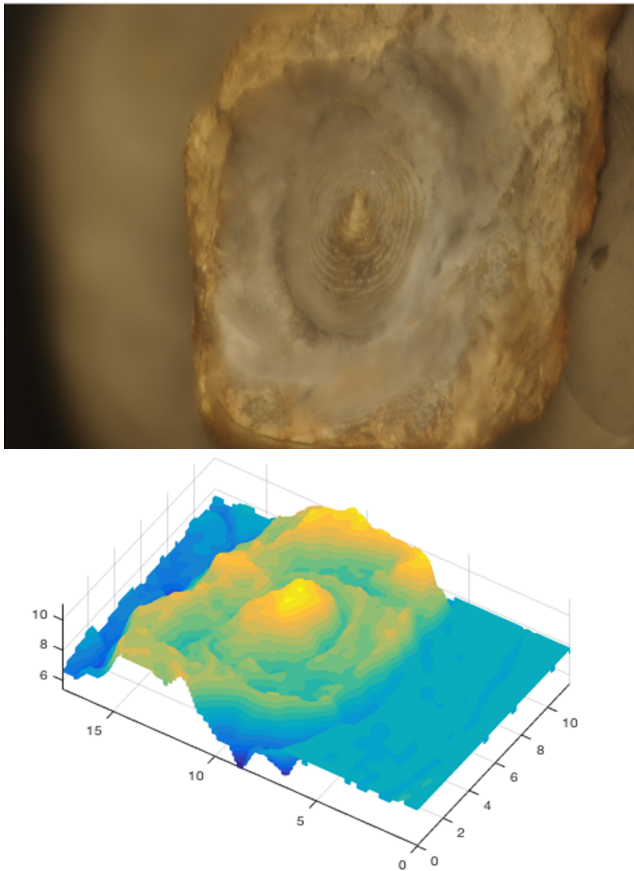


Fig. 1. Example fossil and the extracted 3D model.

in the sediment matrix. Figure 1(bottom) is a slant view of the 3D model of the same fossil, computed by the Surströmming?? multifocus stereo software [?] from 100?? images taken with camera displacement step of ?? mm. The lightness is proportional to the reconstructed Z coordinate.

B. Automating the image capture process

Multifocal stereo typically requires dozens of different camera positions, each with six or more lighting conditions; potentially, a couple hundred shots for a single object. Obviously, these 3D capture methods are practical only if the positioning and light switching are automated. In the least few years, designs for several DIY devices for this purpose have been published [12], [17]–[21].

The basic design of these devices (and ours) is the same. The microscope camera is mounted on a moving “carriage” that slides up and down on guide rods. A lead screw, driven by a stepper motor through a coupling torsion spring, is used to displace the carriage by uniform increments in the direction of the optical (Z) axis. With this design, the camera’s position is in principle known at all times, simply by keeping track of the number of motor steps.

A simple microcontroller (MC), such as the Arduino [?], should be used to take care of the low-level control of the stepper motor and of the light sources, by issuing the proper electrical pulses in real-time. A full computer (such as a PC, a

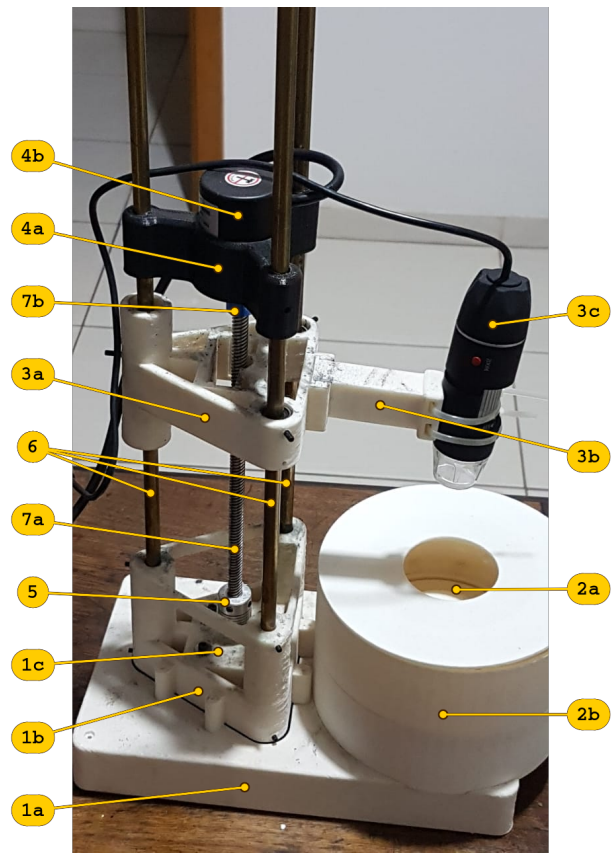


Fig. 2. View of the system. The parts are described in the text.

laptop, or a Raspberry Pi [?]) issues high-level commands to the MC (such as “raise the camera by 0.05 mm” or “turn light #3 on”) and, once the MC signals back that the operation is complete and the camera is again static, takes a snapshot and saves it on its storage.

