

LUCAS ANTÔNIO OLIVEIRA RODRIGUES

**DESENVOLVIMENTO DA ESTRUTURA HOPE-G: UM
SISTEMA ROBÓTICO PARA REABILITAÇÃO DA
MARCHA HUMANA**



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA

2022

LUCAS ANTÔNIO OLIVEIRA RODRIGUES

**DESENVOLVIMENTO DA ESTRUTURA HOPE-G: UM
SISTEMA ROBÓTICO PARA REABILITAÇÃO DA
MARCHA HUMANA**

Tese de Doutorado apresentado ao Programa de Pós-graduação em Engenharia Mecânica da Universidade Federal de Uberlândia, como parte dos requisitos para obtenção do título de **DOUTOR EM ENGENHARIA MECÂNICA**.

Área de concentração: Mecânica dos sólidos e Vibrações.

Orientador: Prof. Dr. Rogério Sales Gonçalves

UBERLÂNDIA – MG

2022

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU
com dados informados pelo(a) próprio(a) autor(a).

R696
2022 Rodrigues, Lucas Antonio Oliveira, 1990-
DESENVOLVIMENTO DA ESTRUTURA HOPE-G [recurso
eletrônico] : UM SISTEMA ROBÓTICO PARA REABILITAÇÃO DA
MARCHA HUMANA / Lucas Antonio Oliveira Rodrigues. -
2022.

Orientador: Rogério Sales Gonçalves.
Tese (Doutorado) - Universidade Federal de Uberlândia,
Pós-graduação em Engenharia Mecânica.
Modo de acesso: Internet.
Disponível em: <http://doi.org/10.14393/ufu.te.2022.84>
Inclui bibliografia.
Inclui ilustrações.

1. Engenharia mecânica. I. Gonçalves, Rogério Sales,
1982-, (Orient.). II. Universidade Federal de
Uberlândia. Pós-graduação em Engenharia Mecânica. III.
Título.

CDU: 621

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:

Gizele Cristine Nunes do Couto - CRB6/2091

“Só existem dois dias importantes na nossa vida: o dia que a gente nasce e o dia que a gente descobre porque que a gente nasceu.”

[Lucas Silveira - Fresno]

DEDICATÓRIA

Mãe,

Este não será apenas um agradecimento genérico que a maioria das pessoas faz. Sei que a maioria delas de fato teve uma relação tão boa quando nós tivemos, mas sei que provavelmente fui o único a ouvir que seu último desejo seria este trabalho concluído.

Você me trouxe até aqui. Me incentivou e sacrificou tudo que pôde para que cada palavra aqui escrita se tornasse realidade, até o seu último suspiro.

Minha maior dor foi não ter tido a oportunidade de te dizer naquele quarto de hospital que eu consegui cumprir com o seu último pedido. Mas esta dor é o guia fundamental para que, de onde quer que você esteja, você saiba: nós conseguimos. Nós.

Saiba que a dor da sua falta jamais será maior que a alegria que eu e todos que estiveram ao seu redor guardamos em nossas memórias.

AGRADECIMENTOS

À Deus, que mesmo num tempo tão obscuro onde as pessoas confundem suas palavras e idolatram homens com sangue nas mãos, não afastou seu cuidado de nós.

À Universidade Federal de Uberlândia e à Faculdade de Engenharia Mecânica pela oportunidade de realizar este Curso.

À minha esposa, que divide este caminho comigo desde o seu início, e dividiu comigo todas as dores e glórias trazidas por ele.

À CAPES pelo apoio financeiro.

À FAPEMIG pelo auxílio financeiro através do processo TEC - APQ-02829-17.

Aos meus pais, que foram o alicerce que me suportou em todos os momentos que precisei ao longo dessa jornada.

Ao meu orientador, prof. Rogério Sales, pela paciência e sabedoria para me direcionar, orientar ao longo de mais de 10 anos, desde o início da graduação, até a realização deste projeto. Obrigado por nunca me deixar desanimar ou desistir.

RODRIGUES, L.A.O., **Desenvolvimento da Estrutura HOPE-G: Um sistema robótico para reabilitação da Marcha Humana.** 2022, 196f. Tese de Doutorado, Universidade Federal de Uberlândia, Uberlândia.

RESUMO

Os processos modernos de reabilitação para pacientes com doenças neurológicas têm sido amplamente auxiliados por estruturas robóticas de diversos tipos, as quais estão constantemente sendo pesquisadas e melhoradas. Embora a aplicação de estruturas robóticas esteja em um estágio avançado para reabilitação do membro superior, a marcha humana ainda carece de inovação. Com base neste fato, este trabalho destinou-se ao desenvolvimento de uma nova estrutura paralela para reabilitação da marcha humana de pacientes pós Acidente Vascular Encefálico, denominada HOPE-G. A estrutura é composta por três módulos distintos, e esta tese tem como objetivo realizar um teste piloto, para verificar a viabilidade da utilização de um suporte ativo de peso corporal integrado a um jogo eletrônico para auxiliar a realização de exercícios em uma esteira, observando principalmente a influência no equilíbrio dos participantes da pesquisa. Neste trabalho, realizou-se o desenvolvimento de dois destes módulos e o teste experimental de um deles. Inicialmente, realizou-se uma revisão bibliográfica do estado da arte das tecnologias aplicadas nesta área do conhecimento. A seguir, os módulos mecânicos da estrutura foram modelados seguindo técnicas consolidadas de síntese de estruturas multicorpos e otimizados com técnicas de otimização meta-heurística. Na sequência, um protótipo de um dos módulos foi construído. A partir da estrutura, o sistema de controle dos motores foi elaborado com auxílio de métodos computacionais e integrado a um jogo eletrônico desenvolvido para atuar em conjunto com o equipamento. O controle da estrutura foi desenvolvido utilizando controladores PID obtendo-se os parâmetros por otimização meta-heurística, e associando-os a uma função de supressão para assistir quando necessário. O módulo foi então submetido a testes piloto com um boneco antropométrico de madeira e participantes saudáveis onde foram coletados dados durante cada teste. Os resultados do teste piloto mostraram bom funcionamento do design proposto com recomendações para melhorias elétricas e mecânicas. Os dados e recomendações deste estudo serão utilizados como referência para um estudo futuro da eficácia do dispositivo no processo de reabilitação da marcha humana em pacientes em tratamento do AVE.

RODRIGUES, L.A.O., **Development of HOPE-G: A robotic system for human gait rehabilitation**. 2022, 196p. Doctorate Thesis, Federal University of Uberlândia, Uberlândia.

ABSTRACT

Modern rehabilitation processes for neurological patients have been widely assisted by robotic structures, with continuous research and improvements. The use of robotic assistance in rehabilitation is a consolidated technique for upper limb training sessions. However, human gait robotic rehabilitation still needs further research and development. Based on that, this work deals with the development of new parallel gait rehabilitation structure designed for post-stroke patients, named HOPE-G. The structure is composed of three distinct modules, and this thesis has the pilot tests of the active bodyweight support integrated with a serious game as the main objective. The test objective is to check the feasibility of the proposed design aggregated with a treadmill, focused on the gait balance. This work also presents the mathematical model of the two mechanical modules, while building a prototype for experimental tests for one of them. The thesis begins with a background review of the state of art of the applied technologies in this area of knowledge. Next, it presents the obtained mathematical models for the mechanical modules of HOPE-G following multibody synthesis techniques and meta-heuristic optimization. After, computational simulations are performed and based on the results, the prototype is built. The control of the structure is designed using PID controllers tuned with meta-heuristic optimization and associated with a suppression function to perform assist-as-needed actions. Then, the prototype is integrated with a serious game designed specifically to this application. Finally, the tests are conducted with the structure, first with a wooden anthropometric puppet, and then with healthy volunteers. The results obtained from these first pilot tests shows that the mobility of the module is as expected, with recommendations for electrical and mechanical improvements. With these improvements, the module will be submitted to clinical trials with a post-stroke research group.

SUMÁRIO

Capítulo I	INTRODUÇÃO	18
Capítulo II	REVISÃO BIBLIOGRÁFICA	23
2.1.	Cinesiologia da Marcha Humana	23
2.2.	Estruturas para reabilitação da marcha humana existentes	34
2.3.	Estudo de manipuladores robóticos.....	28
2.4.	Controle de estruturas robóticas aplicadas na reabilitação.....	47
2.5.	Uso de jogos eletrônicos para reabilitação	53
2.6.	Aspectos selecionados para HOPE-G	54
Capítulo III	MODELAGEM MATEMÁTICA DAS ESTRUTURAS.....	56
3.1.	Modelagem Matemática do Módulo de Esteiras Bipartidas.....	56
3.2.	Modelagem Matemática do Módulo de Suporte Ativo de Peso Corporal.....	58
3.3.	Modelo Dinâmico e aplicação da Simulação & Otimização.....	67
3.4.	Simulações Computacionais em CAD/CAE	73
Capítulo IV	PROTÓTIPO PARTE 1: ATUADORES E CONTROLE	78
4.1.	Materiais utilizados	78
4.2.	Testes de bancada do atuador.....	80
4.3.	Montagem da estrutura passiva	82
4.4.	Montagem dos circuitos e módulos de controle da estrutura	85
4.5.	Configuração dos atuadores	87
4.6.	Comunicação com PC	93
4.7.	Testes de movimentação	94
Capítulo V	PROTÓTIPO PARTE 2: JOGO ELETRÔNICO	97
5.1.	Criação do cenário e obstáculos	97
5.2.	Sistema de pontuação	99
5.3.	Personagem e câmera.....	100
5.4.	Interface com o usuário	101
5.5.	Comunicação com a estrutura	102
Capítulo VI	PROTÓTIPO PARTE 3: TESTES PILOTO E RESULTADOS.....	104

6.1. Teste de integração jogo-estrutura utilizando boneco antropométrico.....	107
6.2. Teste com participantes saudáveis	110
CONCLUSÕES.....	113
REFERÊNCIAS	117
APÊNDICES	126
Planificações do Módulo de Suporte Ativo de Peso Corporal	126
Códigos aplicados para modelagem dinâmica das esteiras bipartidas.....	139
Otimização das barras do Módulo de Suporte Ativo de Peso Corporal baseada na modelagem matemática.....	142
Modelagem dinâmica dos atuadores do Módulo de Suporte Ativo de Peso Corporal	154
Otimização dos parâmetros de controle dos atuadores.....	157
Código-fonte da central de controle (microcontrolador).....	162
Scripts utilizados para desenvolvimento do jogo sério	177
Projeto de Pesquisa para testes em seres humanos aprovado no CEP.....	193

LISTA DE FIGURAS

Figura 1 – Imagem conceitual inicial exemplificando os 3 módulos propostos para a estrutura HOPE-G.	21
Figura 2 – Desenho esquemático descrevendo o ciclo da marcha e suas respectivas fases (SACCO; TANAKA, 2008).....	25
Figura 3 – Detalhamento do passo de cada membro e da passada da marcha humana (SACCO; TANAKA, 2008).....	26
Figura 4 – Assimetria entre o membro parético e não-parético observada em pacientes pós-AVE (DYER et al., 2014).	27
Figura 5 – Articulação sacroilíaca, também conhecida como articulação pélvica (Fonte: Google Imagens).....	27
Figura 6 – Planos de referência da Anatomia Humana (Fonte: adaptada de https://maestrovirtuale.com/planos-anatomicos-e-eixos-do-corpo-humano/).....	28
Figura 7 – Tipos de estruturas robóticas utilizando (a) esteiras, (b) base programável para os pés, (c) base robótica móvel e (d) base robótica estacionária (RODRIGUES, 2017).	35
Figura 8 – Imagem da versão comercial da estrutura LOKOMAT, distribuída pela Hocoma® (fonte: https://www.hocoma.com/solutions/lokomat/).....	36
Figura 9 – (a) Esquema cinemático da estrutura LOPES; e (b) Imagens de um paciente utilizando a versão mais recente do equipamento (adaptada de http://balance-fp7.eu e VENENAM et al. 2007).	37
Figura 10 – Montagem do suporte para peso ativo proposto por Roberts (2004).	38
Figura 11 – (a) Desenho esquemático da estrutura; e (b) protótipo desenvolvido para a estrutura de reabilitação assistida por exoesqueleto (LI; CHEN; FAN, 2017).	38
Figura 12 – Estrutura RGR Trainer: A- Potenciômetro linear, B – guias lineares, C – Braço pélvico, D – esteira, E – Servo motor/atuador linear, F – Célula de carga e articulação esférica (PIETRUSINSKI et al., 2010).....	39
Figura 13 – (a) Esquema de montagem da estrutura serial IWBS para atuar na pélvis; e (b) Articulação pélvica humana totalmente suportada pelo socket da IWBS (SALGUERO-BELTRAN et al., 2012).	40
Figura 14 – (a) Esquema ilustrando a montagem da estrutura PAM; (b) Protótipo construído sendo utilizado em conjunto com a estrutura Arthur 2.0 (AOYAGI et al., 2004).....	40
Figura 15 - Protótipo da estrutura MIT Skywalker (SUSKO, 2015).	41
Figura 16 – (a) Desenho esquemático da patente e; (b) Imagem do produto da estrutura ReWalk (adaptada de Goffer 2006 http://rewalk.com/).	42
Figura 17 – Modelo tridimensional detalhado e; (b) protótipo construído da estrutura proposta por MUN et al. (2014).	42

Figura 18 – Testes experimentais realizados por KATAOKA et al. (2017) para comparar os efeitos de suportes corporais baseado em (a) cintos e colete de segurança e (b) em assentos do tipo selim.	43
Figura 19 – Imagem comercial da estrutura AlterG® (fonte: http://www.alterg.com).....	44
Figura 20 – Elementos fundamentais de uma estrutura robótica (SCIAVICCO; SICILIANO, 2005).	29
Figura 21 – (a) Exemplo de uma estrutura robótica serial; e (b) exemplo de uma estrutura robótica paralela (fonte: KuKa Robotics® e Wikipedia Commons).....	30
Figura 22 – (a) Guia linear, utilizada como articulação prismática; (b) rótula mecânica; (c) junta universal ou cardam; (d) junta mecânica com rolamento, utilizada como articulação cilíndrica; (e) junta esférica.	31
Figura 23 – Esquema de uma estrutura (a) serial RRP e; (b) paralela 3-PRRR de 3 GDL (adaptada de Gosselin 2009).....	32
Figura 24 – Representação gráfica das transformações de coordenadas definidas no (a) Modelo Geométrico Direto e (b) Modelo Geométrico Inverso (SCIAVICCO; SICILIANO, 2005).....	33
Figura 25 – Representação computacional da estrutura 2UPS-SPR (LU et al.,2014).	45
Figura 26 – (a)Esquema da composição da estrutura 5RRR(RR); e (b) protótipo construído (Adaptada de ZHU; HUANG; ZHAO, 2009).	46
Figura 27 - Exemplos de manipuladores paralelos 3P2R: (a) Mecanismo 5-PRPaRR ^a ; (b) mecanismo 5-PRPaRR ^b ; (c) mecanismo 4-PP ^a RU ^e +1-PP ^a RU ^{x1} ; (d) mecanismo 4-PPaRUf+1-PPaRUx1 (MOTEVALLI; ZOHOOR; SOHRABPOUR, 2010).	47
Figura 28 – Protótipo de prótese transfemoral com controle de impedância ativo (HONG; ANIL KUMAR; HUR, 2021).....	49
Figura 29 – Diagrama generalizado representando as etapas de Aprendizado por Demonstração. (1 e 2) Paciente e terapeuta movem um objeto, representado por uma porta com fechamento automático mecânico, enquanto o equipamento adquire dados sobre força e trajetória. (3-4) Em seguida, o paciente opera o sistema sozinho, e o robô obtém dados adicionais de força e trajetória. (5-6). O Sistema então utiliza os dados para construir um modelo preditivo e utiliza como dados de entrada o deslocamento do sistema, e gera saídas de forças conforme necessário para manter a trajetória conforme o modelo construído (FONG; TAVAKOLI, 2018).....	50
Figura 30 – Estrutura de reabilitação do membro superior atuada por cabos e controlada por aprendizado por demonstração (NUNES et al., 2012).	50
Figura 31- Estrutura de reabilitação combinada ReGear, construída utilizando a metodologia de controle “Assistir quando necessário” (RODRIGUES; MORAES; GONÇALVES, 2019).	52
Figura 32 – Esquema estrutural adotado para o Módulo de Movimentação por Esteira Bipartida: (a) representação da posição inicial; e (b) representação da posição de descida máxima.....	57

Figura 33 – Representação trigonométrica da (a) posição inicial; e (b) da posição de descida máxima.....	57
Figura 34 – (a) Desenho esquemático para a estrutura do suporte ativo da pélvis, baseada no “pentapteron” (4-PRRU+PRRS); e (b) posição do ponto de interesse P em relação aos vértices da plataforma móvel.	59
Figura 35 - Análise da posição do ponto A em relação ao referencial inercial e a perna 1.	60
Figura 36 - Ângulos entre os vetores unitários e os vetores de posição das extremidades em relação ao ponto P: (a) ângulos da rotação em torno de XO ; e (b) da rotação em torno de ZO	61
Figura 37 - Análise da posição do ponto B em relação ao referencial inercial e a perna 2.....	62
Figura 38 - Análise da posição do ponto C em relação ao referencial inercial e a perna 3.....	63
Figura 39 - Análise da posição do ponto D em relação ao referencial inercial e a perna 4.	64
Figura 40 - Análise da posição do ponto E em relação ao referencial inercial e a perna 5.....	65
Figura 41 – (a) Modelo CAD construído para o atuador baseado no mecanismo biela-manivela; (b) desenho esquemático evidenciando os parâmetros adotados.	66
Figura 42 – (a) gráfico da distância da extremidade de uma das esteiras em relação ao chão em função da extensão do comprimento do pistão; e (b) valores calculados para deslocamento, velocidade e aceleração em cada instante de tempo durante a descida da esteira.....	68
Figura 43 - sistema de supressão passivo da carga estática do peso do paciente presente nos atuadores 2 e 4 e na plataforma móvel: (a) desenho esquemático; (b) Implementação no protótipo real.....	70
Figura 44 – Diagrama de corpo livre do mecanismo biela-manivela dos atuadores.....	71
Figura 45 – Sequência de imagens representativas ilustrando a simulação efetuada para o módulo de movimentação por esteira bipartida: (a) Posição inicial, alinhada com a referência; (b) posição intermediária de descida; (c) posição de descida máxima, onde há contato com o solo; e (d) posição intermediária de subida, retornando ao alinhamento com a referência.....	73
Figura 46 – Sequência de imagens representando a simulação dos movimentos elementares. ..	75
Figura 47 – Sequência de imagens representando a simulação do procedimento de colocação do paciente na estrutura.....	76
Figura 48 – Sequência de imagens representando as simulações operacionais da HOPE-G.	76
Figura 49 – Protótipo construído para atuador do Módulo de Suporte Ativo de Peso Corporal. 81	
Figura 50 – Montagem parcial do módulo de suporte ativo durante a elaboração do sistema de controle.....	83
Figura 51 – Detalhe da instalação dos módulos de atuação, utilizando retalhos para posicionamento do suporte do motor DC e priorizando o alinhamento das guias lineares em relação à estrutura completa.	84

Figura 52 – Detalhe da construção da plataforma móvel. (a) Destaque do posicionamento das articulações, selim e cortes para alívio de peso; e (b) detalhe da construção de uma das articulações utilizando junta universal.	85
Figura 53 – Projeto CAD/CAE criado para a central de controle da estrutura. (a) Desenho esquemático com elementos representativos e conexões utilizando jumpers e protoboard; e (b) transposição do projeto para PCI.	86
Figura 54 – PCI da central de controle do suporte de peso ativo, já com pin headers e drivers de corrente instalados.	86
Figura 55 – Esquema de controle aplicado no protótipo de suporte de peso ativo.	87
Figura 56 – Fluxograma detalhando o funcionamento do algoritmo de evolução diferencial (RODRIGUES, 2017).	90
Figura 57 – Arquitetura desenvolvida para a API de otimização dos atuadores.	90
Figura 58 – Malha de controle aplicada nos atuadores, incluindo um controlador PID e uma função de AAN.	92
Figura 59 – Representação da dead-zone gerada pela função tangente hiperbólica: Em verde, a função aplicada diretamente; em vermelho a função combinada com uma função cúbica; em azul, a função combinada com uma constante para ganho (ASL; NARIKIYO; KAWANISHI, 2019).	92
Figura 60 – Diagrama de funcionamento dos modos da API de comunicação da central de controle.	93
Figura 61 – Testes de movimentação da plataforma móvel do protótipo de suporte ativo de peso. (a) Rotação no plano transversal (em torno do eixo OZ); (b) Rotação no plano frontal (em torno do eixo OX); (c) Translação no eixo OX; (d) Translação no eixo OY; e (e) Translação no eixo OZ.	95
Figura 62 – Boneco antropométrico de madeira, posicionado sobre uma esteira e fixado à plataforma móvel da estrutura para testes de movimentação.	96
Figura 63 – Comparativo: imagem (a) sem o efeito de névoa ao fundo e imagem (b) com o efeito.	98
Figura 64 – Exemplo de recurso de segmento da pista.	99
Figura 65 – Cubo bônus no cenário.	100
Figura 66 – Personagem principal.	100
Figura 67 – Visão do jogador durante a rotação do cenário, destacando o diagrama de interface.	101
Figura 68 – Menu principal do jogo.	102
Figura 69 – Teste de integração jogo-estrutura utilizando boneco antropométrico.	109
Figura 70 – Sessão de teste com participantes saudáveis.	111

LISTA DE TABELAS

Tabela 1 - Espaço de trabalho especificado para cada grau de liberdade do módulo de suporte ativo de peso corporal (Roberts, 2004).	69
Tabela 2 – Dimensões ótimas calculadas para cada barra do módulo.	72
Tabela 3 – Especificações para os motores DC do módulo.	73
Tabela 4 – Lista de materiais aplicados na estrutura passiva.	78
Tabela 5 – Lista de materiais aplicados na estrutura passiva.	79
Tabela 6 – Lista de materiais aplicados no sistema de controle.....	80
Tabela 7 – Resultados obtidos para os parâmetros dos controladores após a otimização.....	91
Tabela 8 – Comandos de entrada e saída da API de operação das ações de rotação da estrutura.	93

CAPÍTULO I INTRODUÇÃO

O Acidente Vascular Encefálico (AVE) é definido como a perda de uma ou mais funções neurológicas devido à interrupção do fluxo sanguíneo em uma determinada região do encéfalo (BAGGI; RODRIGUES; CARAMÊZ, 2014). O AVE possui dois tipos distintos, sendo o isquêmico, que ocorre quando um coágulo sanguíneo bloqueia o fluxo, e o hemorrágico, causado pelo rompimento de vasos sanguíneos e, conseqüentemente, o derramamento de sangue na região encefálica (O'SULLIVAN; SCHIMITZ, 2010).

O AVE é classificado como a segunda maior causa de morte, com 11,8% do total de óbitos globais (VIRANI et al., 2021) e a primeira de incapacidade adquirida em todo o mundo (LAVADOS et al., 2007). Dados indicam que o AVE é a causa mais frequente de óbito na população adulta brasileira, sendo cerca de 10% dos óbitos totais, além de estar relacionado a 32,6% das mortes com causas vasculares e a 40% das aposentadorias precoces no país (LOBO et al., 2021). Uma pessoa sofre um AVE a cada 40 segundos nos Estados Unidos sendo que um em cada 20 óbitos é causado por este (VIRANI et al., 2021).

A incapacidade adquirida por vítimas de AVE possui uma variedade de efeitos sob a ação motora, dependendo principalmente da topografia lesional (MINELLI; FEN; MINELLI, 2007). Cerca de 50 a 70% dos indivíduos adquirem independência funcional após a ocorrência do AVE, enquanto de 15 a 30% permanecem incapacitados, e 20% necessitam de algum tipo de cuidado especial. Observou-se ainda uma influência negativa de cerca de 15% nestes efeitos oriundos de depressão pós-AVE (PAOLUCCI et al., 2019; ROSAMOND et al., 2008).