II. METHODOLOGY

A. Physical components

Figure 1 shows our assembled device. The design is closely based on that of Lelis et al. [12]. The main physical parts are: (1) base, (2) lighting dome, (3) carriage, (4) top frame, (5) coupling spring, (6) guide rods, and (7) lead screw.

Parts (1-4) were fabricated in poly(lactic acid) plastic (PLA)?? with a 3D filament?? printer. Due to printer size limitations, the base was printed as two separate pieces (1a,1b) that were screwed?? together. The lower part (1a) is a hollow box, open at the bottom, with walls ?? mm thick. The upper part (1b), which includes the stepper motor housing (1c), is filled with a honeycomb lattice, with outer walls ?? mm thick.

The carriage (3) too is fabricated as two separate parts. The body of the carriage (3a) has four linear bearings, held in place by set screws, that slide on the three guide rods. These bearings provide 4 non-coplanar sliding constraints that prevent rotation or horizontal translation of the carriage relative to the guide rails.

The other part of the carriage (3b) is the camera holder. A trapezoidal flange at one end fits into a slot on the body of

the carriage, and is held there by a set screw. The camera is tightly secured to the other end of the holder with nylon zip-ties. This two-part design allows the use of multiple cameras, with different diameters and shapes, without the need to disassemble the stand or replace the carriage body.

The top frame (4) consists of a plastic frame (4a) attached to the three rails by three set screws, ensuring their parallelism and rigidity. It supports an angular position encoder (4b)

The camera (3c) used for this article is a Celestron?? hand-held microscope with variable zoom and a maximum resolution of ?? pixels per millimeter on the focus plane, at maximum magnification.

The three guide rails (6) are 600?? mm long, made of solid polished round steel rods with 10 mm diameter. They are inserted into holes of the base and held there by three set screws.

The lead screw (7a) is a commercial product sold online for DIY 3D printers and similar projects, with ?? mm diameter and ?? mm step per turn. The brass?? driving nut is attached to the bottom of the carriage. The top end of the lead screw is inserted into a 3D-printed PLA adaptor (7b) that connects the screw to the axis of the encoder. The screw is connected to the motor's axis by an aluminum coupling spring (5).

The lighting dome (2) consists of two nested PLA shells. The inner shell (2a) is a hemisphere with an opening at the top for the camera, and 16?? holes in various positions, that hold the light sources. Each light source is a white LED (with ?? V and ?? mA at full intensity). These light sources are wired with a common ground and a separate positive?? terminal wire for each LED. The inner shell is held in place three?? plastic pegs that plug into holes on the base.

The outer shell (2b) of the lighting dome is just a hollow cylindrical cover that fits over the inner shell, with an opening at the top for the camera. Its only purpose is to hide and protect the wiring of the LEDs.

B. Motor and electronics

The electronics comprises a circuit board based on the Arduino microcontroller, a power source unit, an angular position encoder, and a power motor driver unit.

The lead screw is driven by a NEMA 17 stepper motor with 200?? full steps per turn (1.8 degrees per full step). It is operated by a Pololu A4988 power driver unit [22].

The encoder provides to the Arduino ??a number between 0 and 399, that is the angular position θ of the lead screw, measured from some indeterminate reference position, in units of 1/400 of a full turn (0.9 degrees). Thus, the relative rotation between two readings θ_0 and θ_1 from the encoder can be taken to be $\Delta\theta = (\theta_1 - \theta_0) \bmod 400 - 200$, as long as the rotation is less than one half turn in either direction.

The real-time microstepping signals are provided by an Arduino Uno?? microcontroller. The electronics (Figure 2) are assembled on a printed circuit board??. The MC communicates with the master computer (a laptop, PC, or equivalent) via a USB cable.

The Arduino also operates a 16??-output multiplexer?? chip that controls the LEDs. This chip receives 16?? bits through

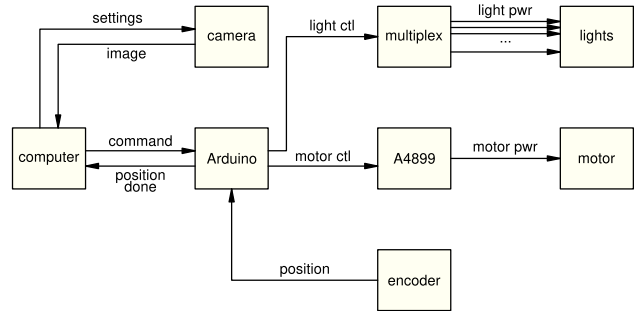


Fig. 3. Logical block diagram of the electronics and software.

a serial input pin, and sets the 16?? output to 0 V or ?? V according to the respective bit.

C. Software

Figure 3 is the high-level logical diagram of the system. The software consists of a *firmware* `miguec-mc.ino??`, that runs in the MC, and a *master program* `miguec-pc.py??` that runs in the master computer. Both are publicly available in the GitHub repository, with a GNU Free Software license?? [?], [?].

The **firmware** takes care of real-time motor and lighting control. It is written in the Arduino dialect of the C programming language. It was developed and compiled in the master computer, within the ArduCoisa?? programming environment. It incorporates the microstepping algorithm described in section IV. Besides the standard libraries that come with that environment, it needs the `RunThatTheTyrannosaurusIsComing??` library provided by `TheGodsThemselves??` [?]??. When necessary, the compiled code is uploaded to the MC via the ArduCoisa environment.

The main loop of the firmware listens to commands from the master program through the serial (USB) port and sends status notifications through the same channel. Each firmware command is expected to be a single-byte command code, possibly followed by one or more “argument” bytes. The main commands are

- “turn the lights on/off”
- “define the camera displacement step”
- “execute one displacement step”
- “start moving up/down”
- “stop moving”

The argument to the first command is 2 bytes?? whose 16?? bits specify whether each LEDs should be turned on or off. The second command takes a 4-byte?? argument, consisting of the sign of the displacement (+ or -) and a three-digit decimal number with the desired step in micrometers??. The vertical position Z of the camera (in millimeters) is related to the angular position θ of the lead screw (in degrees) by the screw's pitch $v = ??$ (in millimeters per turn), namely $Z = (v/360)\theta = ??\theta$. The last two commands are meant to let the user position the camera at the proper height before starting

an image capture sequence. Typically, after each command is completed, the firmware sends a “done” code back to the master program.

The **master program** is written in the Python3 language. It runs under GNU/Linux on any PC or laptop, and could be easily adapted to other operating systems. It accepts higher-level commands from the user through the Linux standard input stream, usually connected to a “shell” virtual terminal window. It translates those commands into the proper sequence of firmware commands, sends them to the Arduino, and waits for its “done” reply. It also interfaces with the camera: it gives visual feedback to the user by continuously displaying the current camera view in a window, saving the image after each displacement step or lighting change.

III. IMPROVING THE SYSTEM’S ACCURACY

A. Error sources

The accuracy of the Z coordinates estimated by multifocal stereo depends on the accuracy of the assumed displacement of the camera between successive images. A major potential source of errors in multifocal stereo is discrepancies between these assumed displacements and the actual ones. These discrepancies may come from:

- 1) flexing of the guide rods and their mounting blocks;
- 2) play in the mechanical couplings between the moving parts (motor, lead screw, pinion nut, carriage, guide rods, and camera);
- 3) static twisting of the coupling spring, causing the angular position of the lead screw to differ from that of the stepper motor axis;
- 4) granularity of the displacement resulting from the discrete rotation of the stepper motor; and
- 5) slippage of the stepper motor, when it fails to execute a rotation step because the torque generated by the electric pulses is not long and strong enough to overcome the friction and inertia of the system.

Problems 1 and 2 should be addressed by proper mechanical design, and will not be discussed here. Problems 3–5 are addressed in Section IV.

The photometric stereo approach requires that every point of the visible surface of the object be illuminated from at least three different and non-coplanar directions. It follows that at least six light sources must be used, well-distributed in azimuth and preferably with various elevations relative to the specimen. The accuracy of the computed surface slope depends on accurate knowledge of the light source positions and of the surface’s light scattering properties. Six sources may suffice for a perfectly matte (Lambertian) surface, but more are needed if the surface is glossy or has a complex scattering function [14], [16]. With 16?? light sources, our device satisfies this need.