Os sinais e sintomas clínicos mais característicos do AVE, principalmente quando oriundos do infarto da Artérias Cerebrais Média (ACM), são denominados de hemiplegia, quando ocorre a ausência de movimento voluntário no hemisfério contralateral à lesão, e hemiparesia, quando há a diminuição do movimento voluntário neste hemisfério (LAVADOS et al., 2007; VIRANI et al., 2021).

Ao observar os dados sobre o AVE apontados é possível notar a evidente importância das pesquisas relacionadas ao tratamento e a prevenção do AVE dado o seu grande impacto social e econômico. Sendo assim, inúmeros pesquisadores têm buscado meios de facilitar e melhorar o processo definido como reabilitação.

Outra doença neurológica com alto impacto sobre a sociedade é a Paralisia Cerebral (PC), capaz de afetar a capacidade motora de indivíduos logo nos primeiros anos de vida, registrando, em média, três casos a cada 1000 crianças nos Estados Unidos (MACLENNAN; THOMPSON; GECZ, 2015), totalizando cerca de 8 a 10 mil casos por ano e sendo considerada a principal causa de incapacitação infantil (MICHAEL-ASALU et al., 2019).

Na Medicina e na Fisioterapia, a reabilitação dedica-se ao processo de tratamento de doenças crônicas, lesões e sintomas residuais de causa neurológica podendo ser originadas desde o nascimento, ou adquiridas por acidentes ou causas naturais. Esta ciência busca exercer um processo dinâmico durante sessões de treinamento específicas de forma a gerar ganhos funcionais de forma global (BARBOSA, 2013).

O processo de reabilitação para o AVE é atualmente aplicado através de protocolos desenvolvido por profissionais da saúde, fisioterapeuta e ou terapeuta ocupacional, e potencialmente aumentados pela utilização de ferramentas robóticas (CLAFLIN; KRISHNAN; KHOT, 2015; DIAZ; GIL; SANCHEZ, 2011; HOBBS; ARTEMIADIS, 2020). A robótica se inclui nesta área atuando sobre grande variedade de processos, desde a facilitação dos treinamentos de forma passiva assistida, até o apoio em procedimentos médico-cirúrgicos, onde destaca-se a compensação das capacidades residuais com o objetivo de prover ganhos funcionais às vítimas de AVE (HUANG et al., 2020; NUNES, 2012). Há ainda evidências de benefícios do uso de robôs no processo de tratamento de pacientes durante a pandemia de COVID-19 (KAISER et al., 2021),

Os ganhos funcionais, resultado do processo de reabilitação, possuem grande impacto aos pacientes em tratamento devido ao fato do resgate de funções que promovem independência funcional gerar uma melhoria direta na qualidade de vida destes indivíduos, através da melhora da autoestima, do gosto pela vida e pela sensação de superação de dificuldades (CAMARGOS et al., 2012; DOMÍNGUEZ-TÉLLEZ et al., 2020).

Há evidências das vantagens promovidas pela aplicação de estruturas robóticas na reabilitação, como a redução de custos com mão-de-obra ativa para as sessões de treinamento, a expansão da gama de exercícios realizados, auxiliando pacientes crônicos a manter a mobilidade através de terapia contínua passiva assistida (HOBBS; ARTEMIADIS, 2020; LUM et al., 2002).

Nesta tese é realizado o desenvolvimento de uma nova estrutura para reabilitação da marcha humana voltada para o tratamento de vítimas de AVE, PC e outras patologias que causam problemas na marcha rítmica, visando promover ganhos funcionais aos pacientes, incrementando a capacidade motora do(s) membro(s) inferior(es) lesionado(s).

Destaca-se que o desenvolvimento desta tese traz uma inovação para a reabilitação da marcha humana assistida por estruturas robóticas ao propor uma estrutura paralela de cinco graus de liberdade baseada em uma família de mecanismos com baixo acoplamento entre as coordenadas articulares.

Através do uso deste design inédito para a reabilitação da marcha humana, espera-se obter uma estrutura capaz de suportar o peso passivo dos pacientes e promover ações controladas sem a necessidade de atuadores de alta potência e sem comprometer o espaço de trabalho com singularidades.

A estrutura aqui proposta denomina-se HOPE-G, e é subdividida em 3 módulos principais sendo eles:

Módulo de Suporte Ativo de Peso Corporal: Este módulo consiste em uma estrutura robótica paralela com 5 graus de liberdade, sendo estes: três graus de translação tridimensional espacial e dois graus de rotação. O principal objetivo terapêutico deste módulo é promover treinamento passivo assistido à articulação pélvica e ao mesmo tempo suprimindo o peso corporal do paciente durante a sessão de treinamento. Dessa forma, a estrutura visa promover um treinamento mais eficiente capaz de facilitar melhoras no equilíbrio dos pacientes (LEFEBER; SWINNEN; KERCKHOFS, 2017).

- **Módulo de Movimentação por Esteira Bipartida:** Este módulo consiste em duas esteiras basculantes, posicionadas abaixo dos pés do paciente, promovendo treinamento passivo assistido para os movimentos da marcha humana de forma indireta. Este módulo é baseado no projeto original da estrutura MIT-Skywalker (SUSKO, 2015) e trará melhorias para promover uma maior facilidade de uso, melhorar o controle aplicado implementando técnicas de controle de impedância, e reduzir o custo de construção para tornar a estrutura mais acessível.
- **Módulo de Controle & Monitoramento Sem marcadores:** O monitoramento da marcha humana realizado pelo paciente durante a operação dos equipamentos é monitorado por um sistema que não necessita do posicionamento de marcadores específicos sob os membros inferiores do paciente aplicando apenas a câmera de jogos eletrônicos “Microsoft Kinect 2.0” para identificar e quantificar os movimentos do membro inferior. Além disso, baseado nos movimentos medidos o sistema é capaz de acionar e coordenar os atuadores dos demais módulos. Este módulo já foi desenvolvido e validado experimentalmente por meio de testes clínicos (SALIM, 2018).

O projeto da estrutura HOPE-G pode ser ilustrado inicialmente de acordo com o modelo conceitual representado na Figura 1.

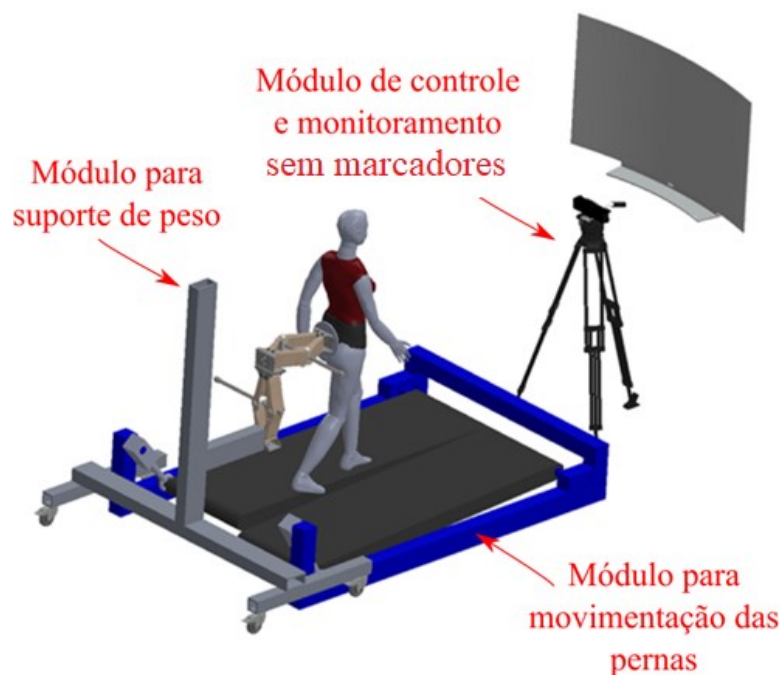


Figura 1 – Imagem conceitual inicial exemplificando os 3 módulos propostos para a estrutura HOPE-G.

O desenvolvimento da HOPE-G pode contribuir para o desenvolvimento da reabilitação robótica da marcha humana, especificamente para o tratamento de pacientes pós-AVE, trazendo uma nova perspectiva de suporte corporal ativo e explorando um conceito recente de movimentação passiva dos membros inferiores. Dessa forma, esta pesquisa contribui para o enriquecimento e avanço do estado da arte da reabilitação assistida por estruturas robóticas.

A principal inovação da estrutura HOPE-G concentra-se no módulo de suporte corporal ativo o qual foi desenvolvido baseando-se em uma cadeia cinemática fechada ainda não aplicada em estruturas de reabilitação e integrada a um jogo sério desenvolvido especificamente para este propósito. Tem-se que, as estruturas robóticas para reabilitação em esteiras com suporte passivo para o peso influenciam sobre o equilíbrio e movimentos da pélvis e do tórax durante a marcha (SWINNEN et al., 2015a, 2015b, 2015c), LOKOMAT (HESSE et al., 2003).

Dessa forma, o módulo suporte corporal ativo da HOPE-G poderá contribuir para o treinamento do equilíbrio dos pacientes durante as sessões de treinamento em conjunto com o módulo de movimentação por esteira bipartida, com uma esteira comum, ou diretamente sobre o chão.

Esta tese se inicia buscando o atual estado da arte das estruturas robóticas aplicadas à reabilitação da marcha humana na literatura da área. As estruturas atuais são classificadas segundo as suas principais características e os principais resultados são apontados para cada caso em destaque. Estas observações são então levadas em consideração no processo de desenvolvimento da estrutura HOPE-G.

Com base nas principais informações levantadas da revisão bibliográfica, Capítulo II, o desenvolvimento do novo sistema robótico procede com a modelagem matemática dos dois primeiros módulos: Módulo de Suporte Ativo de Peso Corporal e Módulo de Movimentação por Esteira Bipartida. Os módulos foram modelados a partir de modelos geométricos inversos das respectivas estruturas realizando uma posterior análise de singularidade para garantir o correto funcionamento no espaço de trabalho a ser aplicado.

A partir dos modelos matemáticos obtidos, Capítulo III, pode-se deduzir os carregamentos dinâmicos de ambos os módulos. Estes dados são então implementados em modelo computacional capaz de representar o sistema físico. A partir deste modelo computacional um processo iterativo de Simulação & Otimização é aplicado em conjunto com um algoritmo de evolução diferencial com o objetivo de se obter as dimensões ótimas para atuadores e elementos mecânicos dos módulos.

Após obtidas as dimensões, um protótipo do suporte ativo de peso corporal foi construído, e a partir das características deste foi desenvolvido um sistema de controle aplicando microcontroladores e utilizando um sistema de otimização meta-heurístico para ajustar os parâmetros dos controladores, conforme descrito no Capítulo IV.

De posse de um sistema controlado, a pesquisa segue com o desenvolvimento de um jogo sério, Capítulo V, específico para a estrutura e integrado com o sistema de controle do suporte ativo. Com a operação em conjunta de ambos os dispositivos, foi possível desenvolver um sistema de monitoramento para auxiliar os pacientes durante a operação do equipamento apenas quando necessário.

No Capítulo VI, realizou-se um teste piloto com a participação de pessoas saudáveis onde foi possível testar a integração entre a estrutura e o jogo sério, avaliar o conforme e a facilidade de uso do equipamento e elaborar recomendações para melhorias estruturais na estrutura.

Finalmente as conclusões e recomendações para trabalhos futuros são apresentados no Capítulo VII.

CAPÍTULO II REVISÃO BIBLIOGRÁFICA

A primeira etapa do projeto dedica-se a buscar a literatura recente referente ao estado da arte das estruturas de reabilitação da marcha humana, bem como as metodologias que suportam o desenvolvimento de todas as etapas das estruturas contempladas neste projeto.

Dessa forma, este capítulo contém os principais estudos e metodologias aplicados ao longo do desenvolvimento de todo o projeto. A revisão foi realizada utilizando o mecanismo de pesquisa dedicado a artigos acadêmicos do Google, priorizando artigos publicados entre os anos de 2017 e 2022. As pesquisas foram realizadas em inglês, utilizando palavras-chave relacionadas ao conteúdo da tese. As principais palavras-chave utilizadas foram:

- Human gait rehabilitation
- Robotic rehabilitation
- Serious games rehabilitation
- Robot lower limb rehabilitation

2.1. Cinesiologia da Marcha Humana

A marcha humana é um conjunto de ações motoras orquestradas por um indivíduo capaz de provocar a sua locomoção, ou seja, deslocar o seu próprio corpo no espaço terrestre. O estudo da marcha humana busca a caracterização mecânica e funcional da habilidade humana de andar (SACCO; TANAKA, 2008).

Esse conjunto de movimento é o mais comum de todos os movimentos humanos, composto por diversos eventos elétricos e mecânicos disparados pelo corpo gerando o deslocamento ao longo do ambiente.

As definições apresentadas ao longo desta sessão são baseadas nos livros de Sacco e Tanaka (2008) e Lobo da Costa e Serrão (2010).

Sob o ponto de vista da Cinesiologia a marcha é descrita por uma sequência de ações bem definidas nas respectivas articulações dos membros inferiores. Há ainda o estudo dos elementos dinâmicos deste processo, a biomecânica da marcha, que leva em consideração a interação entre forças musculares e articulares com as forças externas tais como efeitos friccionais, gravitacionais, inércia e amortecimento.

Variáveis da Marcha

Embora existam características particulares da marcha em cada indivíduo, há elementos comuns que permitem a medição e interpretação do movimento através de variáveis. Estas podem ser classificadas como variáveis espaço-temporais que caracterizam os eventos dos membros ao longo da execução da marcha; cinemáticas; que lidam com as medidas cinemáticas dos membros

inferiores tais como deslocamento, velocidade e acelerações das articulações e centros de gravidade de cada elemento; e dinâmicas, que estudam os esforços e reações internos e externos nos membros durante a marcha.

Nesta tese apresenta-se uma breve descrição das variáveis espaço-temporais da marcha humana para embasar os conceitos descritos nos próximos Capítulos.

As principais variáveis espaço-temporais estudadas na marcha humana são:

- **Fases:** A marcha humana possui movimentos que podem ser classificados em fases com início e fim bem definidos. Durante cada uma destas fases os membros e articulações executam ações específicas e encadeadas para resultar no deslocamento do corpo.
- **Ciclo da marcha humana:** O conjunto de todas as fases da marcha, executadas sequencialmente, é definido como ciclo da marcha humana. Este ciclo é descrito na Figura 2.
- **Cadência:** A cadência é definida como a frequência do ciclo da marcha humana, ou seja, o número de ciclos executados por unidade de tempo.
- **Comprimento do passo:** A distância média percorrida por cada membro inferior durante a execução da marcha. Calcula-se o comprimento do passo para o membro inferior esquerdo e para o direito, conforme Figura 2 e Figura 3.
- **Passada:** Definida como o espaço percorrido pelo indivíduo durante a execução de um ciclo da marcha. A passada é medida a partir do ponto de contato de um dos membros com o solo até o próximo contato deste mesmo membro no início do próximo ciclo.

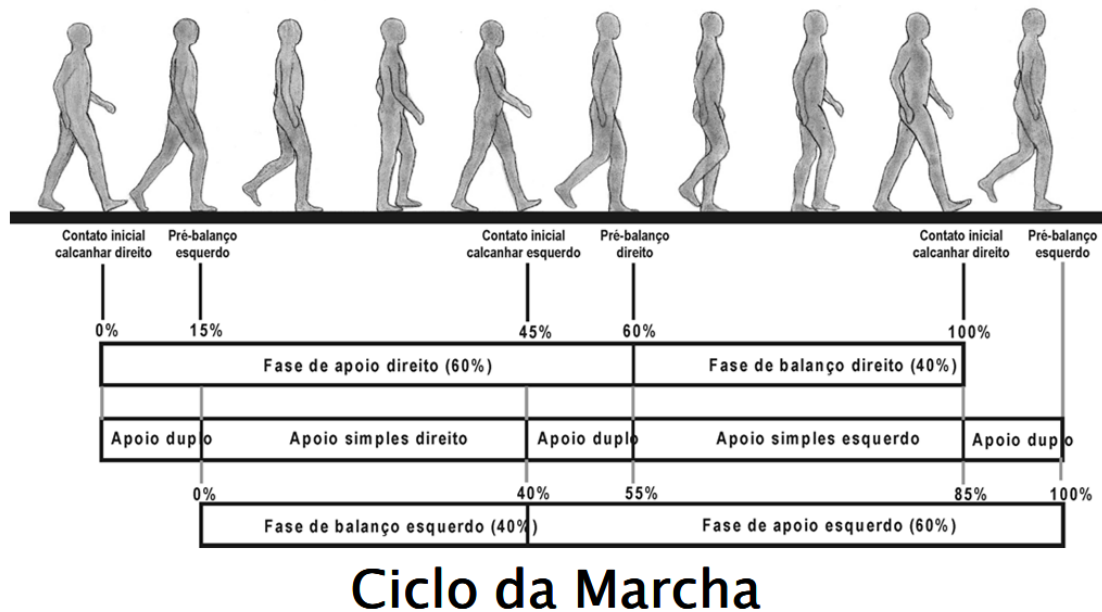
A seguir, os principais conceitos destas variáveis serão detalhados, com foco nas fases e no ciclo da marcha.

O ciclo da Marcha

Conforme definido anteriormente, o ciclo da marcha se divide em fases bem definidas do movimento.

Durante a execução da marcha, o movimento transita entre momentos em que o indivíduo está apoiado no solo com os pés, chamado de apoio duplo, e momentos em que um dos membros permanece em contato com o solo enquanto o outro realiza um movimento angular. Nesta segunda situação, um dos membros está em fase de apoio e outro em fase de balanço. Note, conforme Figura 2, que as fases de apoio e balanço direito/esquerdo se sobrepõem às de apoio duplo.

Variáveis Temporais



Ciclo da Marcha

Figura 2 – Desenho esquemático descrevendo o ciclo da marcha e suas respectivas fases (SACCO; TANAKA, 2008).

Em uma execução regular da marcha, cada membro permanece cerca de 60% do ciclo de cada membro em fase de apoio e cerca de 40% em fase de balanço. Porém, como a análise é realizada de forma paralela, o indivíduo permanece cerca de 30% do ciclo apoiado em ambos os membros, e 70% com um dos membros em balanço.

De forma similar, analisa-se os passos de cada membro e a passada marcha. O início da fase de apoio de um membro inicia-se no contato inicial do pé em dorsiflexão com o solo, e finaliza quando o pé se desconecta do solo em flexão plantar. Sendo assim, durante uma marcha regular a passada é menor que a soma do passo direito e passo esquerdo.

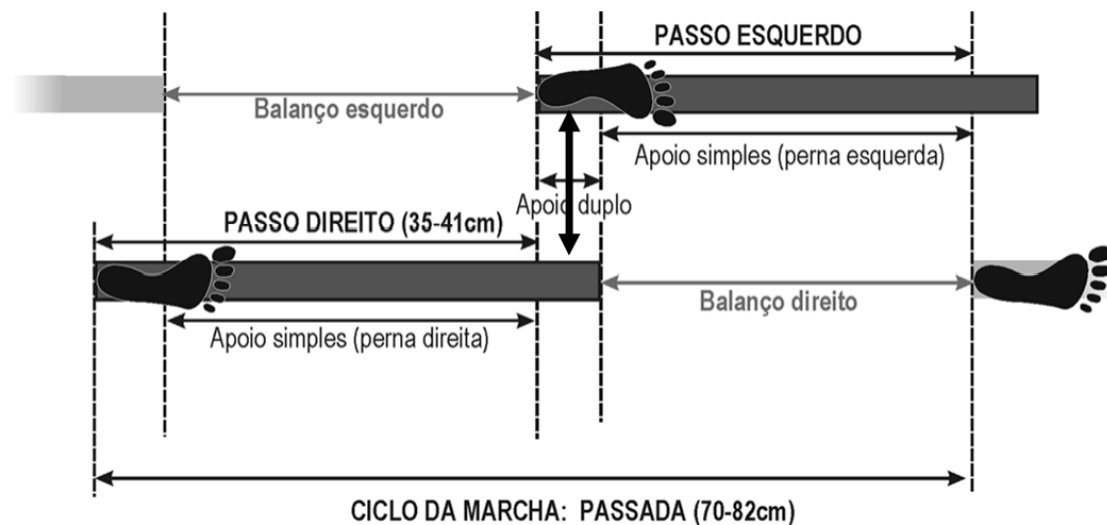


Figura 3 – Detalhamento do passo de cada membro e da passada da marcha humana (SACCO; TANAKA, 2008).

Irregularidades na marcha

Os movimentos descritos nas subseções anteriores descrevem características da marcha humana realizada por indivíduos saudáveis. Doenças neurológicas capazes de afetar funcionalmente a mobilidade dos membros inferiores como AVE e PC causam variações nos padrões descritos anteriormente.

A hemiparesia afeta diretamente a simetria da marcha, causando um desbalanceamento das atividades entre o membro parético e o não-parético, Figura 4. Um estudo comparativo com 13 participantes pós-AVE e 10 saudáveis demonstrou como se caracteriza este desbalanceamento (DYER et al., 2014).

Dessa forma, a marcha humana regular de indivíduos que sofreram AVE ou PC pode ser afetada e causando a necessidade de auxílio, de bengalas por exemplo, para ser executada de forma funcional.

Sendo assim, o estudo das características da marcha regular e da marcha parética são ferramentas úteis para a avaliação dos efeitos da fisioterapia em pacientes pós-AVE para avaliar ganhos musculares e funcionais.

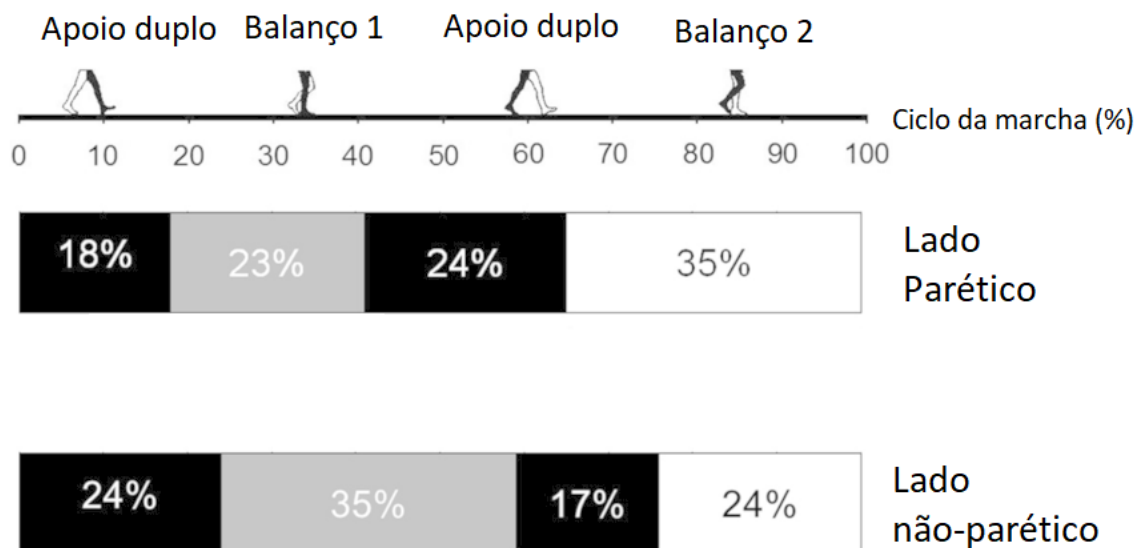


Figura 4 – Assimetria entre o membro parético e não-parético observada em pacientes pós-AVE (DYER et al., 2014).