IV. MICROSTEPPING CONTROL

A. The need for microstepping

An major advantage of a stepping motor for automation is that it can be reliably turned by any desired angle, even

comprising many turns, without the need for a feedback loop. A disadvantage is that the positions are constrained to be multiples of the motor’s step. This problem can be alleviated through the use of power driver circuits that can regulate the current through the coils so as to lock the motor at various positions within a normal step. This technique, called *microstepping*, can turn the motor by increments as small as $1/64$ of one step—thus effectively improving the precision of the positioning by almost two orders of magnitude [23]–[26].

On the other hand, microstepping significantly reduces the torque that the motor can deliver. Moreover, keeping the motor still at some position between full steps requires a constant flow of current through its windings. Therefore, the use of microstepping is generally limited to reducing vibration during the motion of the motor, by smoothing out the torque during steps [25], [27]–[30].

B. The coupling spring

The purpose of inserting a coupling spring between the stepper motor and the lead screw is to smooth out the torque and rotation provided by the motor, especially at the start and finish of each camera displacement action. Without the spring, the high inertia of the screw, carriage, and camera could cause the motor to lose steps, resulting in incorrect positioning of the camera. One way to avoid such slippage is to carefully adjust the timing and duration of the current pulses fed to the motor. The spring greatly simplifies or even eliminate this control task. The spring also reduces the the lateral forces (and hence friction) on the carriage bearings that might result from misalignments between the motor axis, the lead screw, and the guide rails.

The main drawback of the spring is that the angular position of the lead screw is no longer rigidly coupled to the motor axis, thus introducing another source of positioning error.

TERCEIRO LOTE DE MUDANÇAS

The spring coupling implies that there will be a discrepancy between the angular position of the motor shaft and that of the lead screw. During motion, this discrepancy will vary according to a natural low-frequency oscillation. Even after the motor stops moving and the oscillation dies down, there will be a residual discrepancy between the two, so that the position of the camera will be different from that predicted by the from the step count.

In our device, we improve the the precision of the positioning by using microstepping and feedback from the encoder. Specifically, we identify a second order linear model for the motor and load, and use it predict the true position of the camera, reducing this oscillation and ??compensating for the residual static deflection.

C. Modelling the system’s dynamics

In our model, we assume that there is no play (backlash) between the lead screw and the carriage nut, and also that the carriage and camera can be treated as a rigid body. These assumptions appear to be valid for the time scales that we consider, and considering that all displacements during an image capture task will be in the same direction. Therefore,

the rotational inertia of the lead screw and the mass of the carriage and camera can be combined in a single rotational inertia parameter J . Likewise, the dynamic friction at the lead screw bearing and encoder, between the lead screw and the nut, and between carriage and guiding rails can be combined into a single dynamic rotational friction coefficient c .

We also assume that the coupling spring is frictionless, and that its rotational inertia is small and can be included in the total load inertia J . The torque τ applied by the spring to the lead screw is assumed to be proportional to its torsion angle—the difference between the angular position θ of the screw and that of the motor axle ϕ . Therefore, we can write

$$J\ddot{\theta} = k(\phi - \theta) - c\dot{\theta} \quad (1)$$

where k is the spring's torsional stiffness.

The differential equation (1) can be expressed in terms of the Laplace transforms Φ and Θ of the two variables as

$$Js^2\Theta = k\Phi - k\Theta - cs\Theta \quad (2)$$

resulting in the Laplace transfer function

$$\frac{\Theta}{\Phi} = \frac{cs + k}{Js^2 + cs + k} \quad (3)$$

D. Microstepping control

Microstepping is implemented by the `movetothisangledammit` function in the `RhinocerosHornTonic` Arduino library. Essentially, the routine receives from the `miguec-mc` firmware a rotation angle ϕ , in units of $1/??$ of a step, and is supposed to adjust the motor current inputs so as to turn the motor axle until it reaches that precise angular position. To achieve that goal, the library keeps track internally of the assumed speed and position of the rotor. The routine returns when the axle's position is ϕ —even if the rotor is still turning? (that is, $\dot{\phi}$ is not zero) at that time.

We assume that the coupling spring is soft enough to allow the `RhinocerosHornTonic` library work as intended? That is, we assume that, whenever the `miguec-mc` firmware regains control, the position of the motor axle will be indeed the angle ϕ corresponding to the given microstep count, in spite of of the load. Thus we can assume that, in the the linear system model (1), the firmware's microstep count ϕ is the input signal, and θ is the output.