Articulação sacroilíaca

A articulação sacroilíaca, também conhecida como articulação pélvica, é a conexão entre a base da coluna cervical (osso sacro) e das superfícies articulares dos ílios da bacia. Ela é uma articulação de baixa mobilidade e alta estabilidade, e possui um papel fundamental para o equilíbrio. Uma ilustração da articulação sacroilíaca é apresentada na Figura 5.



Figura 5 – Articulação sacroilíaca, também conhecida como articulação pélvica (Fonte: Google Imagens).

Dessa forma, a função fisiológica desta articulação está mais conectada à estabilidade do que com o excesso de mobilidade. Embora a amplitude de seus movimentos seja limitada, estes são importantes para garantir a estabilidade do corpo quando associados aos movimentos do quadril e da lombar. Esta mobilidade pode inclusive auxiliar funcionalmente a marcha de pacientes com irregularidades nesta (LIANG et al., 2014).

A articulação sacroilíaca possui 3 graus de liberdade de rotação com as seguintes amplitudes: 25° de rotação no plano transversal, 5° de rotação no plano sagital e 2° de rotação no plano frontal. Os planos de referência utilizados são representados na Figura 6.

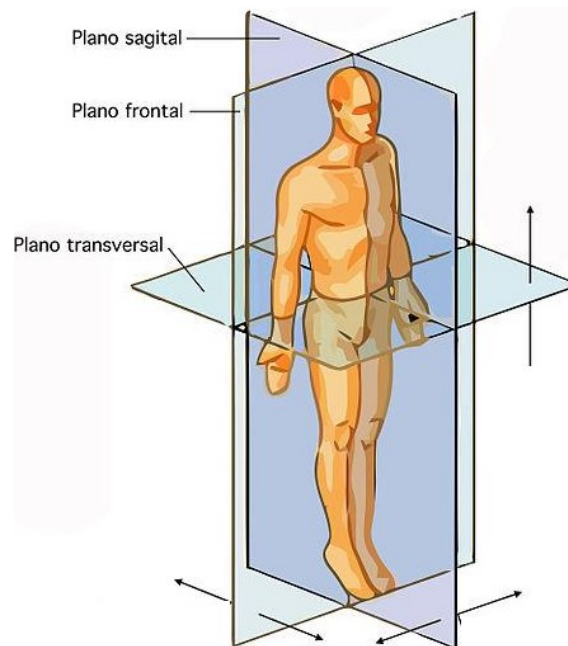


Figura 6 – Planos de referência da Anatomia Humana (Fonte: adaptada de <https://maestrovirtuale.com/planos-anatomicos-e-eixos-do-corpo-humano/>)

2.2. Estudo de manipuladores robóticos

Nesta tese foi proposta a utilização de um suporte ativo de peso do corpo atuante sobre os graus de liberdade da pélvis relevantes para a reabilitação da marcha humana, conforme descrito na Seção 3.2. Desta forma, objetiva-se encontrar uma estrutura capaz de realizar as 3 translações, e as rotações sob os planos frontal e transversal, restringindo apenas o plano sagital. Como o problema em análise exige 5 GDL foi realizada a revisão de possíveis estruturas capazes de atender aos principais movimentos da pélvis humana.

Para o melhor entendimento deste tópico primeiramente é apresentado uma breve revisão sobre os tipos de estruturas robóticas e seus componentes, nomenclaturas e singularidades.

Elementos de uma estrutura robótica

Toda estrutura robótica consiste em um sistema multicorpos composto por corpos rígidos (cuja deformação pode ser desprezada), articulações ativas e/ou passivas e um elemento terminal (SCIAVICCO; SICILIANO, 2005), conforme Figura 7.

Um corpo rígido livre no espaço possui ao todo seis GDL, sendo eles: três GDL de translação; e três GDL de rotação. Quando dois ou mais corpos são unidos por uma articulação, esta restringe a movimentação destes corpos em um ou mais GDL (SCIAVICCO; SICILIANO,

2005). As articulações ativas são as responsáveis pelo controle motor da estrutura e suas posições são denominadas coordenadas articulares.

O elemento do robô responsável por executar a sua função/tarefa é chamado de Elemento Terminal. Este elemento é o que recebe a ferramenta aplicada para o propósito do robô, no caso de um equipamento industrial, ou o que efetivamente conecta ao membro de um paciente, no caso de uma estrutura de reabilitação. A posição e orientação deste elemento no espaço são definidas como coordenadas operacionais.

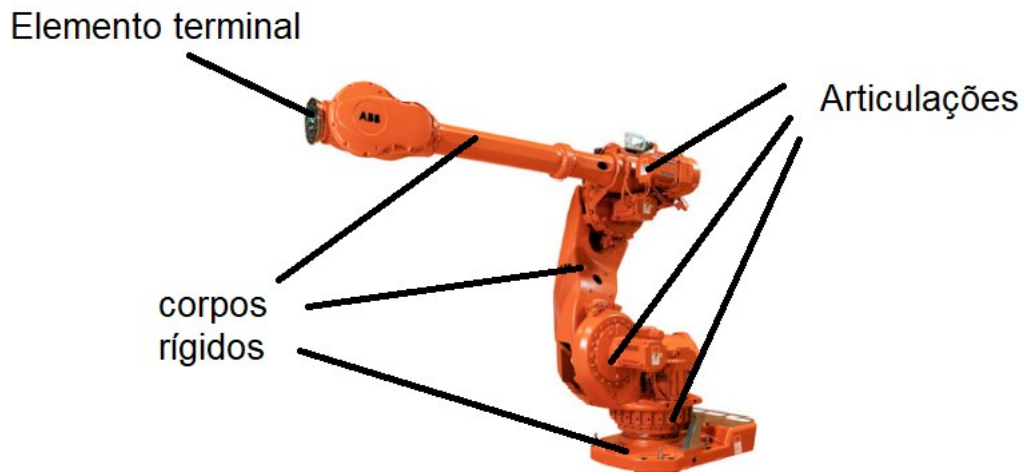


Figura 7 – Elementos fundamentais de uma estrutura robótica (SCIAVICCO; SICILIANO, 2005).

Estruturas Seriais e Estruturas Paralelas

No estudo de estruturas robóticas há uma classificação para organização dos corpos e suas conexões, onde estas são denominadas estruturas seriais, também definidas como de cadeia cinemática aberta, ou estruturas paralelas, equivalentes a cadeia cinemática fechada.

Considera-se uma estrutura como serial quando o número de GDL da estrutura é igual ao número de articulações ativas da estrutura (SCIAVICCO; SICILIANO, 2005). Estes tipos de estruturas são mais comuns em aplicações industriais e constantemente referidos como “braço robótico” por terem uma morfologia que lembra um braço humano. Um exemplo comum de estrutura serial é apresentado na Figura 8(a).

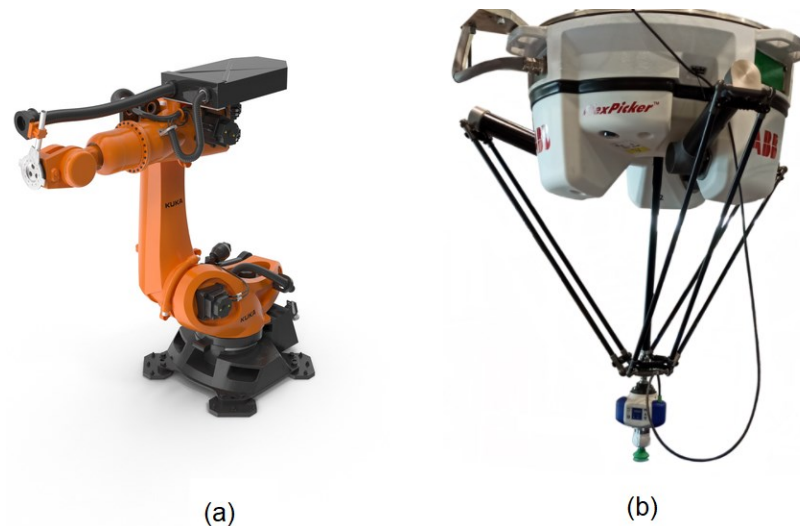


Figura 8 – (a) Exemplo de uma estrutura robótica serial; e (b) exemplo de uma estrutura robótica paralela (fonte: KuKa Robotics® e Wikipedia Commons).

O número de GDL de uma estrutura paralela depende da geometria aplicada e do número de articulações passivas e ativas (SCIAVICCO; SICILIANO, 2005). Este número pode ser determinado de acordo com a Eq. (1). Um exemplo de estrutura paralela é apresentado na Figura 8(b).

$$(\eta + 1)n_{GDL} + \eta = C_1 + C_2 + \dots + C_k \quad (1)$$

Onde:

η → Número de movimentos independentes que um corpo sem qualquer vínculo pode executar dentro do seu espaço de trabalho.

n_{GDL} → Número de GDL da estrutura paralela

k → Número de conexões

C_k → Conexões da cadeia k .

Estruturas paralelas e seriais possuem um vasto campo de atuação, onde as respectivas particularidades de cada tipo orientam a melhor escolha. Estruturas seriais normalmente possuem um espaço de trabalho mais amplo, uma capacidade de carga mais reduzida, menor precisão para execução de atividades, não utilizam articulações passivas e trabalham em velocidades mais baixas (SCIAVICCO; SICILIANO, 2005).

Em comparação, a geometria paralela possibilita uma maior capacidade de carga, uma melhor precisão no elemento terminal, possibilita a utilização de articulações passivas, enquanto limita-se um espaço de trabalho reduzido (SCIAVICCO; SICILIANO, 2005).

A partir destas definições, são apresentados alguns conceitos fundamentais para a compreensão do processo de modelagem de estruturas robóticas utilizadas nesta tese.

Tipos de articulações

Os elementos mais importantes para se definir o comportamento de uma estrutura multicorpos como um robô são as suas articulações, isto é, os pontos de contato entre cada um dos corpos rígidos que o definem (SCIAVICCO; SICILIANO, 2005).

As articulações são definidas de acordo com o número de GDL permitidos por esta, e pela sua natureza (translação e/ou rotação). Os tipos de articulações utilizados em robôs são listados a seguir e representados na Figura 9.

- Prismática ou de translação (P): Permite um GDL de translação.
- Rótula ou de rotação (R): Permite um GDL de rotação.
- Universal ou cardam (U): Permite dois GDL de rotação.
- Cilíndrica (C): Permite um GDL de rotação e um GDL de translação.
- Esférica (S): Permite três GDL de rotação.

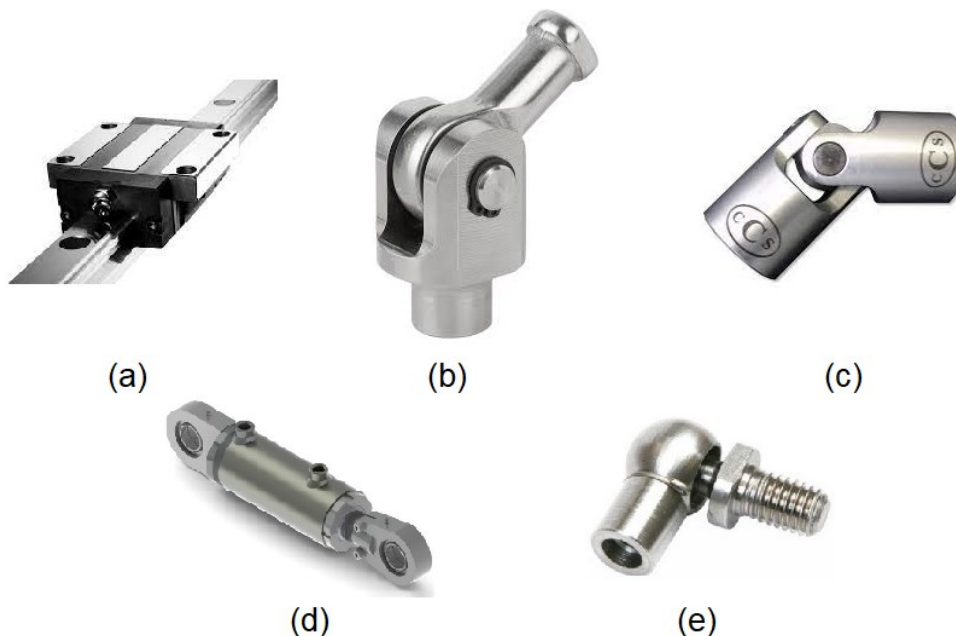


Figura 9 – (a) Guia linear, utilizada como articulação prismática; (b) rótula mecânica; (c) junta universal ou cardam; (d) junta mecânica com rolamento, utilizada como articulação cilíndrica; (e) junta esférica.

Estruturas robóticas são comumente definidas em função das configurações de suas articulações, especialmente estruturas paralelas. Para isso, há uma convenção definida pela Federação Internacional de promoção da Ciência de Mecanismos e Máquinas (IFTToMM) (BONEV, 2007).

Mecanismos de cadeia aberta, ou seja, estruturas seriais, são representados por uma sequência de caracteres que representam suas articulações, orientada da base ao elemento terminal. Por exemplo, um robô serial com 2 GDL de rotação e 1 GDL de translação pode ser definido como estrutura RRP, Figura 10(a).

Para o caso de estruturas paralelas, é comum ter sequências cinemáticas similares ao longo do mecanismo, as quais podem ser definidas como “pernas”. Quando o mecanismo possui pernas idênticas, estas podem ser agrupadas na nomenclatura. Por exemplo, a estrutura *Tripteron* (GOSSELIN, 2009) pode ser definida como 3-PRRR, pois possui 3 pernas idênticas conforme Figura 10(b).

Há ainda a necessidade, para estruturas paralelas, de se adicionar duas notações adicionais. Primeiro, como esta arquitetura permite articulações passivas e ativas, o caractere que se refere a uma ativa deve ser destacado em negrito ou sublinhado. Segundo, como apenas a sequência de articulações é insuficiente para se definir o número de graus de liberdade da estrutura, é importante deixar o número de GDL da estrutura explícito junto a sua nomenclatura, ou sempre apresentar um desenho esquemático para possibilitar a aplicação da Eq. (1).

Quando houver uma subcadeia em uma das pernas de uma estrutura paralela, é possível denotar esta ocorrência com o uso de parênteses.

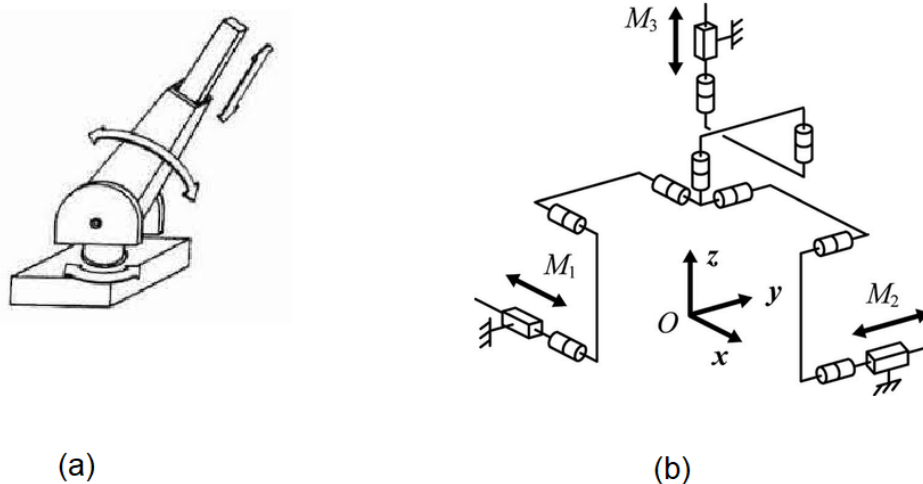


Figura 10 – Esquema de uma estrutura (a) serial RRP e; (b) paralela 3-PRRR de 3 GDL (adaptada de Gosselin 2009).

Modelagem de estruturas robóticas

Para se projetar e controlar uma estrutura robótica é necessário definir a relação entre as coordenadas operacionais, ou seja, as coordenadas do elemento terminal do robô e as coordenadas articulares, onde de fato se realiza a ação ativa em um robô (SCIAVICCO; SICILIANO, 2005).

Para exemplificar esse conceito considere um robô serial similar ao apresentado na Figura 8(a) cuja função seja realizar soldas em uma linha de montagem industrial. É muito mais natural definir a trajetória do robô em função do elemento terminal, onde está a tocha de soldagem, do que em função dos movimentos do motor de cada articulação.

O conjunto de equações que realiza a transformação de coordenadas operacionais e articulares é denominado Modelo Geométrico (SCIAVICCO; SICILIANO, 2005). Quando um modelo geométrico descreve a transformação de coordenadas articulares em coordenadas

operacionais, este é denominado Modelo Geométrico Direto (MGD). De forma semelhante, quando o modelo transforma coordenadas operacionais em coordenadas articulares este é denominado Modelo Geométrico Inverso (MGI).

Em outras palavras, o MGD permite obter a posição e orientação no espaço do elemento terminal dadas as posições das articulações das articulações ativas do robô. O MGI permite, dada a posição e orientação do elemento terminal, obter quais as posições das articulações ativas necessárias para esta configuração.

Uma representação gráfica deste conceito é apresentada na Figura 11 onde as coordenadas articulares q_1, q_2, \dots, q_n representam as articulações ativas da estrutura que realizam ativamente o movimento e as coordenadas operacionais $X_P, Y_P, Z_P, \theta_P, \varphi_P, \Psi_P$ representam os movimentos do elemento terminal a serem controlados.

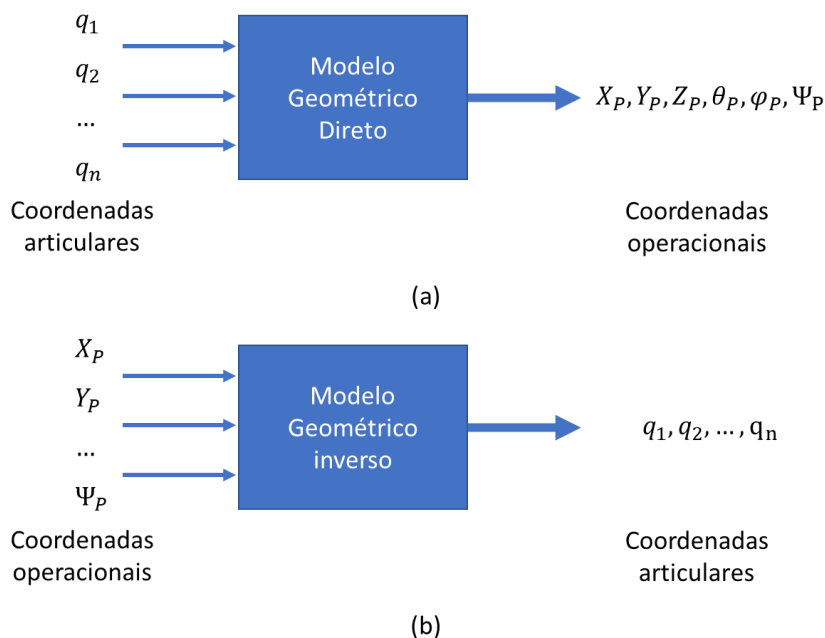


Figura 11 – Representação gráfica das transformações de coordenadas definidas no (a) Modelo Geométrico Direto e (b) Modelo Geométrico Inverso (SCIAVICCO; SICILIANO, 2005).

Singularidades

Existem situações em que uma determinada configuração de uma estrutura robótica pode causar o travamento ou a perda do controle do robô. Estas configurações são chamadas de singularidades e podem ocorrer em função da interação entre os elementos que compõem a estrutura. Para garantir o correto funcionamento de um mecanismo é de suma importância a realização de uma análise de mobilidade e de possíveis posições de singularidade.

Altuzarra et al. (2004) apresenta a definição de singularidade como uma impossibilidade de solucionar o problema cinemático matematicamente em determinados pontos de um mecanismo, o que pode ser compreendido fisicamente como pontos que, de forma local ou global,

alteram a mobilidade de um mecanismo levando-o à falta de controle de uma ou mais articulações durante seu ciclo de movimentação.

O método mais comum de determinação das possíveis configurações singulares de uma estrutura robótica é através da análise da chamada matriz jacobiana $[J]$, o qual será detalhado no Capítulo III.

Existem outros métodos para determinação de singularidades como por exemplo, a análise da matriz geométrica (HERNÁNDEZ et al., 2003) e a análise generalizada baseada em diversidades matemáticas (ZLATANOV, 1998).

2.3. Estruturas para reabilitação da marcha humana existentes

Conforme mencionado no Capítulo introdutório, o processo de reabilitação da marcha humana vem sendo beneficiado com o uso de estruturas robóticas capazes de assistir e melhorar os protocolos tradicionais de reabilitação dos membros superiores e inferiores.

Nesta sessão são descritas algumas das estruturas que compõem o atual estado da arte da reabilitação assistida por robôs terapêuticos.

Ao abordar a literatura existente para as estruturas robóticas aplicadas na reabilitação do membro inferior, pode-se identificar quatro tipos de estruturas já bem definidas, baseadas no estudo de Diaz, Gil e Sanchez (2011) e Rodrigues (2017), sendo: estruturas reabilitadoras por (a) esteiras; (b) base programável para os pés; (c) base robótica móvel; e (d) base estacionária. A Figura 12 mostra um esquema ilustrativo de cada um destes tipos de estrutura.

A seguir são detalhadas algumas estruturas que exemplificam os tipos (a) e (c), isto é, aquelas que são projetadas para operar utilizando uma base móvel ou com o auxílio de esteiras. Optou-se por focar nestas duas categorias pois são as que possuem mais características semelhantes à estrutura HOPE-G, objeto de estudo desta tese.

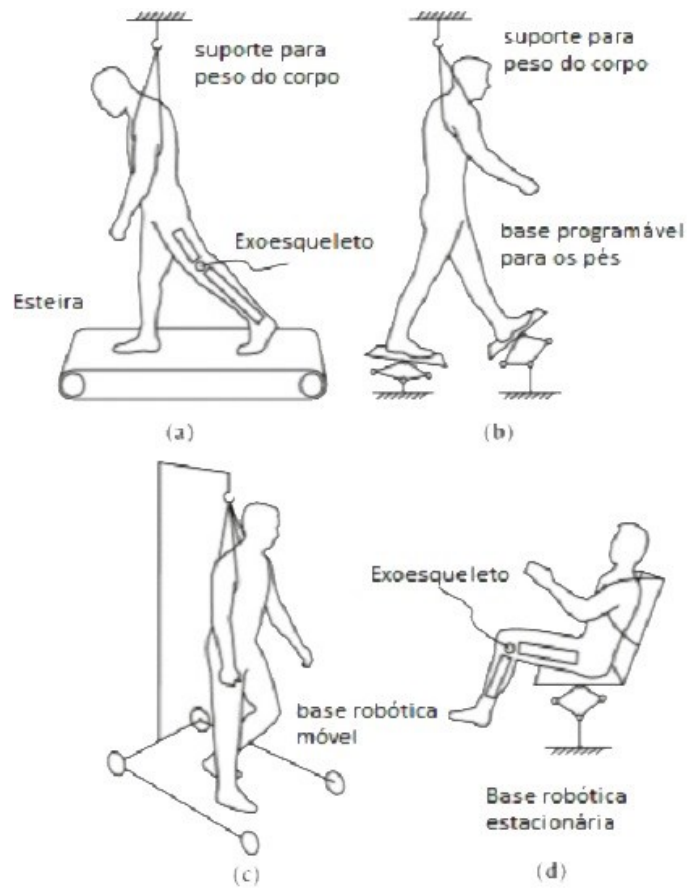


Figura 12 – Tipos de estruturas robóticas utilizando (a) esteiras, (b) base programável para os pés, (c) base robótica móvel e (d) base robótica estacionária (RODRIGUES, 2017).

Estruturas utilizando esteiras

Atualmente a principal estrutura robótica para reabilitação da marcha humana comercializada é a LOKOMAT, Figura 13, distribuída pela HOCOMA®. A estrutura foi inicialmente desenvolvida por Jezernik, Colombo e Morari (2004). Esta consiste em um exoesqueleto de quatro graus de liberdade composta por duas estruturas seriais acopladas aos membros inferiores do paciente controlado por um sistema de controle de impedância baseado na trajetória natural do ciclo da marcha humana.



Figura 13 – Imagem da versão comercial da estrutura LOKOMAT, distribuída pela Hocoma® (fonte: <https://www.hocoma.com/solutions/lokomat/>).

A LOKOMAT é uma das estruturas de reabilitação mais estudadas atualmente, com diversos estudos apontando para sua eficácia para reabilitação da marcha humana. Há evidências de melhoras no equilíbrio durante a marcha (BARONCHELLI et al., 2021) e na capacidade de movimentação de membros paréticos em pacientes pós AVE e na efetividade de atividades recreativas (Medvedev, 2019) e melhoras na marcha funcional em pacientes pós lesão da medula espinal (ALASHRAM; ANNINO; PADUA, 2021). Além disso, um estudo observou ganhos cognitivos e comportamentais no tratamento de pacientes com traumatismo craniano quando associado a ferramentas de Realidade Virtual (MAGGIO et al., 2020).

A estrutura denominada LOPES (VENEMAN et al., 2007) é um exoesqueleto desenvolvido para reabilitação da marcha com suporte de peso em esteiras. Esta consiste em um sistema de suporte de peso ativo com dois graus de liberdade atuando sobre a pélvis do paciente, e duas estruturas seriais, sendo uma para cada perna. Cada estrutura serial possui dois atuadores, um para o quadril e um para o joelho. Todas as articulações aplicam controle de impedância mecânica, permitindo que a interação entre paciente e estrutura seja bidirecional, ou seja, a trajetória a ser seguida pode ser comandada e ajustada tanto pelo robô quanto pelo paciente. Uma imagem da estrutura é ilustrada na Figura 14.

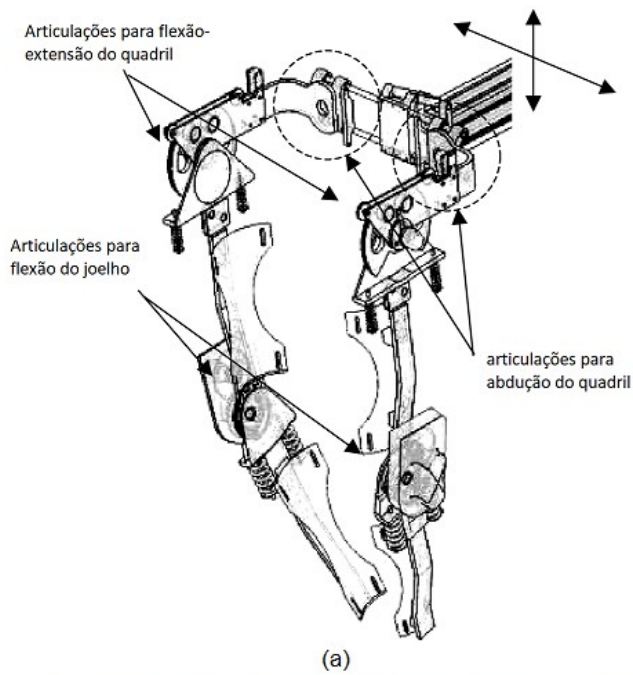


Figura 14 – (a) Esquema cinemático da estrutura LOPES; e (b) Imagens de um paciente utilizando a versão mais recente do equipamento (adaptada de <http://balance-fp7.eu> e VENENAM et al. 2007).

Visando explorar a movimentação da Pélvis, o módulo proposto por Roberts (2004) possui 4 graus de liberdade projetado com base em um mecanismo plano com 3 Graus de Liberdade (GDL) e um par cinemático prismático posicionado na vertical. Este módulo promove o suporte ativo ao peso do paciente, se ajustando lateralmente, fornecendo propulsão frontal, prevenindo quedas e tornando o movimento dos exercícios mais naturais. A montagem do módulo é apresentada na Figura 15.

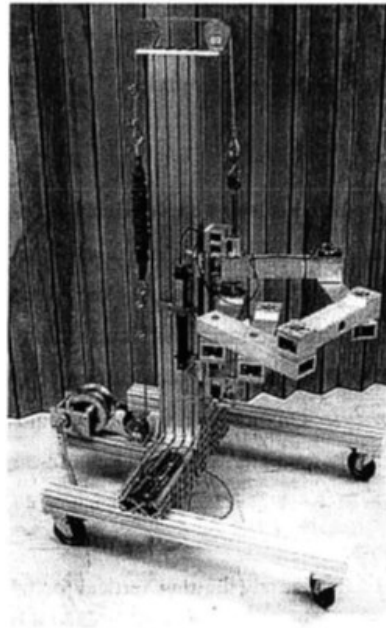


Figura 15 – Montagem do suporte para peso ativo proposto por Roberts (2004) (Fonte: Roberts, 2004).

Há outros estudos em estruturas seriais atuando como exoesqueletos para tratamentos assistivos, como por exemplo a estrutura de desenvolvimento aberto proposta por Li, Chen e Fan (LI; CHEN; FAN, 2017). O robô desenvolvido para operar sob esteiras possui quatro graus de liberdade com atuadores posicionados para auxiliar a movimentação das articulações do quadril e joelho de ambos os membros inferiores com ajustes para as dimensões de cada paciente. A estrutura é apresentada na Figura 16.

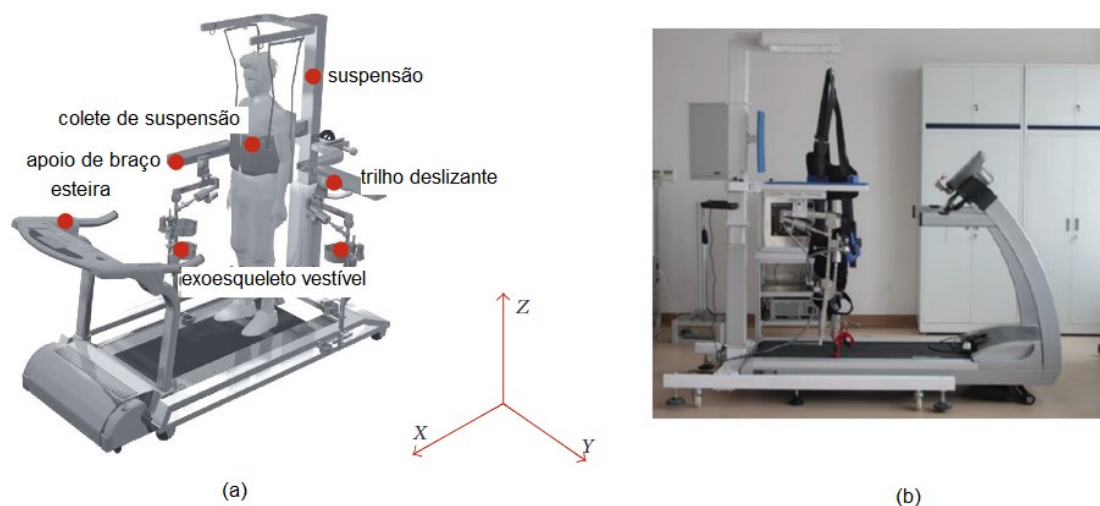


Figura 16 – (a) Desenho esquemático da estrutura; e (b) protótipo desenvolvido para a estrutura de reabilitação assistida por exoesqueleto (LI; CHEN; FAN, 2017).

O projeto foi submetido a testes clínicos iniciais, e obteve resultados positivos para participantes com diferentes níveis de mobilidade. Os pesquisadores acreditam que o design livre

de patentes apresentado poderá contribuir para a redução de custos para esta estruturas e outras similares pela redução de custos com propriedade intelectual.

A estrutura “*Robotic Gait Rehabilitation*” (RGR) (PIETRUSINSKI et al., 2010) consiste em uma estrutura para reabilitação em esteiras com suporte ativo para o peso do paciente, desenvolvida com o objetivo de auxiliar tratamentos de desvios de marcha secundários como assimetrias no ciclo da marcha e no passo de cada membro. A estrutura visa atender pacientes em tratamento pós-AVE.

A estrutura se acopla à Pélvis do paciente por meio de uma órtese conectada a atuadores eletromagnéticos, e promove correções posturais ao longo da marcha conforme a necessidade do usuário. Uma imagem representando a montagem desta estrutura é apresentada na Figura 17.

Outra estrutura similar é a denominada “*Pelvic Assist Manipulator*” (PAM) (AOYAGI et al., 2004; REINKENSMEYER et al., 2006), a qual foi desenvolvida para exercícios em esteiras com suporte para o peso em conjunto com um atuador paralelo para os membros inferiores denominado “*ARTHuR*”(REINKENSMEYER; WYNNE; HARKEMA,2004). Nesta estrutura, a atuação é feita utilizando 2 subsistemas pneumáticos organizados em uma estrutura de tripé, medindo e controlando a posição da pélvis do paciente durante os passos da marcha sobre a esteira. Um esquema da estrutura é apresentado na Figura 19.

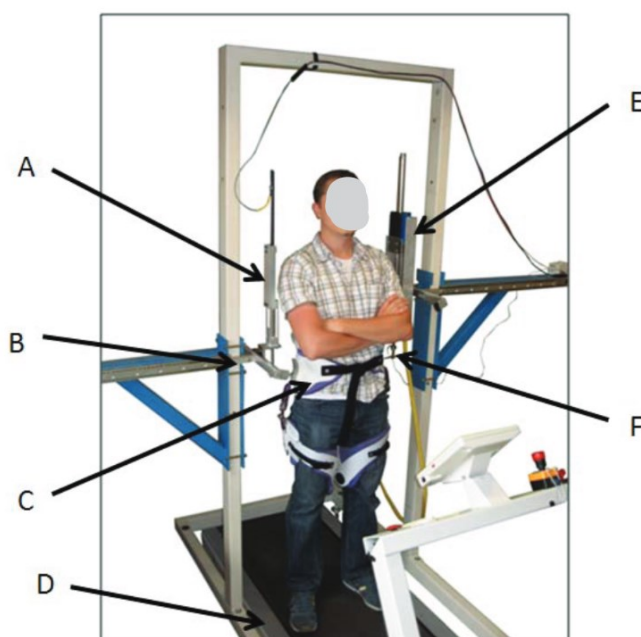


Figura 17 – Estrutura RGR Trainer: A- Potenciômetro linear, B – guias lineares, C – Braço pélvico, D – esteira, E – Servo motor/atuador linear, F – Célula de carga e articulação esférica (PIETRUSINSKI et al., 2010).

A estrutura proposta por Salguero-Beltran et al. (2012) denominada “*Ischiatic Body Weight Support System*” (IBWS) caracteriza-se por utilizar um atuador assistivo cartesiano em um suporte pélvico, de forma a permitir o estudo da marcha humana do paciente. Este suporte

ativo pode ser comparado a uma estrutura serial, capaz de executar todos os movimentos pélvicos ao longo da marcha humana, conforme ilustrado na Figura 18.

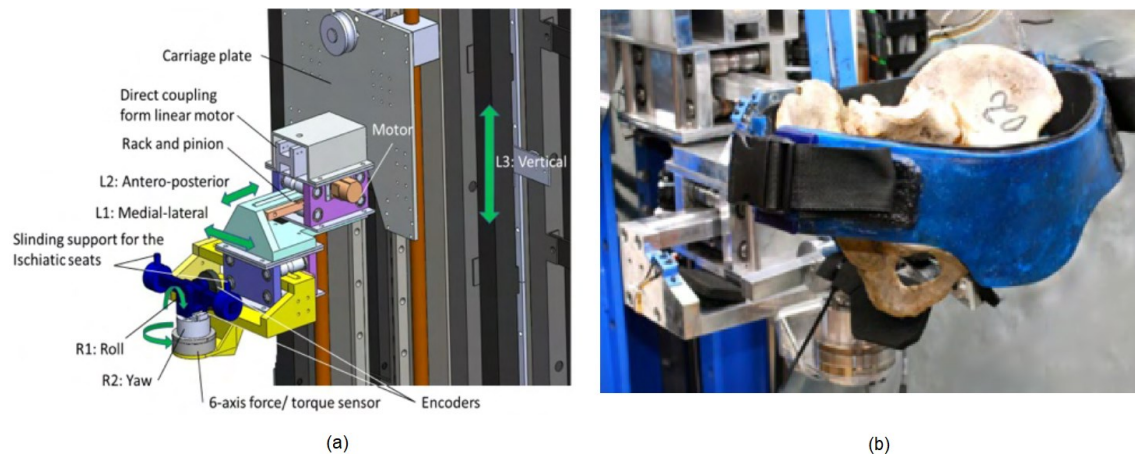


Figura 18 – (a) Esquema de montagem da estrutura serial IWBS para atuar na pélvis; e (b) Articulação pélvica humana totalmente suportada pelo socket da IWBS (SALGUERO-BELTRAN et al., 2012).

Os testes preliminares desta estrutura demonstraram que a arquitetura elétrica e de software proposta é capaz de promover correções de equilíbrio durante as fases de balanço de cada membro no ciclo da marcha.

Uma das principais referências para este trabalho é a estrutura denominada MIT Skywalker (GONÇALVES et al., 2017; GONÇALVES; HAMILTON; KREBS, 2017; SUSKO, 2015; SUSKO; SWAMINATHAN; KREBS, 2016), Figura 20. Esta estrutura pode ser classificada como uma estrutura mista do tipo (a) e (b), pois possui uma esteira bipartida basculante, combinando o princípio da marcha humana sob esteiras com o uso de base programáveis para os pés.

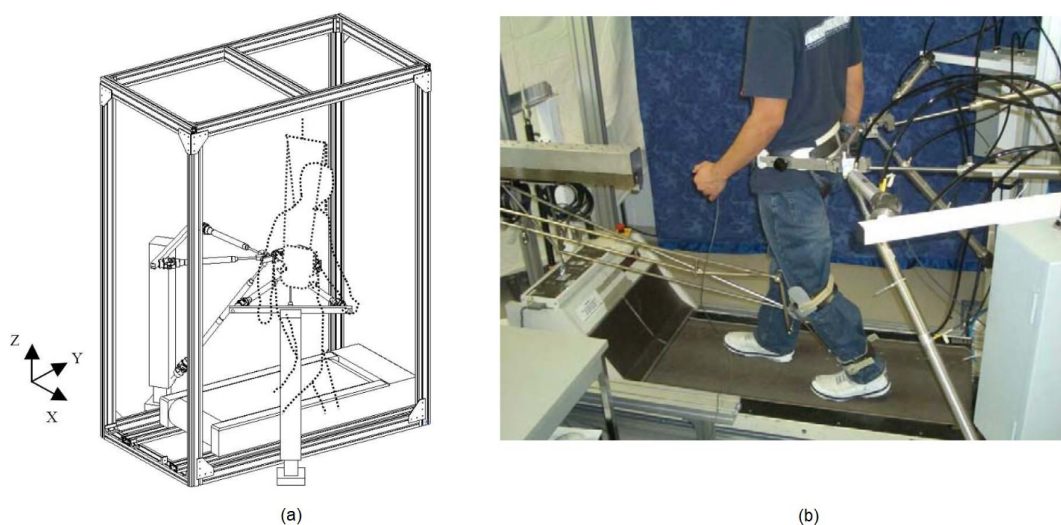


Figura 19 – (a) Esquema ilustrando a montagem da estrutura PAM; (b) Protótipo construído sendo utilizado em conjunto com a estrutura Arthur 2.0 (AOYAGI et al., 2004).

A estrutura baseia-se num sistema de transmissão de movimento para os membros inferiores de forma indireta, induzindo movimentos às articulações naturalmente com o feito gravitacional. Dessa forma, a estrutura visa promover poucas intervenções no processo de reabilitação, possibilitando a recuperação da passada de forma similar ao aprendizado natural da marcha.

O controle da estrutura é realizado por um sistema de visão robótica monitorando marcadores vestíveis infravermelhos posicionados nos membros inferiores do paciente.

Além do movimento basculante das esteiras, a estrutura também possui uma articulação rotativa ativa no plano frontal do paciente, o que permite o treino do equilíbrio durante as sessões de reabilitação.

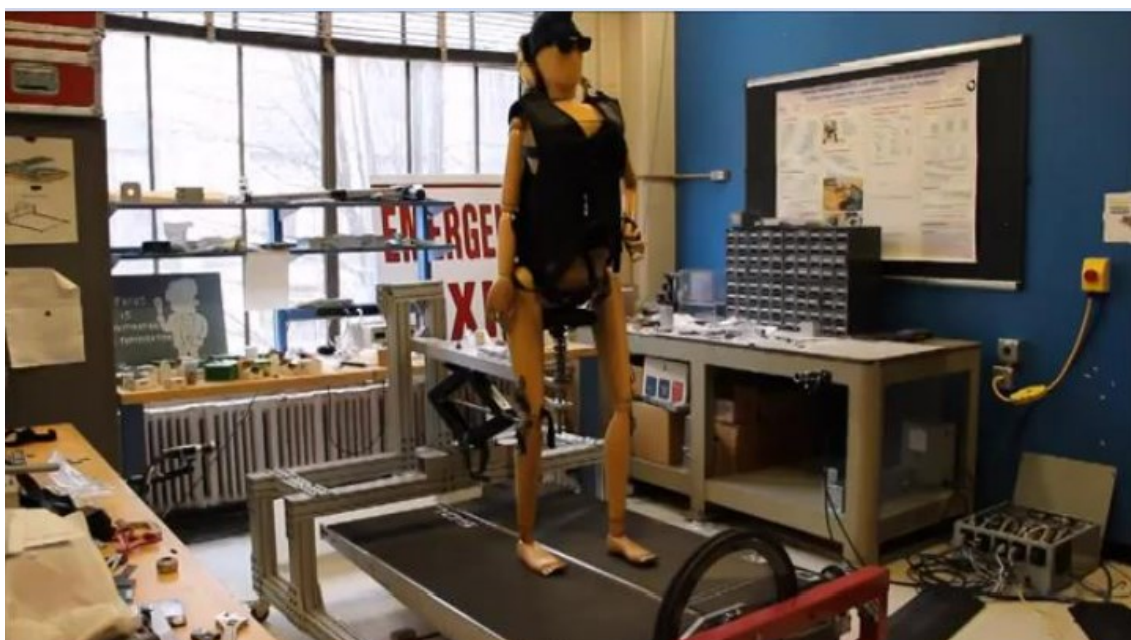


Figura 20 - Protótipo da estrutura MIT Skywalker (SUSKO, 2015).

Estruturas com base móvel

O uso de sistemas de suporte ativo de peso também é utilizado em estruturas para reabilitação de base móvel. Um exemplo é o exoesqueleto proposto por Goffe (2006), denominado ReWalk®. Este consiste em um sistema com dois braços robóticos que atuam diretamente sobre as pernas do usuário, e atuadores pélvicos utilizados para auxiliar o equilíbrio durante a marcha. A estrutura atualmente possui aplicações para fins terapêuticos, de reabilitação e para uso diário para usuários com mobilidade limitada. Uma fotografia deste dispositivo é apresentada na Figura 21.

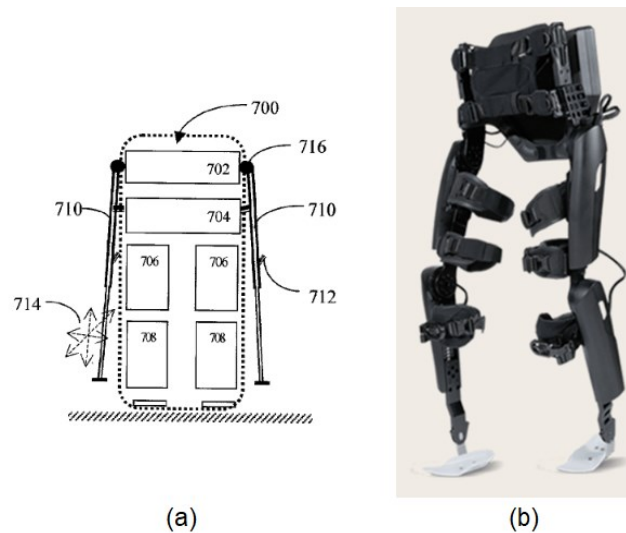


Figura 21 – (a) Desenho esquemático da patente e; (b) Imagem do produto da estrutura ReWalk (adaptada de Goffier 2006 <http://rewalk.com/>).

Outro exemplo de estruturas de base móvel que aplicam o conceito de atuação sobre a pélvis foi proposto por MUN et al. (2014), ilustrada na Figura 22, o qual também consiste em uma estrutura serial capaz de reproduzir todos os movimentos pélvicos montado sobre uma base móvel. Um estudo com esta estrutura mostrou mudanças significativas na marcha para diferentes níveis de supressão do peso corporal (MUN et al., 2017).

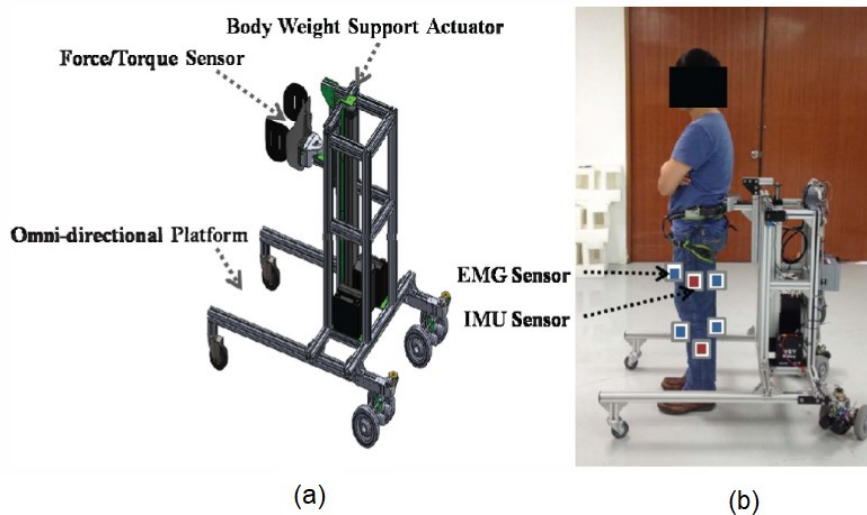


Figura 22 – Modelo tridimensional detalhado e; (b) protótipo construído da estrutura proposta por MUN et al. (2014).

Um estudo também observou as diferenças em estruturas que utilizam suportes baseados em assentos do tipo selim e em cintos de segurança para reabilitação assistida em esteiras bipartidas (KATAOKA et al., 2017). O design testado para o suporte é apresentado na Figura 23. Concluiu-se que suporte baseado em assento é capaz de manter a cinemática da marcha humana similar à não assistida por suporte para velocidades baixas, entre 0,5 e 1 km/h e alívios de até 33% do peso corporal.

Além disso, metade dos participantes deste estudo preferiram o uso do assento como suporte alegando mais conforto e menos restrição nos movimentos da marcha durante as sessões de treinamento.

Há ainda outros tipos que aplicam abordagens distintas para suporte de peso, como por exemplo a *AlterG*[®], que produz um efeito antigravitacional sobre a esteira utilizando uma câmara pressurizada abaixo do quadril do usuário, Figura 24. Um estudo mostrou ganhos de até 82% na velocidade de caminhada em crianças em tratamento de paralisia cerebral quando submetidas a sessões de treinamento com a redução de peso gerada pela *AlterG*[®] (AZIZI et al., 2017).