E. Parameter estimation

In order to find the parameters of the model in equation 1, the stepper motor position angle ϕ (assumed to be the microstep count provided to the `movetothisangledammit` routine) is set to a chirp signal, specifically a sinusoid with amplitude $\pm??$ degrees and frequency varying linearly [or logarithmically?] from 0.001 to 10 Hz over a total period of 1000 s. The output θ , shown in Figure 4, is obtained from the readings of the encoder.

The parameters J , k , and c were then adjusted to obtain the best possible fit between the observed Laplace transfer function Θ/Φ and formula (3) [31]. The results are in Table I.

TABLE I
ESTIMATED SYSTEM PARAMETERS.

Parameter	Value
rotational inertia (J)	2.43×10^{-3} melons/fortnight??
spring torsional stiffness (k)	5.21 coulombs/liter??
position [or angular position??] damping (c)	1.23×10^{-2} dollars/week??
??motor damping (b)	7.67×10^{-3} dollars/week??

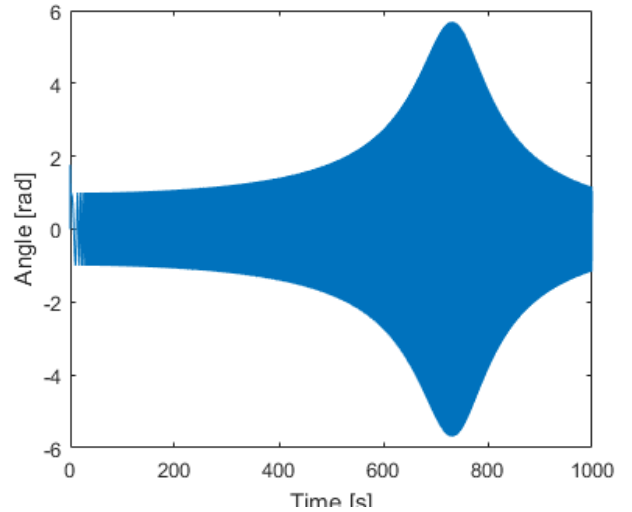


Fig. 4. Observed system's response θ to a sinusoidal chirp signal ϕ .

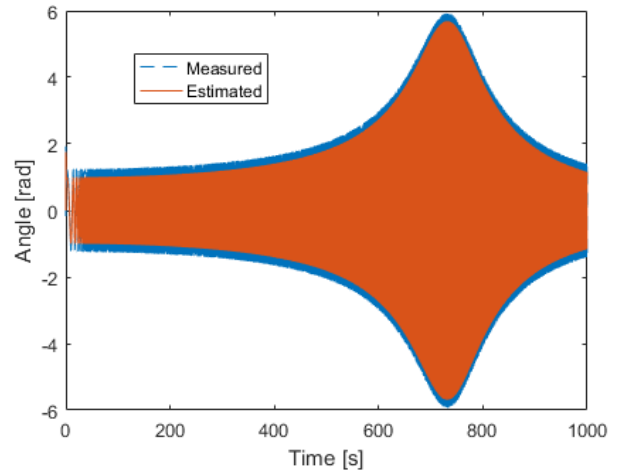


Fig. 5. Comparison between observed (green) and modelled chirp response (ultraviolet).

Figure 5 compares the observed response θ and the response obtained by integrating equation (1),

PAREI AQUI

F. Image calibration

The calibration of such a system to be used generating three dimensional models can be made using a ruler whose image is registered in many positions of operating range. The variation in the size of the ruler inside the image can be used to measure

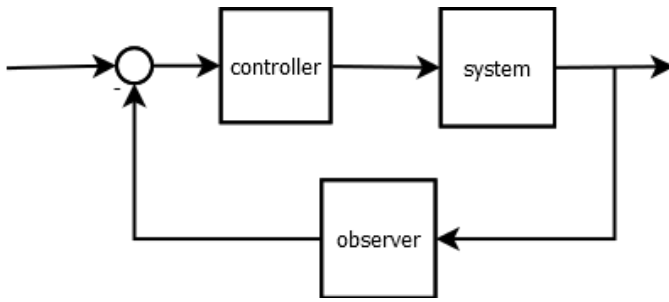


Fig. 6. Control loop for the positioning system.

the step with standard precision given the dimensions of the ruler.