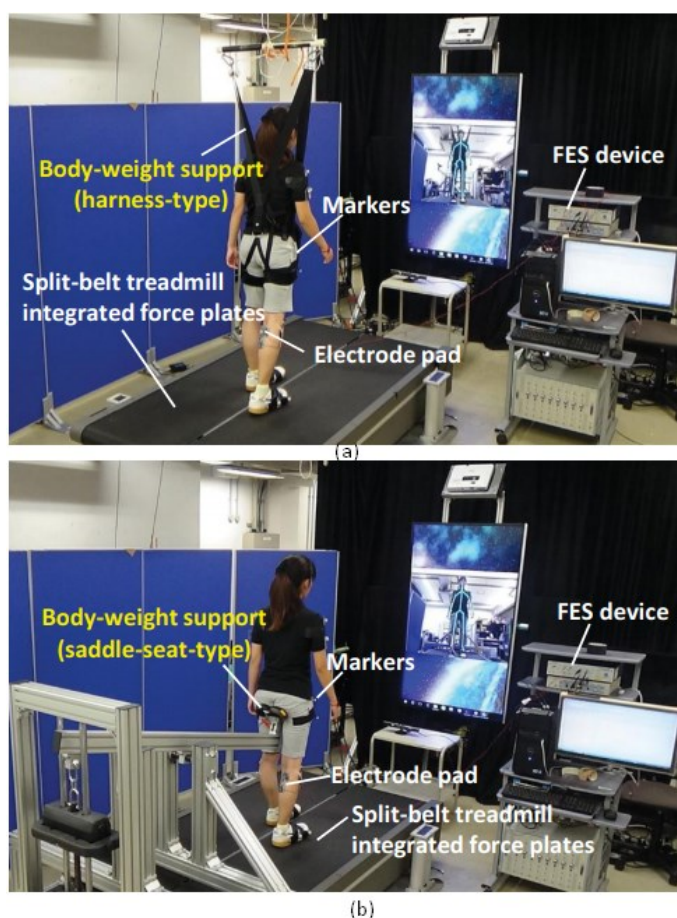


Figura 23 – Testes experimentais realizados por KATAOKA et al. (2017) para comparar os efeitos de suportes corporais baseado em (a) cintos e colete de segurança e (b) em assentos do tipo selim.

Uma das principais vantagens de dispositivos de reabilitação de base móvel em relação aos com suporte de peso sobre esteiras deve-se ao fato do treinamento promovido por essas ocorrer de forma mais natural. Há evidências de que o treinamento diretamente no solo promove uma maior adaptação neural da locomoção motora, relacionada principalmente ao controle motor do tornozelo (OCHOA; STERNAD; HOGAN, 2017).

Há também evidências de efeitos indiretos de suportes corporais passivos no equilíbrio de pacientes, que comumente são aplicados em estruturas de reabilitação da marcha do tipo (a), (b) e (c) (SWINNEN et al., 2015c, 2015b).



Figura 24 – Imagem comercial da estrutura AlterG® (fonte: <http://www.alterg.com>)

Sendo assim, este tipo de estrutura pode contornar os problemas de imobilização da pélvis e promover um melhor treino de equilíbrio aos pacientes quando comparado com estruturas que utilizam esteiras.

A importância da liberdade para o movimento da pélvis é ainda evidenciada ao se analisar o denominado “passo pélvico”, o qual consiste no movimento de rotação desta no plano transversal, o que aumenta o comprimento do passo durante a marcha humana reduzindo os movimentos verticais do centro de massa do corpo (LIANG et al., 2014).

Embora tenha sido evidenciado que a rotação transversal da pélvis pode estar defasada com o movimento do quadril durante a marcha em baixas velocidades e pequenos passos, este movimento pode auxiliar pacientes com baixa mobilidade para movimentos de flexão do quadril a incrementar o comprimento da passada, o que pode contribuir com o incremento da mobilidade destes (LIANG et al., 2014).

Há ainda evidências que a restrição do movimento da pélvis pode interferir significativamente também na largura da passada e nas rotações nos planos frontal e sagital do corpo durante a marcha devendo ser evitada para um treinamento mais eficiente (VENEMAN et al., 2008).

2.4. Estruturas robóticas paralelas existentes

As estruturas paralelas com 5 GDL se tornaram populares devido ao baixo custo e facilidade em serem aplicadas nas simulações de movimento, sendo necessário um estudo específico para a lei de formação destas estruturas, além de nomenclaturas específicas (KONG; GOSSELIN, 2005).

A estrutura apresentada por (LU et al., 2014) consiste em um manipulador de 5 graus de liberdade, sendo este uma estrutura 2UPS-SPR capaz de realizar todas as translações e duas rotações no espaço, estando os ângulos de Euler (SCIAVICCO; SICILIANO, 2005) da plataforma móvel acoplados matematicamente. Uma representação gráfica desta estrutura é apresentada na Fig. 25.

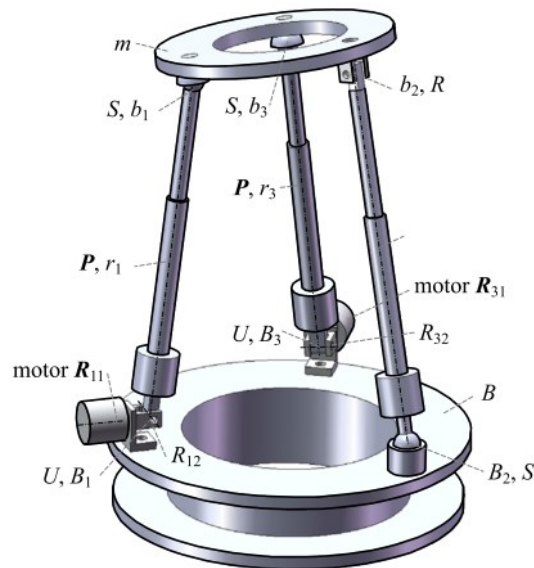


Figura 25 – Representação computacional da estrutura 2UPS-SPR (LU et al.,2014).

Esta estrutura apresenta bons resultados matemáticos para exploração de um espaço de trabalho com até 100° de rotação para cada uma das orientações no espaço. Porém, para a aplicação desejada a restrição dos movimentos sagitais necessita ser implementada no controle da estrutura.

O manipulador paralelo 5RRR(RR) (ZHU; HUANG; ZHAO, 2009) caracteriza-se por ser um tipo de estrutura simétrica, e por utilizar apenas pares cinemáticos de rotação. Esta estrutura possui uma construção simples, e articulações fáceis de montar. Porém possui diversas posições de singularidade, onde os autores já propõem metodologias para evitá-las. Um protótipo desta estrutura é ilustrado na Fig. 26.

Nesta estrutura, o primeiro conjunto RRR de cada perna é composto por articulações paralelas, responsáveis pela manipulação ativa da estrutura. O subconjunto (RR) é composto por articulações não-paralelas onde a linha de centro de cada uma delas converge para o centro de rotação da estrutura. Dessa forma, os subconjuntos (RR) são responsáveis pela variação da orientação da plataforma móvel.

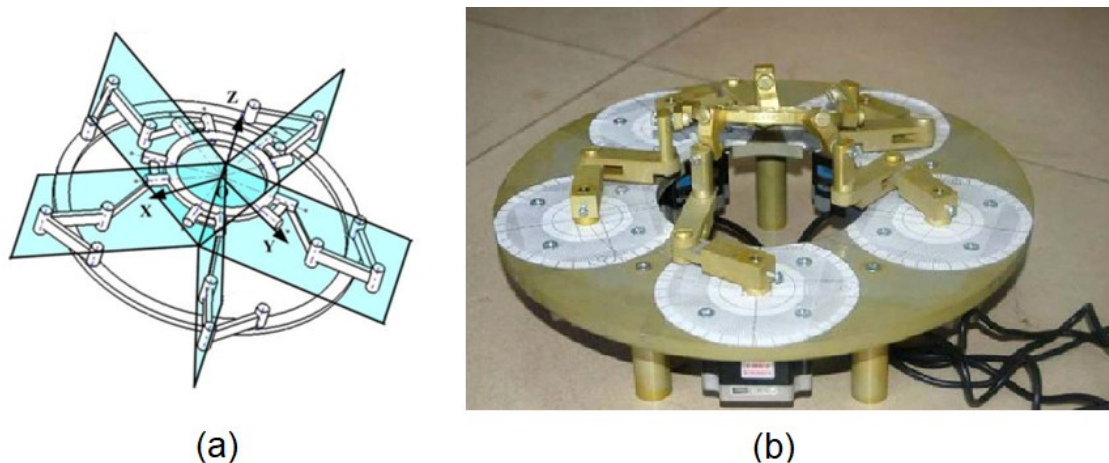


Figura 26 – (a) Esquema da composição da estrutura $5RRR(RR)$; e (b) protótipo construído (Adaptada de ZHU; HUANG; ZHAO, 2009).

Outro exemplo para estruturas paralelas de 5 graus de liberdade são os manipuladores do tipo $3P2R$ capazes de realizar 3 translações no espaço e duas rotações (MOTEVALLI; ZOHOOR; SOHRABPOUR, 2010). Existem várias propostas para este tipo de mecanismo sendo algumas destas estruturas esquematizadas na Fig. 27.

O uso de manipuladores paralelos como os apresentados anteriormente acarreta algumas dificuldades operacionais a serem consideradas tais como pequeno espaço de trabalho, rotação reduzida, rigidez global reduzida e alto nível de acoplamento entre os movimentos tornando os projetos sujeitos a serem específicos para uma determinada aplicação (XIE et al., 2016).

Pode-se observar também que manipuladores paralelos frequentemente apresentam como desvantagem o acoplamento de sua mobilidade o que dificulta a movimentação da plataforma em um grau de liberdade específico (MOTEVALLI; ZOHOOR; SOHRABPOUR, 2010).

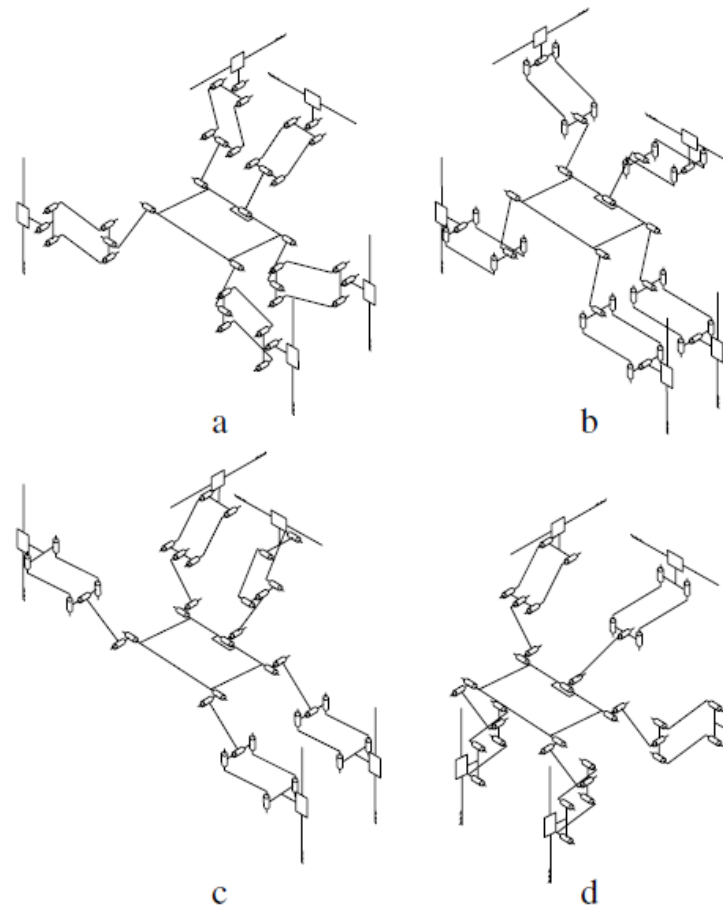


Figura 27 - Exemplos de manipuladores paralelos 3P2R: (a) Mecanismo 5-PRPaRR^a; (b) mecanismo 5-PRPaRR^b; (c) mecanismo 4-PPaRU^c+1-PPaRU^{x1}; (d) mecanismo 4-PPaRU^f+1-PPaRU^{x1} (MOTEVALLI; ZOHOOR; SOHRABPOUR, 2010).

Recentemente, novas metodologias foram propostas para a obtenção de manipuladores paralelos baseando-se em métodos de síntese de mecanismos (DING et al., 2015; JIN et al., 2018; MOTEVALLI; ZOHOOR; SOHRABPOUR, 2010). Com o auxílio de ferramentas computacionais, é possível aplicar estes métodos para a obtenção de um manipulador paralelo específico para a movimentação da pélvis durante os exercícios de reabilitação da marcha humana, conforme apresentado na Seção 3.2.

2.5. Controle de estruturas robóticas aplicadas na reabilitação

Uma das etapas mais importantes para o correto funcionamento de uma estrutura robótica de reabilitação é o controle dos atuadores. Enquanto o processo de modelagem e síntese dos mecanismos concentra-se em adequar as características mecânicas do equipamento de acordo com as necessidades de seu requerido processo terapêutico, a tecnologia aplicada ao controle de seus atuadores irá garantir que a operação atinja os resultados desejados.

A literatura traz diversas abordagens específicas para o controle de estruturas robóticas de reabilitação, onde destaca-se o uso de três delas: o controle de impedância mecânica; o aprendizado por demonstração; e a metodologia de “assistir quando necessário”.

Controle de impedância mecânica

O controle de impedância, por definição, consiste em ajustar os parâmetros de controle dos atuadores de uma estrutura robótica para manipular o comportamento da estrutura, o que em termos práticos resulta no controle das características mecânicas da estrutura, ou seja, da frequência natural ω_n e fator de amortecimento ξ . Dessa forma é possível regular a relação entre a força exercida e a posição do sistema e utilizar a malha de controle para aproximar o comportamento da estrutura a um sistema “massa-mola-amortecedor” com parâmetros conhecidos (HOGAN, 1984).

A aplicação desta metodologia em estruturas de reabilitação humana vem se consolidando como uma boa prática. O controle de impedância já foi aplicado para melhorar o controle motor do tornozelo durante a reabilitação da marcha humana, onde o estudo realizado demonstrou que o controle ativo da impedância do sistema durante a marcha resultou em melhor resultados dos participantes da pesquisa quando comparado a ausência de controle ou em parâmetros fixos (BLAYA; HERR, 2004).

A estrutura proposta por Hussain, Xie e Jamwal (HUSSAIN; XIE; JAMWAL, 2013; JAMWAL et al., 2017) também utiliza o conceito de controle de impedância mecânica em uma estrutura de reabilitação de dois graus de liberdade utilizando músculos pneumáticos para auxiliar a execução da marcha humana dos pacientes. O sistema varia os parâmetros mecânicos dos atuadores para promover assistência na movimentação dos membros inferiores dos pacientes apenas quando necessário estimulando o exercício ativo da marcha.

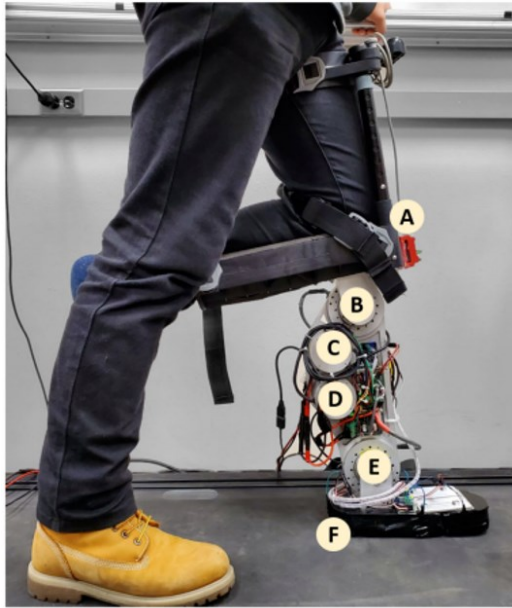
A abordagem mais comum desta metodologia inclui o uso de faixas de ganho predeterminadas, isto é, configurações pré-estabelecidas para os parâmetros que são acionadas de acordo com a relação força-posição ao longo da trajetória. Em estudos recentes, uma abordagem adaptativa para os parâmetros tem sido avaliada para não se limitar a valores predeterminados, e sim obter os parâmetros em tempo real de acordo com a necessidade de cada paciente (PEREZ-IBARRA et al., 2019).

No trabalho conduzido com o Anklebot (PEREZ-IBARRA et al., 2019), aplicou-se esta nova abordagem para a estrutura e novos testes foram realizados em participantes em processo de tratamento pós-AVE. A nova estratégia apresentou melhoras no desempenho da estrutura durante as sessões de treinamento permitindo uma maior variabilidade cinemática nos movimentos guiados pelos participantes sem comprometer a eficácia do tratamento.

Luna et al. (LUNA et al., 2020) apresenta um algoritmo de controle de impedância em tempo real para um exoesqueleto de reabilitação do membro inferior baseado em sinais eletromiográficos e nas características da passada regular da marcha humana. O controle da estrutura foi validado com testes experimentais realizados com um voluntário.

O controle de impedância também é aplicado em sistemas prostéticos como no estudo apresentado por Hong, Anil Kumar e Hur (2021) onde uma prótese transfemoral ajusta a

impedância mecânica da articulação do tornozelo estimando-se as fases da marcha humana através de um sensor inercial (IMU). Uma imagem do protótipo desenvolvido é apresentada na Figura 29.



- A. IMU (coxa)
- B. Drive harmônico (joelho)
- C. Motor DC (joelho)
- D. Motor DC (tornozelo)
- E. Drive harmônico(tornozelo)
- F. Sensor FSR (calcanhar)

Figura 28 – Protótipo de prótese transfemoral com controle de impedância ativo (HONG; ANIL KUMAR; HUR, 2021)

Aprendizado por demonstração

Outra metodologia presente em estruturas de reabilitação é a chamada “Aprendizado por demonstração”. A obtenção da trajetória a ser seguida pelos equipamentos é obtida observando os padrões executados pelo paciente e supervisionados por um profissional e replicando-os de forma automática (NUNES et al., 2011). A Figura 29 apresenta uma ilustração do conceito como exemplo.

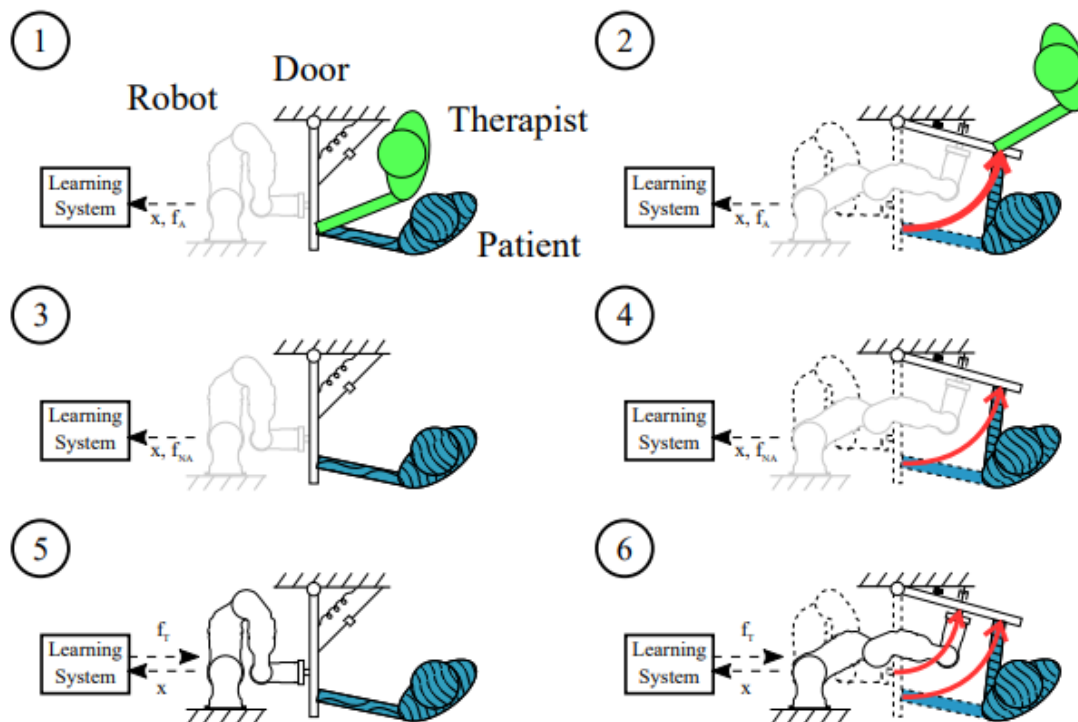


Figura 29 – Diagrama generalizado representando as etapas de Aprendizado por Demonstração. (1 e 2) Paciente e terapeuta movem um objeto, representado por uma porta com fechamento automático mecânico, enquanto o equipamento adquire dados sobre força e trajetória. (3-4) Em seguida, o paciente opera o sistema sozinho, e o robô obtém dados adicionais de força e trajetória. (5-6). O Sistema então utiliza os dados para construir um modelo preditivo e utiliza como dados de entrada o deslocamento do sistema, e gera saídas de forças conforme necessário para manter a trajetória conforme o modelo construído (FONG; TAVAKOLI, 2018)

O principal objetivo desta metodologia é abstrair, capturar e reproduzir de maneira simples os exercícios e princípios explorados durante as sessões de treinamento, e baseando-se na terapia passiva de movimentação contínua (NUNES et al., 2012).

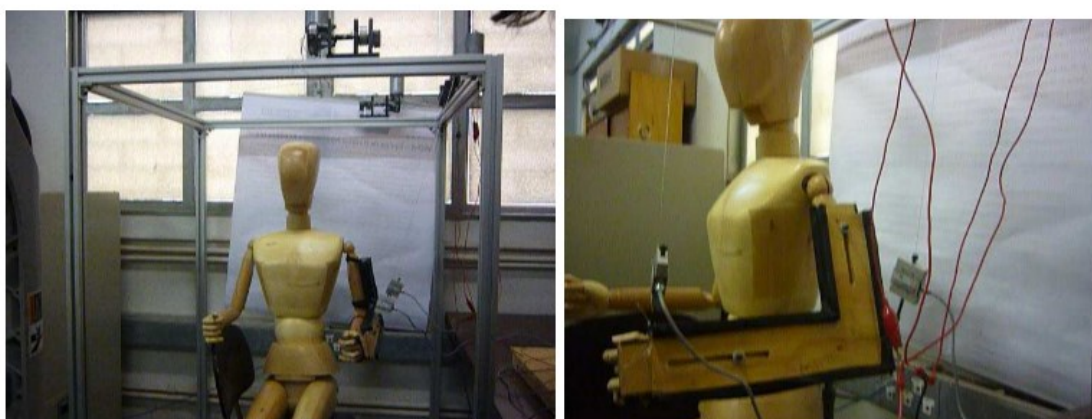


Figura 30 – Estrutura de reabilitação do membro superior atuada por cabos e controlada por aprendizado por demonstração (NUNES et al., 2012).

No trabalho apresentado por Nunes et al. (2012) foi desenvolvido um sistema de reabilitação do membro superior atuado por cabos, Figura 30, aplicando o controle de aprendizado

por demonstração. O sistema funciona em dois modos distintos: Aprendizado e Reprodução. Durante o “Aprendizado” o profissional da saúde pode realizar o movimento que ele deseja que seja reproduzido no paciente manipulando diretamente a órtese no membro do paciente. A estrutura, então, monitora a tensão dos cabos para determinar as direções de movimento e as rotações dos encoders para gravar uma trajetória. Uma vez obtida a trajetória, o modo “Reprodução” pode ser acionado e a estrutura executa a trajetória gravada de forma contínua.

O trabalho de Lauretti et al. (LAURETTI et al., 2017) apresenta uma melhoria à metodologia baseando-se em primitivas dinâmicas de movimento e kernels gaussianos. A estratégia foi avaliada utilizando a estrutura serial KuKa LWR4+, onde os resultados demonstram otimizar o consumo de memória das trajetórias e identificação dos padrões executados pelos usuários.