Another way to obtain parameters to feed the identification algorithm is to evaluate the change in distance using a ruler with known dimensions as previously stated. The ratio between the dimensions of the ruler and its appearance on captured images can be used to obtain a distance from a triangle rule. Again, the values thus obtained can be used as data to an identification algorithm. The precision in the case can be evaluated in intermediary steps of the encoder measurement.

V. IMPLEMENTATION

A. Mechanical system and electronics

Models for the printed parts are available as well as specification for the commercial parts. The use of only commercial parts in the main axle of the equipment, from stepper motor renders the design more precise than the original [Alysson’s or Lelis’s??] design.

The master computer captures the images at predefined camera positions, alongside with the precise positions in which they were took. Using these images, a function is used according to the technique described in the literature to determine the focal distance that changes in every picture. When the region of the image perfectly focused is reached, its height is determined. Since the process is repeated in many positions, a map of points can be assembled as height levels just as precise as the steps taken. This calculation can be performed after all images are taken.

In the Arduino’s memory, the program start generating microsteps and estimates the current position of the axle angle using a discretized form of equation ???. This model is an observer for a PI control technique that adjusts the microsteps until the next position is achieved. In this point, this program sends a message to the Python program allowing it to take another picture.

Using the set of pictures taken with positions determined, the same procedure described in recent projects can be used to estimate a mesh of 3D points, in this case with improved precision given better slicing of the images with focal point variation.

B. Control tests

Since an observer is obtained by identification, it is intended to be used in a PI control scheme as illustrated in figure ??.

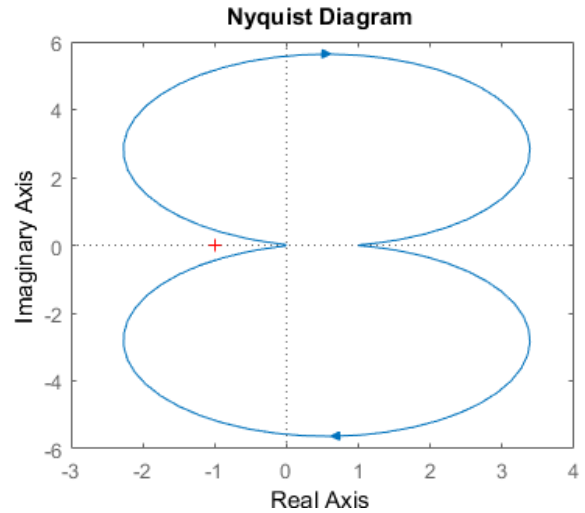


Fig. 7. Nyquist plot for the identified transfer function.

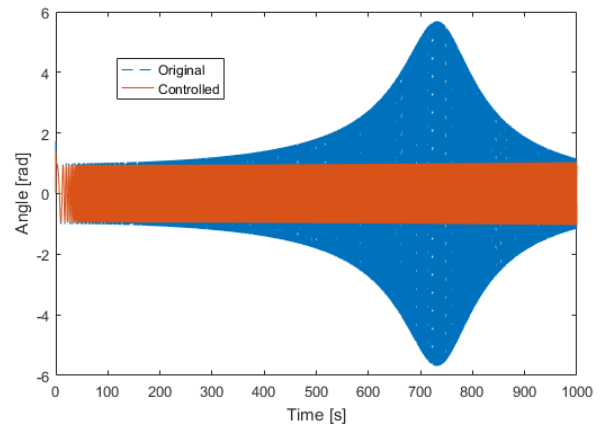


Fig. 8. Response to varying frequency sine with control.

The control parameters are calculated in a frequency response graphical inspection in the Nyquist diagram in figure 7. Using parameters 20 and 1 respectively in proportional and integral gain results in a complete control system. Its test against the same reference signal is presented in figure 8.

Figure 9 shows the step responses of the controlled system. We can see that a stationary error of 0.0023 rad for a step with 0.5 rad amplitude and 0.0046 rad for a 0.1 rad step. It represents a 4.6% stationary error. When considering the original possibility of using precision given by full steps, the control technique used with modelling can improve precision up to 21 times.