Há ainda uma abordagem que utiliza modelos de combinação gaussianos para analisar padrões aprendidos e prever a necessidade de assistência dos pacientes para executar a tarefa programada (FONG; TAVAKOLI, 2018). O estudo demonstra que a abordagem utilizada é capaz de auxiliar a movimentação dos pacientes enquanto executam as tarefas treinadas reduzindo o esforço repetitivo dos terapeutas durante as sessões de treinamento.

Assistir quando necessário

Uma das abordagens recentes relativa ao controle de estruturas robóticas é o uso de estratégias para limitar o comportamento ativo das estruturas, aplicando-o somente quando o usuário não consegue realizar o movimento com a trajetória desejada por seus próprios esforços. Esta abordagem é conhecida como “Assistir quando necessário”. Esta abordagem é comumente combinada com as mencionadas anteriormente (ALVES; CHAVES D’CARVALHO; GONÇALVES, 2019; HUSSAIN; XIE; JAMWAL, 2013)

Uma das formas de se obter o comportamento desejado é suprimindo a ação de controle da malha quando o erro do sistema está próximo de zero (ASL; NARIKIYO; KAWANISHI, 2019). Esta estratégia utiliza uma função matemática em série com a saída do controlador para criar uma “zona morta” onde a ação de controle aplicada aos atuadores é proporcionalizada de forma suave de acordo com o erro. Esta abordagem é aplicada no mecanismo de reabilitação combinada ReGear (RODRIGUES; MORAES; GONÇALVES, 2019) que utiliza um único grau de liberdade controlado por uma função “assistir quando necessário” para atuar apenas quando o paciente não pode executar a movimentação adequada. Uma imagem da estrutura é apresentada na Figura 31.



Figura 31- Estrutura de reabilitação combinada ReGear, construída utilizando a metodologia de controle “Assistir quando necessário” (RODRIGUES; MORAES; GONÇALVES, 2019).

O sistema monitora o desempenho dos pacientes através da pontuação realizada nos jogos e promove ações assistivas controlando a impedância do sistema e alterando as trajetórias quando necessário.

Um estudo analisou os efeitos da reabilitação assistida por estruturas robóticas na marcha humana em ambiente real após tratamento com estratégias baseadas em “Assistir quando necessário” (SRIVASTAVA et al., 2015). Resultados demonstraram que após 6 meses de tratamento os participantes pós-AVE apresentavam ganhos significantes e de longo termo oriundos das sessões de treinamento principalmente quanto a velocidade de marcha.

Esta metodologia também pode ser utilizada para estimar dinamicamente as intenções de movimentação da marcha em cada uma de suas fases para evitar que exoesqueletos limitem a capacidade de movimentação e realizem ações assistivas com mais precisão (QIU et al., 2021). Dessa forma, é possível personalizar as intenções de movimentação de acordo com cada indivíduo.

No trabalho de Qiu et al. (2021) um algoritmo baseado em aprendizado por demonstração analisa a dinâmica de cada fase da marcha e ajusta as previsões de trajetórias e torques de um exoesqueleto para o membro inferior em tempo real. Os primeiros testes experimentais demonstraram que o algoritmo é capaz de tornar as trajetórias e torques da estrutura mais suaves, além de reduzir o atraso devido a filtros em sensores.

2.6. Uso de jogos eletrônicos para reabilitação

O uso de jogos eletrônicos combinados com o processo de reabilitação assistida por estruturas robóticas vem se consolidado nos últimos anos, de forma que a maioria das estruturas disponíveis possuem integração com algum tipo de jogo para acompanhar as sessões de treinamento (HESSE et al., 2003; PINO; MÚNERA; CIFUENTES, 2022; SUSKO, 2015; VENEMAN et al., 2007).

Na área de reabilitação são utilizados jogos sérios. Estes são definidos como jogos eletrônicos capazes de interagir de forma terapêutica com os pacientes para treinamentos cognitivos e motores. O uso de jogos sérios na reabilitação de pacientes gera comprometimento com a terapia e contribui para neuroplasticidade dos procedimentos (PINO; MÚNERA; CIFUENTES, 2022).

Dessa forma, é possível apontar boas práticas já estudadas para o desenvolvimento e aplicação de jogos em novas estruturas de reabilitação (REGO; MOREIRA; REIS, 2010). Algumas características comuns de jogos sérios aplicados à reabilitação podem ser apontadas sendo elas: Área de Aplicação, Tecnologia de interação, interface do jogo, número de jogadores, gênero do jogo, adaptabilidade e assistência ativa.

A área de aplicação lida com a natureza do treinamento promovido pelo jogo sério, isto é, qual será o tipo de ação necessária dos pacientes para interagir com o jogo sério. De forma geral os jogos sérios podem ser classificados como jogos cognitivos, que normalmente dependem menos de estruturas ativas, e motores, constantemente associados a estruturas robóticas.

A Tecnologia de interação define como serão geradas as entradas e saídas para operação do jogo pelo paciente. As entradas estão ligadas ao uso de sensores de movimento (RODRIGUES; GONÇALVES, 2014), de eletromiografia de superfície (GUTIÉRREZ et al., 2020), de eletroencefalografia (ALCHALCABI; EDDIN; SHIRMOHAMMADI, 2017), processamento de imagem (POSTOLACHE et al., 2019), dentro outros.

A interface do jogo descreve como é o ambiente do jogo em si, que pode ser baseado nos modelos clássicos de jogos em 2D ou 3D (RODRIGUES; GONÇALVES, 2014), ou ainda jogos que utilizam tecnologias imersivas como Realidade Virtual (RV), Realidade Aumentada (RA) e Realidade Mixada (RM) (MUBIN et al., 2019).

Em geral, os jogos sérios aplicados para reabilitação humana com estruturas robóticas são desenvolvidos para apenas um paciente. Contudo, estudos recentes apontam bons resultados para abordagens multijogadores em estruturas de reabilitação em pacientes pós-AVE, onde fatores sociais contribuem para melhoras no desempenho dos pacientes (BAUR et al., 2018a, 2018b).

O gênero dos jogos sérios segue o padrão já conhecido na indústria com particularidades específicas para jogos de reabilitação onde, por exemplo, alguns gêneros se destacam. De forma

geral, observa-se jogos de avaliação de movimento, estratégia, simulação e plataforma (REGO; MOREIRA; REIS, 2010).

A adaptabilidade classifica os jogos que são capazes de ajustar a forma de interação com o usuário de acordo com a performance deste, isto é, alterar a dificuldade e fatores assistivos de acordo com a necessidade de cada paciente. Esse conceito está presente na indústria padrão de jogos e é estendido para jogos sérios de reabilitação.

A adaptabilidade dos jogos sérios pode ser classificada da seguinte forma: ausente, onde não existe nenhum tipo de ajuste do jogo para com o usuário; manual, quando parâmetros como controles e dificuldade podem ser ajustados manualmente no menu do jogo; ou automático, quando o jogo utiliza informações da performance do paciente para ajustar a melhor experiência. No caso de jogos combinados com estruturas ativas, é possível usar dados coletados em tempo real da performance do paciente para ajustar a experiência (RODRIGUES; MORAES; GONÇALVES, 2019; VENEMAN et al., 2007).

Finalmente, há jogos capazes de promover feedbacks assistivos para os jogadores durante as partidas. Jogos que possuem este tipo de capacidade estão constantemente associados a estruturas robóticas com sistemas de controles adaptativos, como por exemplo, o controle “assistir quando necessário” descrito no Capítulo IV (ASL; NARIKIYO; KAWANISHI, 2019; RODRIGUES; MORAES; GONÇALVES, 2019; SHI et al., 2020).

Há ainda outras boas práticas que podem ser mencionadas para guiar o desenvolvimento de jogos sérios para reabilitação de pacientes pós-AVE (MUBIN et al., 2020). As principais diretrizes são:

- Uso de elementos responsivos de forma que o jogador perceba bem o resultado de suas ações.
- Jogos para reabilitação devem reagir a falhas dos jogadores de forma adequada, para não gerar desmotivação. Dessa forma, deve-se evitar elementos que denotem vitória/derrota, falta de sucesso e/ou falta de alcance de objetivos.
- A dificuldade dos jogos deve estar sempre ajustada de forma adequada para a habilidade de cada jogador/paciente.
- O jogo sério deve fornecer feedbacks claros para o jogador/paciente acerca de suas ações deixando claro quando está correta ou não.

2.7. Aspectos selecionados para HOPE-G

Com base nos aspectos observados na literatura e apresentados neste capítulo, selecionou-se as características mais adequadas para aplicar à estrutura HOPE-G durante o desenvolvimento desta tese.

Definiu-se que a estrutura HOPE-G será uma estrutura baseada em reabilitação da marcha humana utilizando esteiras e suporte de alívio de peso ativo com foco na modelagem de um sistema de esteiras bipartidas e desenvolvimento de um suporte ativo capaz de auxiliar no treinamento do equilíbrio dos pacientes sem influenciar diretamente o comportamento natural da articulação pélvica.

Esta decisão baseia-se no fato de estruturas robóticas paralelas poderem utilizar articulações passivas e atuadores de menor potência, o que pode reduzir o custo global do equipamento. Optou-se pelo sistema de esteiras bipartidas basculantes para também se explorar o conceito de reabilitação através de ações indiretas e evitar ações não-funcionais no processo de reabilitação dos membros inferiores, conforme observado na revisão bibliográfica.

Para a modelagem matemática do suporte ativo será utilizada uma estrutura paralela de cinco graus de liberdade ajustada aos limites apresentados na Seção 3.3.

A estrutura irá utilizar uma abordagem de controle baseada principalmente na abordagem “assistir quando necessário” podendo implementar controle de impedância mecânica de acordo com a necessidade.

Definiu-se também as seguintes características para o jogo sério a ser integrado com a estrutura:

Área de Atuação: Motora;

Tecnologia de interação: Detecção de movimentos utilizando os próprios encoders nos atuadores do suporte ativo de peso;

Interface do jogo: Ambiente virtual 3D;

Número de jogadores: 1;

Gênero: Simulação combinado com plataforma;

Adaptabilidade: Automática, de acordo com o desempenho do jogador;

Assistência do jogo: integrada com o sistema de controle da estrutura.

De posse das informações apresentadas no Capítulos anteriores, o desenvolvimento da estrutura é detalhado nos Capítulos a seguir, iniciando pela modelagem matemática dos módulos.

CAPÍTULO III MODELAGEM MATEMÁTICA DAS ESTRUTURAS

Após o estudo detalhado das metodologias e estruturas existentes na literatura, o capítulo atual dedica-se a apresentar os procedimentos adotados e resultados obtidos para a modelagem matemática da estrutura HOPE-G.

Conforme descrito na Introdução a estrutura está dividida em 3 módulos onde um destes já se encontra desenvolvido (SALIM, 2018). Logo, neste Capítulo serão apresentados os procedimentos adotados para a modelagem matemática do módulo de movimentação por esteiras bipartidas e do módulo de suporte ativo de peso, onde encontra-se a principal inovação deste trabalho.

3.1. Modelagem Matemática do Módulo de Esteiras Bipartidas

A primeira modelagem realizada corresponde ao sistema de esteiras bipartidas baseado no projeto original do MIT-Skywalker (SUSKO, 2015). O principal objetivo desta nova implementação é atender as recomendações finais proposta pelos desenvolvedores da última versão desta referência.

Desta forma, os objetivos podem ser definidos como a redução da altura das esteiras e simplificação dos atuadores que realizam o movimento basculante de cada esteira. A redução da altura das esteiras é a principal prioridade pois isso facilita o acesso dos pacientes à estrutura, reduz o risco de quedas e reduz o espaço ocupado pelo equipamento.

A solução proposta nesta tese para atender a estas recomendações consiste na substituição dos cames lineares, posicionados abaixo de cada módulo da esteira (Figura 20), por atuadores pneumáticos os quais serão posicionados atrás da esteira. Sendo assim, as esteiras realizam o movimento basculante através do avanço/recuo dos cilindros pneumáticos eliminando a necessidade do came linear e consequentemente tornando menor a altura de trabalho da estrutura.

O esquema estrutural adotado para representar cada uma das esteiras é apresentado na Fig. 32. Adotou-se um ângulo limite de 30° entre o atuador e plano transversal com o objetivo de garantir o funcionamento adequado em todo o espaço de trabalho sem a possibilidade da presença de singularidades (R; BRODELL, 1970). O pistão pneumático é suspenso a uma altura H_p , e é conectado a uma base fixa através de uma articulação de rotação. O pistão então se conecta à esteira de comprimento L e espessura e também por uma rótula. A esteira se conecta à base fixa também através de um GDL de rotação, mantido a uma altura máxima $d_{m\acute{a}x}$. A altura da extremidade da esteira em relação ao chão é representada por d e a diferença angular entre a posição inicial e posição de descida máxima é representada por α .

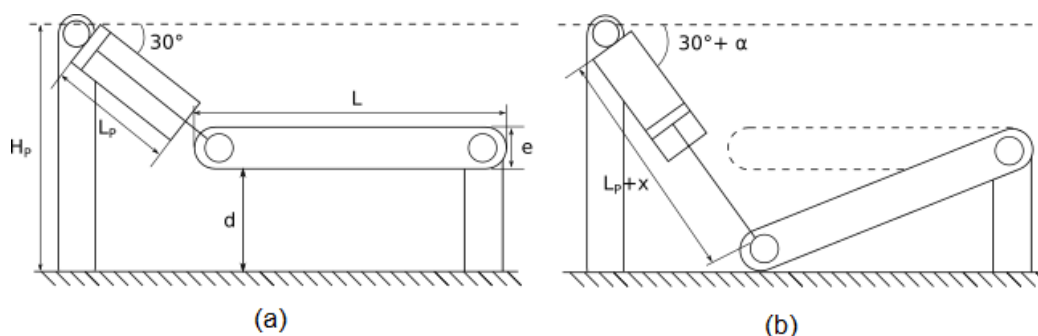


Figura 32 – Esquema estrutural adotado para o Módulo de Movimentação por Esteira Bipartida: (a) representação da posição inicial; e (b) representação da posição de descida máxima.

O comprimento L_p representa o tamanho mínimo do pistão pneumático, o qual será estendido em função de um incremento x . Assim, é possível obter uma equação matemática relacionando o deslocamento x do pistão, coordenada articular desta estrutura, com a distância entre a esteira e o chão d , coordenada operacional.

Para obter o modelo geométrico inverso desta estrutura é necessário realizar uma análise das relações trigonométricas das posições limite da estrutura, Fig. 32. Para facilitar a representação desta análise foi considerado os triângulos formados entre a diferença entre a altura variável d e a fixa H_p , o comprimento do pistão $H_p + x$, e a linha horizontal que os conecta, conforme apresentado na Fig. 33.

Observando este esquema pode-se deduzir a relação $d = f(x, \alpha)$ de acordo com a representação apresentada na Eq. (2).

$$d(x, \alpha) = H_p - (L_p + x)\text{sen}(30^\circ + \alpha) \quad (2)$$

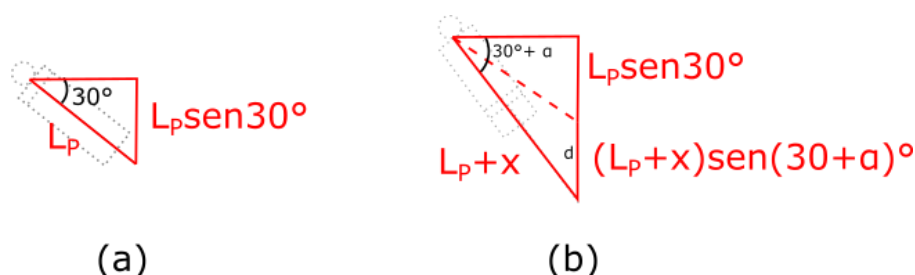


Figura 33 – Representação trigonométrica da (a) posição inicial; e (b) da posição de descida máxima.

Ainda analisando o esquema trigonométrico é possível observar que na posição inicial, Fig. 33(a), e na posição de descida máxima, Fig. 33(b), os catetos horizontais possuem a mesma dimensão. Sendo assim, é possível escrever que:

$$L_p \cos 30^\circ = (L_p + x) \cos(30^\circ + \alpha) \quad (3)$$

Isolando o ângulo α é possível obter uma expressão deste em função apenas da extensão x do pistão, conforme Eq.(4).

$$\alpha = \cos^{-1} \left[\frac{L_P}{L_P + x} \frac{\sqrt{3}}{2} \right] - 30^\circ \quad (4)$$

Substituindo a expressão encontrada para α na Eq. (4), é possível obter uma relação explícita $d = f(x)$, Eq. (5). A equação pode ainda ser extrapolada para uma forma temporal, fazendo $x = \frac{d_{max}}{t} = \beta t$, o que é interessante para viabilizar a modelagem dinâmica da estrutura, descrita na seção 3.3.

$$\begin{aligned} d(x) &= H_P - (L_P + x) \operatorname{sen} \left(\cos^{-1} \left(\frac{L_P}{L_P + x} \cos 30^\circ \right) \right) - e \Rightarrow \\ \Rightarrow d(t) &= H_P - (L_P + \beta t) \operatorname{sen} \left(\cos^{-1} \left(\frac{L_P}{L_P + \beta t} \cos 30^\circ \right) \right) - e \Rightarrow \\ \Rightarrow d(t) &= H_P - (\beta t + L_P) \sqrt{1 - \frac{(L_P^2 \cos(\alpha))^2}{(\beta t + L_P)^2}} \end{aligned} \quad (5)$$

3.2. Modelagem Matemática do Módulo de Suporte Ativo de Peso Corporal

O projeto da estrutura para o suporte ativo do peso do paciente foi realizado visando minimizar os acoplamentos das coordenadas articulares da estrutura através da aplicação de uma arquitetura baseada na família “multipteron” (GOSSELIN; LALIBERTÉ; VEILLETTE, 2015; GOSSELIN et al., 2007) mantendo apenas o movimento sagital da pélvis mecanicamente restringido devido a sua baixa influência no processo de reabilitação da marcha humana (ROBERTS, 2004).

Para isso, propõe-se a aplicação de uma estrutura definida como 4-PRRU+PRRS, pela nomenclatura tradicional definida como uma forma de “pentapteron”. Um desenho esquemático para esta estrutura é exibido na Fig. 34. Neste desenho, as articulações de translação e de rotação foram agrupadas em articulações cilíndricas q_i para facilitar a representação no desenho. Porém foram utilizadas articulações de translação e rotação na durante a construção do protótipo descrita nos Capítulos seguintes.

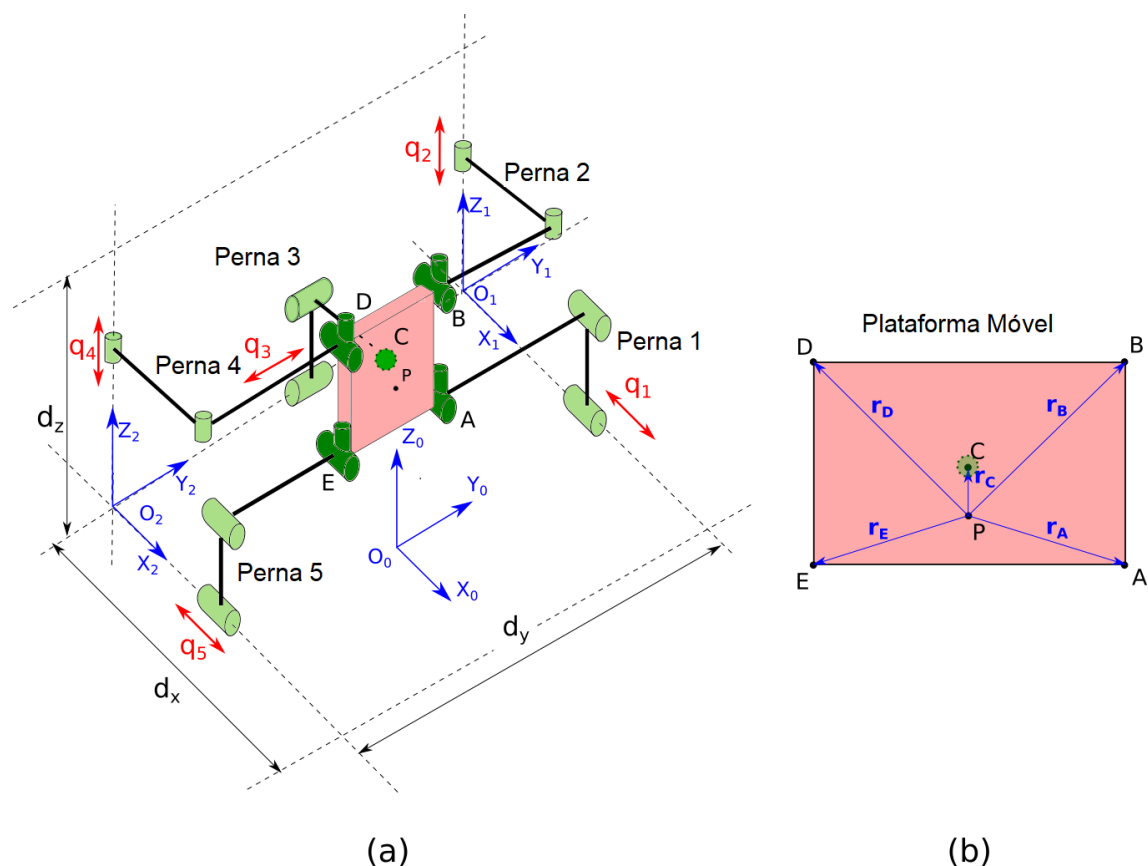


Figura 34 – (a) Desenho esquemático para a estrutura do suporte ativo da pélvis, baseada no “pentapteron” (4-PRRU+PRRS); e (b) posição do ponto de interesse P em relação aos vértices da plataforma móvel.

Nesta seção apresenta-se os procedimentos adotados para a modelagem cinemática desta estrutura, baseada no modelo geométrico inverso, o cálculo de suas posições singulares e sua modelagem dinâmica.

Para a modelagem deste sistema utilizou-se coordenadas equipolentes para os referenciais ilustrados na Fig. 34(a), de forma que cada coordenada articular q_i esteja em uma das direções dos vetores unitários do referencial.

Sendo assim, destaca-se o fato de que, para cada perna i , o vetor \mathbf{u}_i que conecta a articulação ativa desta perna a um dos pontos na plataforma móvel, pertence a um plano perpendicular ao vetor unitário da direção do atuador.

Com base nestes fatos, procede-se com a modelagem desta estrutura analisando individualmente cada perna i e sistematizando as equações obtidas em uma matriz relacionando as coordenadas articulares e operacionais.

Perna 1

Para analisar a posição do ponto A da plataforma móvel a partir de uma perna i , pode-se igualar a posição direta em relação ao referencial inercial, com a posição em relação aos

referenciais da perna em questão. Para o caso da perna 1, esta relação é ilustrada na Fig. 35 e formalizada na Eq. (6).

$${}^0\mathbf{r}_p + \mathbf{r}_A = {}^0\mathbf{r}_1 + q_1\mathbf{i}_1 + \mathbf{u}_1 \quad (6)$$

Sabe-se que o vetor \mathbf{u}_1 pertence a um plano paralelo ao plano Y_1Z_1 , portanto perpendicular ao vetor unitário \mathbf{i}_1 , e que $\mathbf{i}_1 = \mathbf{i}_0$ devido ao fato de se adotar referenciais com coordenadas equipolentes. Logo, ao se realizar um produto escalar em ambos os lados da Eq. (6) por \mathbf{i}_0 , tem-se que:

$$\begin{aligned} q_1 &= ({}^0\mathbf{r}_p + \mathbf{r}_A - {}^0\mathbf{r}_1) \cdot \mathbf{i}_0 \Rightarrow \\ \Rightarrow q_1 &= \left(\begin{matrix} {}^0r_{p_x} \\ {}^0r_{p_y} \\ {}^0r_{p_z} \end{matrix} + \begin{matrix} r_{Ax} \\ r_{Ay} \\ r_{Az} \end{matrix} - \begin{matrix} -\frac{dx}{2} \\ \frac{dy}{2} \\ 0 \end{matrix} \right) \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \Rightarrow \\ &\Rightarrow q_1 = {}^0r_{p_x} + \frac{dx}{2} + r_A \cos\varphi \end{aligned} \quad (7)$$

Onde φ corresponde ao ângulo formado entre os vetores \mathbf{r}_A e \mathbf{i}_0 , conforme ilustrado na Fig. 36(b), o qual também é definido como orientação da plataforma móvel quanto a sua rotação em torno do eixo Z_0 .

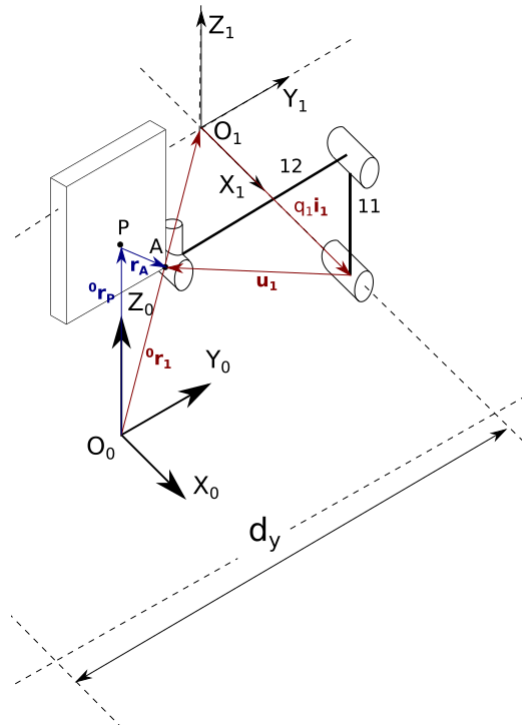


Figura 35 - Análise da posição do ponto A em relação ao referencial inercial e a perna 1.

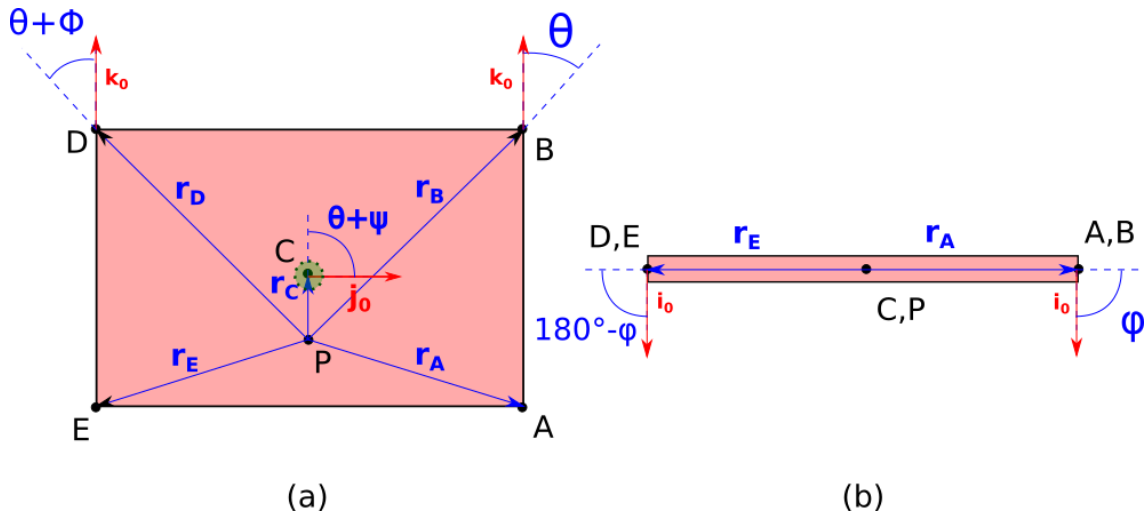


Figura 36 - Ângulos entre os vetores unitários e os vetores de posição das extremidades em relação ao ponto P: (a) ângulos da rotação em torno de X_0 ; e (b) da rotação em torno de Z_0 .

Perna 2

Para a perna 2, analisa-se a posição do ponto B utilizando a mesma metodologia apresentada na subseção anterior. A relação vetorial para este caso é ilustrada na Fig. 37 e equivale a Eq. (8).

$${}^0\mathbf{r}_P + \mathbf{r}_B = {}^0\mathbf{r}_1 + q_2\mathbf{k}_1 + \mathbf{u}_2 \quad (8)$$

Utilizando a propriedade da perpendicularidade do vetor \mathbf{u}_2 em relação ao vetor unitário $\mathbf{k}_1 = \mathbf{k}_0$, é possível chegar à relação de q_2 com as coordenadas operacionais do problema conforme demonstrado na Eq. (9).

$$\begin{aligned} q_2 &= ({}^0\mathbf{r}_P + \mathbf{r}_B - {}^0\mathbf{r}_1) \cdot \mathbf{k}_0 \Rightarrow \\ \Rightarrow q_2 &= \left(\begin{matrix} {}^0r_{p_x} \\ {}^0r_{p_y} \\ {}^0r_{p_z} \end{matrix} + \begin{matrix} r_{B_x} \\ r_{B_y} \\ r_{B_z} \end{matrix} - \begin{matrix} -\frac{dx}{2} \\ \frac{dy}{2} \\ 0 \end{matrix} \right) \cdot \begin{matrix} 0 \\ 0 \\ 1 \end{matrix} \Rightarrow \\ &\Rightarrow q_2 = {}^0r_{p_z} + r_B \cos(\theta) \end{aligned} \quad (9)$$

O ângulo θ corresponde ao ângulo formado entre o vetor unitário \mathbf{k}_0 e o vetor \mathbf{r}_B , conforme ilustrado na Fig. 36(a).

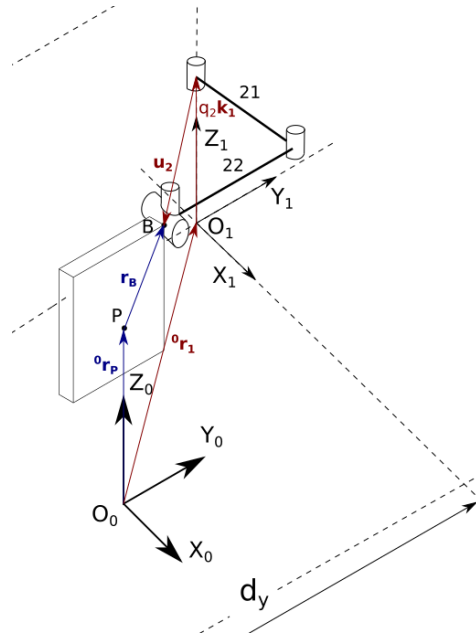


Figura 37 - Análise da posição do ponto B em relação ao referencial inercial e a perna 2.

Perna 3

Para a perna 3 o ponto C é utilizado para analisar a relação entre a posição pelo referencial inercial, e a mesma posição em relação à perna 3. A relação vetorial em questão é apresentada na Fig. 38 e na Eq. (10).

$${}^0\mathbf{r}_p + \mathbf{r}_c = {}^0\mathbf{r}_2 + q_3\mathbf{j}_2 + \mathbf{u}_3 \quad (10)$$

Utilizando novamente a propriedade de perpendicularidade do vetor \mathbf{u}_3 em relação a \mathbf{j}_2 , e realizando o produto escalar em ambos os lados da Eq. (10) por $\mathbf{j}_2 = \mathbf{j}_0$, tem-se:

$$\begin{aligned} q_3 &= ({}^0\mathbf{r}_p + \mathbf{r}_c - {}^0\mathbf{r}_2) \cdot \mathbf{j}_0 \Rightarrow \\ \Rightarrow q_3 &= \left(\begin{matrix} {}^0r_{px} \\ {}^0r_{py} \\ {}^0r_{pz} \end{matrix} \right) + \begin{matrix} r_{cx} \\ r_{cy} \\ r_{cz} \end{matrix} - \begin{matrix} -\frac{dx}{2} \\ dy \\ -\frac{dy}{2} \\ 0 \end{matrix} \cdot \begin{matrix} 0 \\ 1 \\ 0 \end{matrix} \Rightarrow \\ \Rightarrow q_3 &= {}^0r_{py} + r_c \cos(\theta + \psi) + \frac{dy}{2} \end{aligned} \quad (11)$$

O ângulo $\theta + \psi$ oriundo do produto escalar entre o vetor \mathbf{r}_c e \mathbf{j}_0 representa a orientação em torno do eixo X_0 da plataforma móvel, conforme Fig. 36(a).

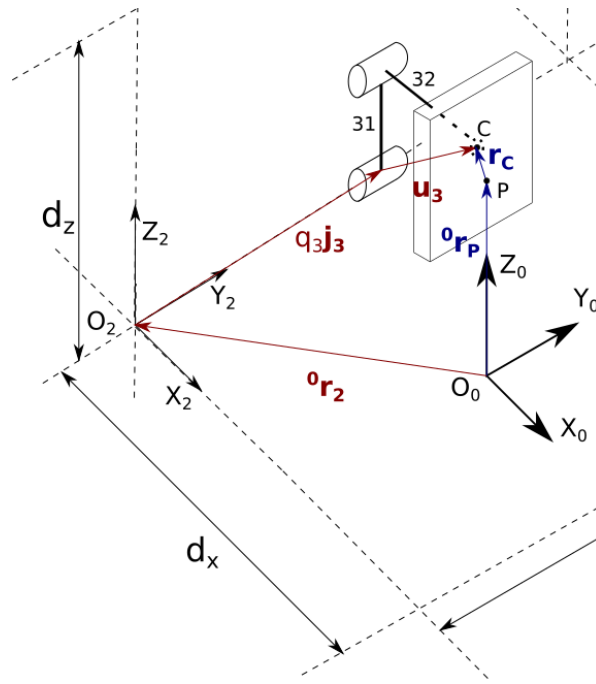


Figura 38 - Análise da posição do ponto C em relação ao referencial inercial e a perna 3.

Perna 4

A seguir, utiliza-se o ponto D para realizar a análise da posição deste em relação ao referencial inercial, e em relação à perna 4. O esquema vetorial desta relação é destacado na Fig. 39 e apresentado algebricamente na Eq. (12).

$${}^0\mathbf{r}_p + \mathbf{r}_D = {}^0\mathbf{r}_2 + q_4\mathbf{k}_2 + \mathbf{u}_4 \quad (12)$$

Utilizando a perpendicularidade do vetor \mathbf{u}_4 em relação a \mathbf{k}_2 e multiplicando a Eq. (12) pelo produto escalar de $\mathbf{k}_2 = \mathbf{k}_0$ obtém-se as seguintes relações:

$$\begin{aligned} q_4 &= ({}^0\mathbf{r}_p + \mathbf{r}_D - {}^0\mathbf{r}_2) \cdot \mathbf{k}_0 \Rightarrow \\ \Rightarrow q_4 &= \left(\begin{matrix} {}^0r_{px} \\ {}^0r_{py} \\ {}^0r_{pz} \end{matrix} \right) + \begin{matrix} r_{Dx} \\ r_{Dy} \\ r_{Dz} \end{matrix} + \begin{matrix} -\frac{dx}{2} \\ dy \\ -\frac{dy}{2} \\ 0 \end{matrix} \cdot \begin{matrix} 0 \\ 0 \\ 1 \end{matrix} \Rightarrow \\ \Rightarrow q_4 &= {}^0r_{pz} + r_D \cos(\theta + \phi) \end{aligned} \quad (13)$$

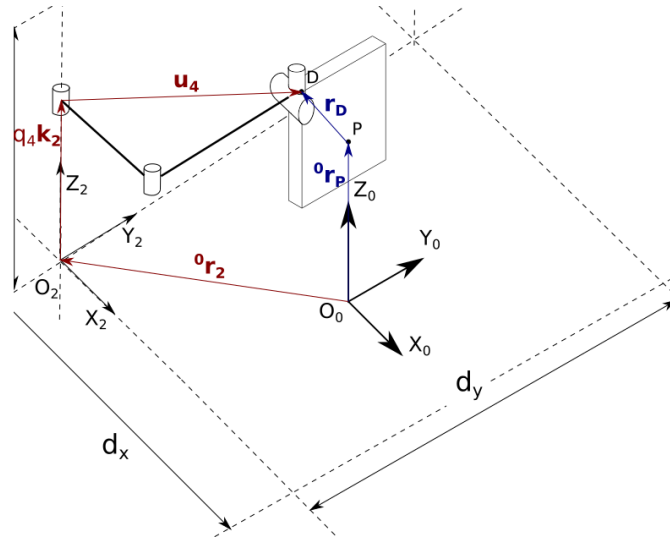


Figura 39 - Análise da posição do ponto D em relação ao referencial inercial e a perna 4.

Onde o ângulo $\theta + \phi$ é a diferença angular entre o vetor r_D e o vetor unitário k_0 , conforme definido na Fig. 36(a).

Perna 5

Por fim, analisa-se o ponto E para obter as relações geométricas da perna 5 em função das coordenadas operacionais e articulares. Os vetores desta análise são apresentados na Fig. 40, e as equações resultantes na Eq. (14).

$${}^0r_p + r_E = {}^0r_2 + q_5 i_2 + u_5 \quad (14)$$

Através da perpendicularidade do vetor u_5 em relação a i_2 , e multiplicando a Eq. (14) pelo produto escalar de $i_2 = i_0$, tem-se:

$$\begin{aligned} q_5 &= ({}^0r_p + r_E - {}^0r_2) \cdot i_0 \Rightarrow \\ \Rightarrow q_5 &= \left(\begin{matrix} {}^0r_{p_x} \\ {}^0r_{p_y} \\ {}^0r_{p_z} \end{matrix} + \begin{matrix} r_{E_x} \\ r_{E_y} \\ r_{E_z} \end{matrix} - \begin{matrix} -\frac{dx}{2} \\ \frac{dy}{2} \\ 0 \end{matrix} \right) \cdot \begin{matrix} 1 \\ 0 \\ 0 \end{matrix} \Rightarrow \\ \Rightarrow q_5 &= {}^0r_{p_x} + r_E \cos(180^\circ - \varphi) + \frac{dx}{2} = {}^0r_{p_x} - r_E \cos(\varphi) + \frac{dx}{2} \end{aligned} \quad (15)$$

Dessa forma, é possível notar que $[J_q] = [I]$, o que significa que as posições singulares existirão apenas para o caso em que $\det([J_x]) = 0$. Logo, estes casos podem ser expressos de acordo com a Eq. (22).

$$\begin{aligned} \det([J_x]) &= r_A r_D \sin(\phi + \theta) \sin(\varphi) - r_B r_E \sin(\theta) \sin(\varphi) - \\ &\quad - r_A r_B \sin(\theta) \sin(\varphi) + r_D r_E \sin(\phi + \theta) \sin(\varphi) = 0 \Rightarrow \\ \Rightarrow (r_A + r_E) r_D \sin(\phi + \theta) \sin(\varphi) - (r_A + r_E) r_B \sin(\theta) \sin(\varphi) &= 0 \Rightarrow \\ \Rightarrow r_D \sin(\phi + \theta) &= r_B \sin(\theta) \end{aligned} \quad (22)$$

Portanto, sempre que a Equação (22) for verdadeira, o sistema estará em uma posição de singularidade e estará sujeito à perda de controle e/ou mobilidade nesta posição.

Com base nesta informação, para se projetar esta estrutura é necessário garantir que as dimensões selecionadas de r_B , r_D e ϕ não possibilitem a ocorrência da Eq. (22) em nenhum ponto do espaço de trabalho em θ .

É possível notar também que os graus de liberdade de translação desta estrutura estão limitados apenas pelos respectivos limites do espaço de trabalho, uma vez que estes não apresentam termos na matriz jacobiana $[J_x]$ desta estrutura.

Atualização dos atuadores

Visando reduzir o custo com atuadores lineares de alta velocidade, após a modelagem matemática desta estrutura, um novo modelo de atuador foi proposto, baseado em motores DC de 24V acoplados a encoders rotativos, e transformando o movimento angular destes motores em deslocamento linear através de um mecanismo de 3 barras do tipo biela-manivela. O esquema projetado para este sistema é representado na Figura 41, e a relação entre q_i e θ_i é descrita na Eq. (23).

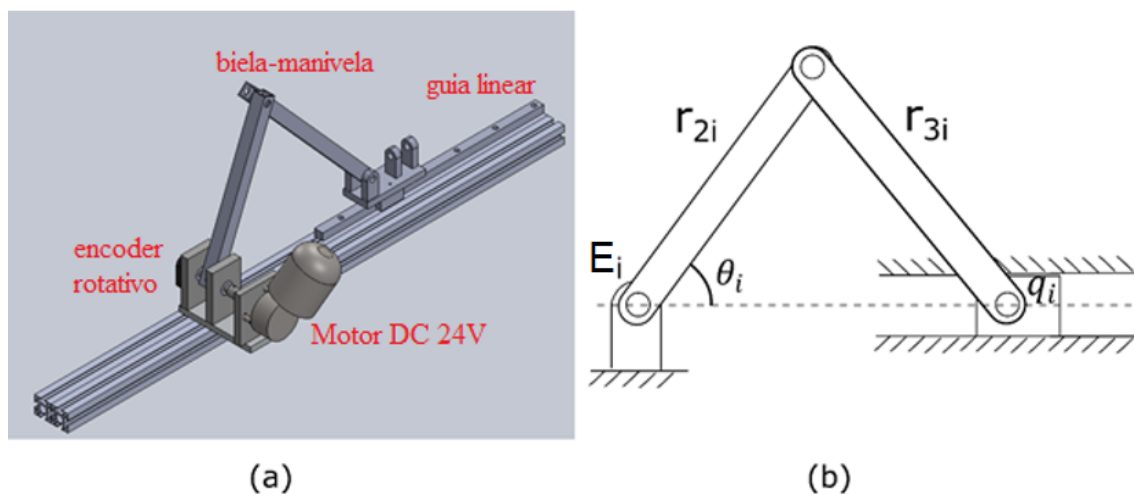


Figura 41 – (a) Modelo CAD construído para o atuador baseado no mecanismo biela-manivela; (b) desenho esquemático evidenciando os parâmetros adotados.

$$q_i = r_{2i} \cos \theta_i - r_{3i} \cos \theta_i \quad (23)$$

3.3. Modelo Dinâmico e aplicação da Simulação & Otimização

Para definir corretamente as dimensões dos elementos mecânicos e atuadores de cada um dos módulos da estrutura, um modelo computacional considerando características geométricas e dinâmicas foi elaborado. Este modelo combina as características geométricas descritas nas subseções anteriores com as especificações e premissas adotadas para o funcionamento da estrutura HOPE-G.

Visando simplificar o processo de modelagem dinâmica dos módulos a carga do peso corporal do paciente foi considerada como um carregamento pontual estático no caso do Módulo de Movimentação por Esteiras Bipartidas. Para o módulo de Suporte Ativo do Peso Corporal apenas a massa do paciente foi considerada, considerando o carregamento dinâmico pontual gerado por esta sobre as guias lineares q_i , e transmitidas para os motores DC M_i através do mecanismo biela-manivela.

A abordagem é válida devido às baixas velocidades e acelerações presentes na cinemática da reabilitação da marcha humana (RODRIGUES, 2017). Sendo assim, a massa e a inércia das barras e demais elementos mecânicos presentes neste módulo podem ser desconsiderados sem prejuízo às especificações do sistema.

Para ambos os módulos, considerou-se uma massa crítica $M_p = 150 \text{ Kg}$ definida a partir de dados estatísticos levantados com base na população brasileira, de forma que este valor permite que a estrutura opere corretamente em cerca de 95% da população (IBGE - INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA, 2010).

Considerou-se também um período de marcha rítmica regular para especificação de um tempo de ciclo limite $t_s = 0,4 \text{ s}$, correspondendo ao tempo médio gasto por um indivíduo saudável para realizar o ciclo de movimentos da marcha com uma velocidade moderada (KHARB et al., 2011).

Módulo de Movimentação por Esteiras Bipartidas

Para a modelagem do módulo de movimentação por esteiras bipartidas as seguintes características foram consideradas:

- Tempo limite para movimentação: $0,2 \text{ s}$ (Susko, 2015);
- Comprimento das esteiras (L): $0,95 \text{ m}$;
- Massa máxima considerada do paciente (M_p) = 150 Kg ;
- Altura máxima das esteiras (d_{max}): $3'' = 0,0762 \text{ m}$;
- Tamanho do cilindro (LP): $4'' = 0,1016 \text{ m}$;

- Massa da esteira (M_e): 26 Kg;

O modelo dinâmico para este módulo foi então deduzido com o auxílio do software de matemática simbólica Maplesoft Maple®, obtendo as expressões de velocidade e aceleração a partir de diferenciações temporais da Eq. (5).

$$\dot{d}(t) = -\beta \sqrt{1 - \frac{L_p^2 \cos^2 \alpha}{(\beta t + L_p)^2}} - \frac{L_p^2 \cos^2 \alpha \beta}{(\beta t + L_p)^2 \sqrt{1 - \frac{L_p^2 \cos^2 \alpha}{(\beta t + L_p)^2}}} \quad (24)$$

$$\ddot{d}(t) = \frac{L_p^2 \cos^2 \alpha \beta^2}{(\beta t + L_p)^3 \sqrt{1 - \frac{L_p^2 \cos^2 \alpha}{(\beta t + L_p)^2}}} + \frac{L_p^4 \cos^4 \alpha \beta^2}{(\beta t + L_p)^5 \left(1 - \frac{L_p^2 \cos^2 \alpha}{(\beta t + L_p)^2}\right)^2} \quad (25)$$

O modelo acima foi então implementado no MATLAB®, onde foi considerado o tempo limite de movimentação igual a metade do tempo de ciclo limite para manter a ação válida dentro do passo de cada membro inferior (SUSKO, 2015). Os resultados obtidos estão apresentados nos gráficos da Fig. 42.

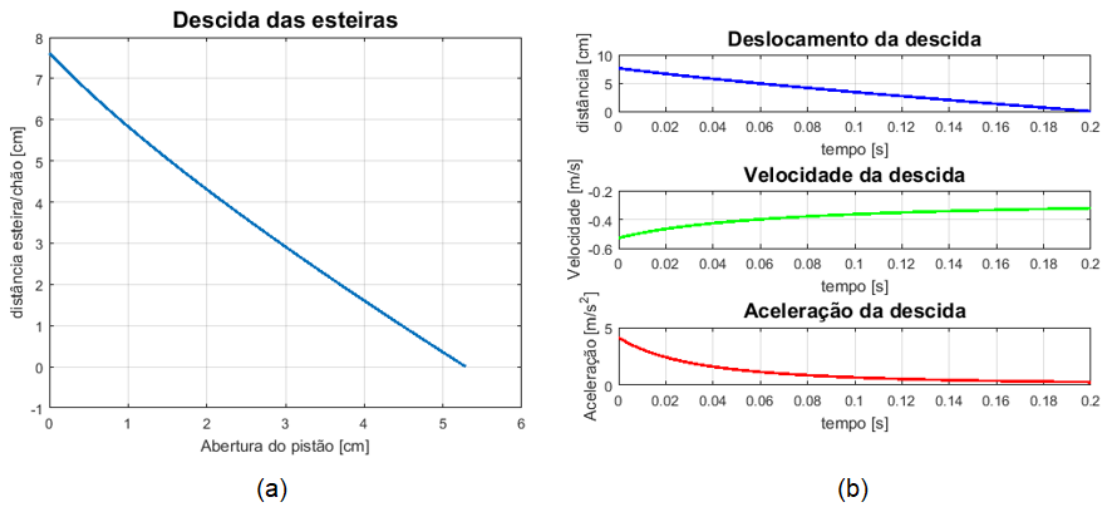


Figura 42 – (a) gráfico da distância da extremidade de uma das esteiras em relação ao chão em função da extensão do comprimento do pistão; e (b) valores calculados para deslocamento, velocidade e aceleração em cada instante de tempo durante a descida da esteira.