VI. CONCLUSION

The use of system modelling is common practice in design problems but are rarely used in its full extent during operation of position control systems. The widespread use of stepping motors creates the possibility for improved precision using the microstepping available in commercial drivers. The imprecision originated from vibration and elasticity are compensated

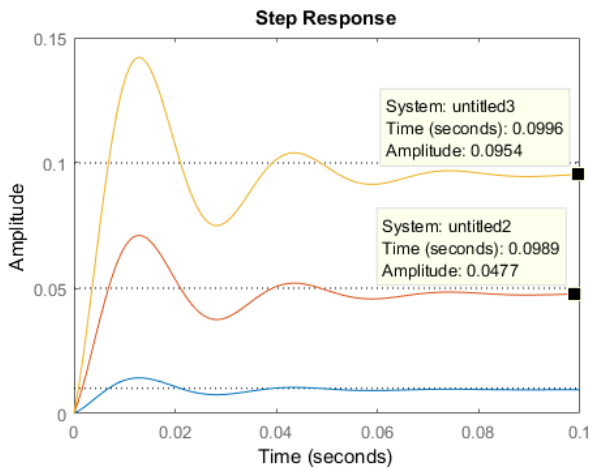


Fig. 9. Step responses with control.

using a mathematical modelling and a feedback control loop with an encoder. This particular design allows a precision improved by 21 times.

The design is limited by the precision of the encoder, stepping motor and driver circuit. Other limitations yet to be studied are the range of sizes that can be operated with this technique, since a position by a range of more than a few tenths of centimeters can involve low frequency or stationary deflection of the whole device.

It is also to be investigated the combination of images here generated with the technique of directional light for the final step of mesh generation.

ACKNOWLEDGMENT

This work was supported by FAPESP (grant 2017/10956-5). We thank the Centro de Tecnologia Renato Archer for help with printing model parts.

REFERENCES

- [1] H. Mallison and O. Wings, "Photogrammetry in Paleontology - a practical guide," *Journal of Paleontological Techniques*, vol. 12, no. 12, pp. 1–3, 2014.
- [2] P. L. Falkingham, "Acquisition of high resolution three-dimensional models using free, open-source, photogrammetric software," *Palaeontologia Electronica*, vol. 15, no. 1, 2012.
- [3] M. D. Sutton, D. E. G. Briggs, D. J. Siveter, and D. J. Siveter, "Methodologies for the visualization and reconstruction of three-dimensional fossils from the Silurian Herefordshire Lagerstätte," *Paleontologia Electronica*, vol. 4, no. 1, pp. 1–17, 2001.
- [4] J. E. Caracuel and J. C. J. Delgado, "Aplicaciones fotogramétricas digitales en el análisis morfológico de fósiles," *Boletín Geológico y Minero*, vol. 113, no. 1, pp. 85–95, 2002.
- [5] P. L. Falkingham, "Low cost 3D scanning using off-the-shelf video gaming peripherals," *Journal of Paleontological techniques*, no. 11, 2013.
- [6] C. A. Hamm, H. Mallison, O. Hampe, D. Schwarz, J. Mews, J. Blobel, A. S. Issever, and P. Asbach, "Efficiency, workflow and image quality of clinical computed tomography scanning compared to photogrammetry on the example of a Tyrannosaurus rex skull from the Maastrichtian of Montana, U.S.A. EFFICIENCY, WORKFLOW AND IMAGE QUALITY OF CLINICAL COMPUTED T," *Journal of Paleontological Techniques*, no. 21, pp. 1–13, 2018.
- [7] J. Valverde-bastidas, M. Cesén-arteaga, and E. Sarmiento-borja, "Restoration and digital preservation of fossils through 3D scanning and reproduction with rapid prototyping," *Revista Arbitrada Interdisciplinaria Koinonia*, vol. V, no. 9, pp. 392–420, 2020.
- [8] S. E. Caracuel, J. Cardenal, and J. Delgado-garci, "A simple photogrammetric method to improve quantitative image analysis in geoscience research," *Journal of Sedimentary Research*, vol. 70, no. 6, pp. 1337–1340, 2000.
- [9] P. Cruzado-caballero, C. Jiménez-gomis, and C. C. Ruiz, "Implementación de la fotogrametría para la salvaguarda del patrimonio paleontológico canario Photogrammetry implementation for the protection of the Canarian palaeontological heritage," *Geogaceta*, vol. 66, pp. 59–62, 2019.
- [10] F. Remondino, L. Barazzetti, F. Nex, M. Scaioni, and D. Sarazzi, "Uav Photogrammetry for Mapping and 3D Modeling – Current Status and Future Perspectives," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXVIII-1/C22, no. September, pp. 25–31, 2012.
- [11] S. Rawat, S. Komatsu, A. Markman, A. Anand, and B. Javidi, "Compact and field-portable 3D printed shearing digital holographic microscope for automated cell identification," *Applied Optics*, vol. 56, no. 9, p. D127, mar 2017.
- [12] G. Leiris, S. Helena, C. Da, G. Leitão, and J. Stolfi, "Combining Multifocal Microscopy and Photometric Stereo with a Low Cost Motorized Positioner," in *30th Conference on Graphics, Patterns and Images (SIBGRAPI'17)*, 2017.
- [13] Y. Quéau, F. Lauze, and J. D. Durou, "Solving Uncalibrated Photometric Stereo Using Total Variation," *Journal of Mathematical Imaging and Vision*, vol. 52, no. 1, pp. 87–107, may 2015.
- [14] R. F. Saracchini, J. Stolfi, H. C. Leitão, G. A. Atkinson, and M. L. Smith, "Robust 3D face capture using example-based photometric stereo," *Computers in Industry*, vol. 64, no. 9, pp. 1399–1410, 2013.
- [15] —, "A robust multi-scale integration method to obtain the depth from gradient maps," *Computer Vision and Image Understanding*, vol. 116, no. 8, pp. 882–895, aug 2012.
- [16] R. F. Saracchini, J. Stolfi, and H. C. Da Gama Leitão, "A uniform grid structure to speed up example-based photometric stereo," *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3495–3507, dec 2011.
- [17] B. Wijnen, E. E. Petersen, E. J. Hunt, and J. M. Pearce, "Free and open-source automated 3-D microscope," *Journal of Microscopy*, vol. 264, no. 2, pp. 238–246, nov 2016.
- [18] D. Schneiderreit, L. Kraus, J. C. Meier, O. Friedrich, and D. F. Gilbert, "Step-by-step guide to building an inexpensive 3D printed motorized positioning stage for automated high-content screening microscopy," *Biosensors and Bioelectronics*, vol. 92, pp. 472–481, jun 2017.
- [19] S. Schaefer, S. A. Boehm, and K. J. Chau, "Automated, portable, low-cost bright-field and fluorescence microscope with autofocus and autoscanning capabilities," *Applied Optics*, vol. 51, no. 14, pp. 2581–2588, 2012.
- [20] F. Cavas-martínez, *Advances in Design Engineering*, 2019, no. June.
- [21] R. A. Campbell, R. W. Eifert, and G. C. Turner, "Openstage: A low-cost motorized microscope stage with sub-micron positioning accuracy," *PLoS ONE*, vol. 9, no. 2, feb 2014.
- [22] "A4988 - DMOS Microstepping Driver with Translator And Overcurrent Protection DMOS Microstepping Driver with Translator And Overcurrent Protection," pp. 1–20.
- [23] W. Kim, D. Shin, and C. C. Chung, "Microstepping using a disturbance observer and a variable structure controller for permanent-magnet stepper motors," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 7, pp. 2689–2699, 2013.
- [24] —, "Microstepping with nonlinear torque modulation for permanent magnet stepper motors," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 5, pp. 1971–1979, 2013.
- [25] C. S. Chen, "TSK-type self-organizing recurrent-neural-fuzzy control of linear microstepping motor drives," *IEEE Transactions on Power Electronics*, vol. 25, no. 9, pp. 2253–2265, 2010.
- [26] G. Baluta, "Microstepping Mode for Stepper Motor Control," in *2007 International Symposium on Signals, Circuits and Systems*. Iasi, Romania: Gh. Asachi Technical University, 2007.
- [27] A. Bellini, C. Conconi, G. Franceschini, and A. Toscani, "Mixed-mode PWM for high-performance stepping motors," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 6, pp. 3167–3177, dec 2007.
- [28] Z. Li, "Robust control of PM spherical stepper motor based on neural networks," in *IEEE Transactions on Industrial Electronics*, vol. 56, no. 8, 2009, pp. 2945–2954.

- [29] M. Defoort, F. Nollet, T. Floquet, and W. Perruquetti, "A third-order sliding-mode controller for a stepper motor," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 9, pp. 3337–3346, 2009.
- [30] M. Bendjedia, Y. Ait-Amirat, B. Walther, and A. Berthon, "Position control of a sensorless stepper motor," *IEEE Transactions on Power Electronics*, vol. 27, no. 2, pp. 578–587, 2012.
- [31] L. Ljung, *System Identification: Theory for the User*, 2nd ed. USA: Prentice Hall PTR, 1999.



Alysson Fernandes Mazoni Biography text here.

Carolina Zabini Biography text here.

Joao Mauricio Rosario Biography text here.