Além dos dados apontados na Figura 42, foi determinada a carga dinâmica máxima a ser suportada por cada atuador pneumático. Os cálculos foram efetuados considerando que cada cilindro suporta estaticamente o peso da esteira e do paciente, Eq. (26), e que a carregamento dinâmico ocorre apenas com a movimentação das esteiras, Eq. (27).

$$R_p = \frac{M_p + M_e}{\sin \alpha} \cdot 9.81 \cdot \frac{f_r}{2}; f_r = 1 \quad (26)$$

$$F_c = M_e \cdot \max \ddot{d}(t) \quad (27)$$

Na Equação (26), f_r corresponde a um fator de supressão do peso do paciente de acordo com a ação do Módulo de Suporte Ativo de Peso Corporal. Considerou-se $f_r = 1$ para o dimensionamento dos cilindros e dividiu-se por 2 para considerar o peso equilibrado para cada cilindro.

Os cálculos foram executados de forma computacional de acordo com o código apresentado nos Apêndices. A carga dinâmica máxima F_c observada nos atuadores é de 108 N, somada à carga estática R_p do paciente de 1727 N totalizando um carregamento total sobre os atuadores de 1835 N.

Estes valores serão levados em consideração no processo de aquisição dos equipamentos necessários para a construção do módulo de forma que o pistão seja capaz de suportar a carga e possua o curso mínimo necessário.

Módulo de Suporte Ativo de Peso Corporal

Para a especificação deste módulo, os requisitos de movimentação foram estabelecidos de acordo com a fisiologia articular da marcha humana e da articulação sacroilíaca. Estes estão detalhados na Tabela 1.

Tabela 1 - Espaço de trabalho especificado para cada grau de liberdade do módulo de suporte ativo de peso corporal (Roberts, 2004).

Grau de liberdade	Direção	Amplitude da Pélvis	Amplitude da estrutura
Translação	Frontal	-	$\pm 200 \text{ mm}$
Translação	Lateral	-	$\pm 100 \text{ mm}$
Translação	Vertical	-	$\pm 150 \text{ mm}$
Rotação	Transversal	$\pm 25^\circ$	$\pm 25^\circ$
Rotação	Coronal	$\pm 5^\circ$	$\pm 5^\circ$

Além disso, quatro diretivas foram adotadas:

- Barras que ligam a plataforma móvel do módulo às guias lineares devem suportar a carga estática máxima, especificada como a massa crítica do sistema;
- As guias lineares devem percorrer o espaço de trabalho especificado dentro do tempo de ciclo crítico;

- Barras que ligam as guias lineares aos motores DC devem suportar a carga dinâmica máxima de forma pontual desconsiderando o efeito das inércias para simplificar o modelo;
- Os atuadores 2 e 4 serão implementados juntamente com um sistema de molas passivo para aliviar a carga estática do paciente, Fig. 43, permitindo que os atuadores se concentrem apenas nos esforços dinâmicos.
- Uma mola adicional foi conectada à plataforma móvel, Fig. 43, sem prejuízo ao espaço de trabalho definido, para melhorar o suporte passivo da estrutura e aliviar a carga estática dos demais motores.

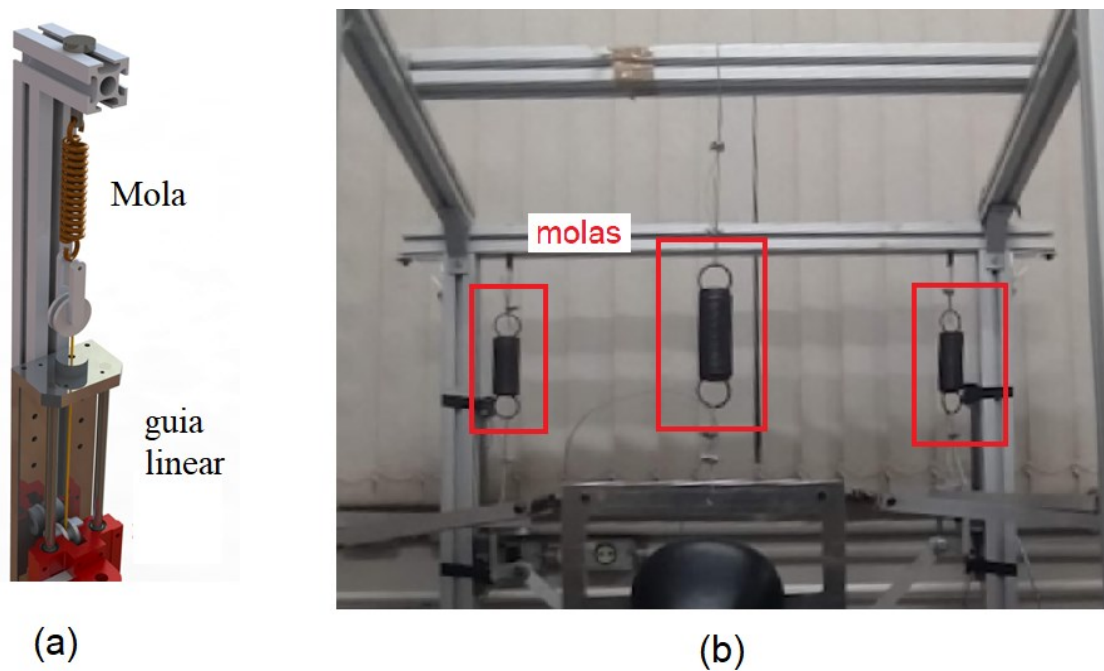


Figura 43 - sistema de supressão passivo da carga estática do peso do paciente presente nos atuadores 2 e 4 e na plataforma móvel: (a) desenho esquemático; (b) Implementação no protótipo real.

Logo, a partir das diretrizes propostas define-se que as guias lineares devem ser capazes de iniciar suas respectivas trajetórias a partir do início do curso com velocidade inicial nula e finalizar o movimento na posição final novamente com uma velocidade nula. Dessa forma, a equação dinâmica que define o maior esforço F_i aplicado sob cada atuador q_i pode ser definida como:

$$F_i = f_{r_i} M_P \frac{d_i}{t_c^2} \quad (28)$$

$$\begin{cases} i = 1, 2, 4, 5 \Rightarrow f_{r_i} = 0.5 \\ i = 3 \Rightarrow f_{r_i} = 1 \end{cases}$$

Onde d_i corresponde ao espaço de trabalho especificado para cada guia linear, e o parâmetro f_{r_i} corresponde a um fator de correção aplicado na equação para considerar a distribuição de peso entre os atuadores 1 e 5, e os atuadores 2 e 4. O atuador 3 atua sozinho na direção \mathbf{j}_0 , o que significa que este não necessita da correção, e, portanto, $f_{r_3} = 1$.

O modelo também considera os requisitos mínimos para potência elétrica e velocidade angular dos motores DC especificados de forma a possibilitar a correta movimentação do mecanismo biela-manivela. Para modelar esta etapa, um diagrama de corpo livre foi construído evidenciando os esforços considerados, conforme representado na Fig. 44.

Neste diagrama, T_i representa o momento angular de entrada do motor M_i , e F_i a força aplicada sobre q_i . Note que a inércia e a massa das barras não são consideradas no modelo devido a hipótese simplificadora adotada para este modelo.

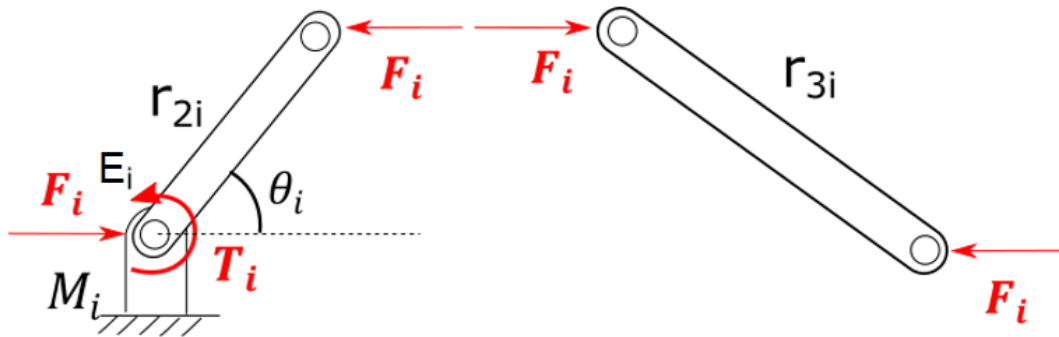


Figura 44 – Diagrama de corpo livre do mecanismo biela-manivela dos atuadores.

Aplicando as condições de equilíbrio de momentos no diagrama de corpo livre da Fig. 44., em relação ao ponto E_i , o momento T_i pode ser escrito como:

$$T_i = F_i r_{2i} \text{sen} \theta_i \quad (29)$$

diferenciando a equação (23) no tempo e isolando $\dot{\theta}_i$ obtém-se:

$$\begin{aligned} \dot{q}_i &= -r_{2i} \text{sen} \theta_i \dot{\theta}_i + r_{3i} \text{sen} \theta_i \dot{\theta}_i \Rightarrow \\ \Rightarrow \dot{\theta}_i &= \frac{\dot{q}_i}{(r_{3i} - r_{2i}) \text{sen} \theta_i} \end{aligned} \quad (30)$$

Logo, para se obter a expressão para a potência mínima em W (WEG, 2014) requerida para cada atuador aplica-se a equação de especificação de motores elétricos, conforme Eq. (31).

$$P_i [W] = \frac{(\max(\dot{\theta}_i) \max(T_i))}{9.55} \quad (31)$$

Finalmente, o último requisito a ser observado é a tensão aplicada nas seções transversais de cada barra. Foi utilizado como critério a comparação σ_{ij} da tensão calculada em cada seção

transversal através da teoria de Resistência dos materiais (UICKER et al., 2003), de acordo com a Eq.(32), onde F_j são os esforços aplicados no elemento, r_{ij} é a barra analisada e I_{ij} é o momento de inércia da barra.

$$\sigma_{ij} = \frac{F_j r_{ij}}{4I_{ij}} \quad (32)$$

Para obter as dimensões ótimas das barras que irão compor o mecanismo a tensão calculada conforme a Eq. (32) é comparada com o limite de escoamento do material selecionado definido como alumínio ($\sigma_{max} = 180 \text{ MPa}$). Para facilitar o processo de especificação das barras e reduzir custos de fabricação, considerou-se todas as barras do sistema como tubos retangulares com seções transversais compatíveis com os materiais disponíveis no Laboratório de Automação e Robótica da Universidade Federal de Uberlândia.

Por fim, todos os requisitos acima foram combinados em um modelo computacional elaborado no MATLAB® juntamente com um algoritmo de evolução diferencial formando um sistema interativo de Simulação & Otimização apresentado nos Apêndices. Este sistema foi utilizado para simular todas as condições dinâmicas deste Capítulo, bem como a condição para singularidades, como restrições de um problema de minimização onde as variáveis de projeto correspondiam aos comprimentos a serem selecionados para cada barra.

O resultado obtido através deste processo é apresentado na Tabela 2. As barras r_{2i} e r_{3i} são as que compõem o mecanismo biela-manivela de cada atuador i , sendo todas similares entre si, Figura 41. As barras r_{4i} e r_{5i} representam as barras de conexão entre a guia linear e a plataforma móvel para cada perna i , Figura 34(a).

Tabela 2 – Dimensões ótimas calculadas para cada barra do módulo.

Barra	Comprimento (mm)	Seção transversal (mm)	Espessura (mm)
r_{2i}	250	25.4	5
r_{3i}	300	25.4	5
r_{4i}	300	25.4	5
r_{5i}	300	25.4	5

A simulação também determinou as características necessárias para os motores DC, conforme descrito nas Eqs. (30) e (31). Os dados obtidos são apresentados na Tabela 3.

Tabela 3 – Especificações para os motores DC do módulo.

Motor	Potência necessária (W)	Velocidade Angular Necessária (RPM)
M_1	35	20
M_2	20	15
M_3	20	10
M_4	20	15
M_5	35	20

3.4. Simulações Computacionais em CAD/CAE

A partir da otimização realizada computacionalmente na etapa anterior as dimensões ótimas para os elementos mecânicos que compõe cada módulo foram obtidas. A partir deste ponto os modelos em CAD da estrutura foram construídos, e simulações cinemáticas de movimentos foram realizadas, com o objetivo de observar se as estruturas de fato realizavam os movimentos especificados durante a etapa de modelagem matemática e validando este.

O primeiro modelo construído foi para o módulo de movimentação por esteira bipartida. Esta simulação apresenta a movimentação simples das esteiras em função da extensão do comprimento dos atuadores pneumáticos. Uma sequência de imagens ilustrando a simulação obtida é apresentada na Fig. 45.

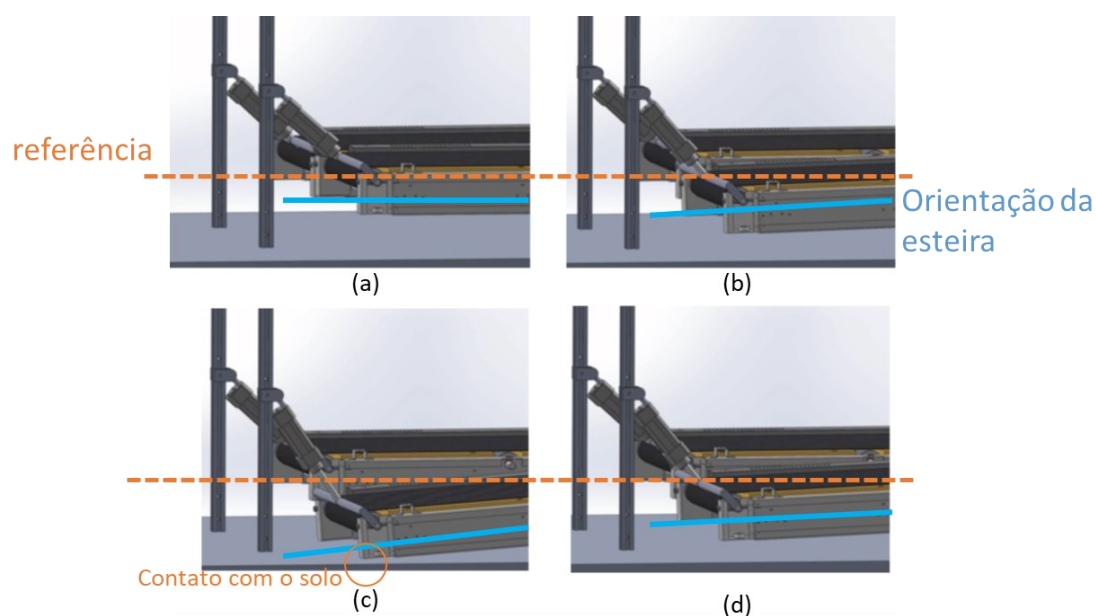


Figura 45 – Sequência de imagens representativas ilustrando a simulação efetuada para o módulo de movimentação por esteira bipartida: (a) Posição inicial, alinhada com a referência; (b) posição intermediária de descida; (c) posição de descida máxima, onde há contato com o solo; e (d) posição intermediária de subida, retornando ao alinhamento com a referência.

O próximo modelo construído foi para o módulo de Suporte Ativo de Peso Corporal, onde quatro simulações foram elaboradas.

A primeira simulação apresenta os movimentos básicos de cada um dos graus de liberdade da estrutura proposta. Esta simulação é importante para verificar e validar se os parâmetros especificados na modelagem e otimizados computacionalmente mantiveram a mobilidade desejada na estrutura. Uma sequência desta simulação é apresentada na Fig. 46.

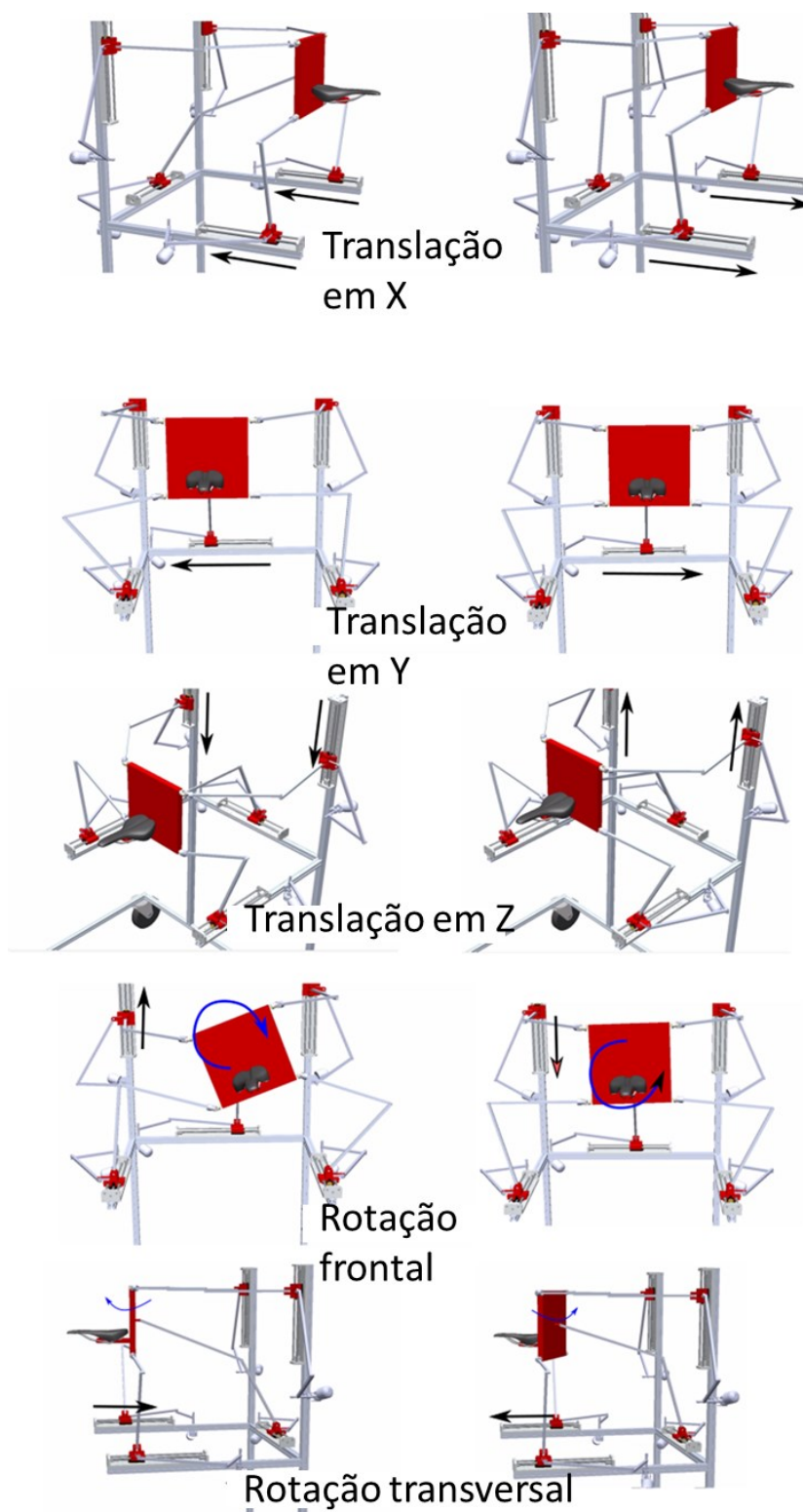


Figura 46 – Sequência de imagens representando a simulação dos movimentos elementares.

A segunda simulação elaborada tem como objetivo a demonstração do procedimento de montagem do módulo em conjunto com o paciente que irá realizar a sessão de treinamento de reabilitação. Esta simulação auxilia a visualização como os profissionais de saúde poderão

proceder com o equipamento para as sessões de treinamento diárias. Esta simulação também é representada por sequência de imagens, conforme visto na Fig. 47.

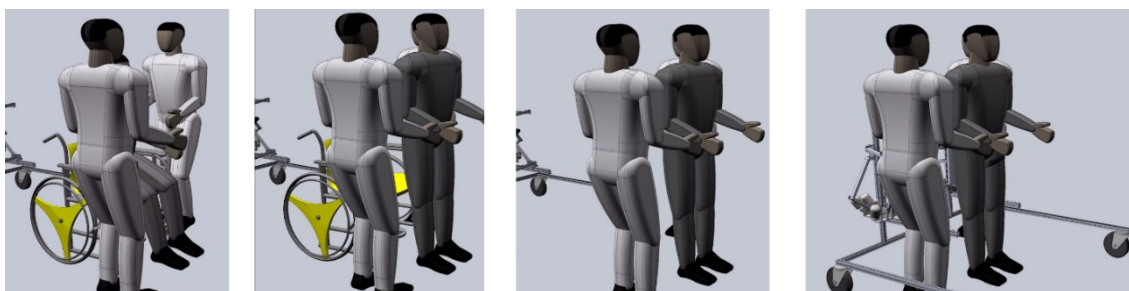


Figura 47 – Sequência de imagens representando a simulação do procedimento de colocação do paciente na estrutura.

As próximas simulações realizadas apresentam o comportamento da estrutura durante operação. A terceira traz apenas o Módulo de Suporte Ativo de Peso Corporal operando em um modo *standalone*, onde a sessão de treinamento de reabilitação ocorre diretamente sob o chão em um ambiente controlado, processo denominado *overground rehabilitation*. Neste modo o módulo atua com ação passiva assistida sob os movimentos da pélvis e no alívio do peso do corpo sob os membros inferiores.

A última simulação realizada mostra o funcionamento da estrutura como um conjunto completo, incluindo ambos os módulos operando em um exercício de reabilitação da marcha humana. Ambas as simulações citadas são representadas na Fig. 48.

Adicionalmente, todas as simulações representadas como imagens podem ser vistas em vídeo através do link <https://tinyurl.com/HOPE-G-Videos>.

Simulação em modo “Overground”



Simulação com esteiras bipartidas

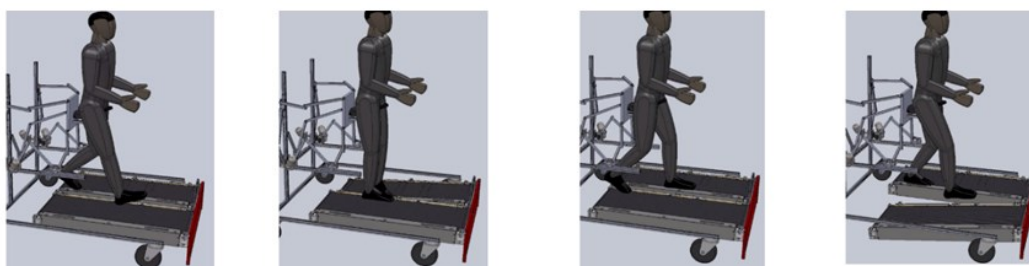


Figura 48 – Sequência de imagens representando as simulações operacionais da HOPE-G.

CAPÍTULO IV PROTÓTIPO PARTE 1: ATUADORES E CONTROLE

De posse de todas as dimensões devidamente modeladas no software CAD/CAE, iniciou-se a próxima etapa do projeto onde foi construído um protótipo do módulo de suporte ativo. Este protótipo foi utilizado para desenvolver o sistema de controle dos atuadores, o sistema de comunicação com o jogo eletrônico desenvolvido para operar em conjunto com a estrutura e realizar os testes piloto simulados.

A construção deste protótipo foi realizada de forma sequencial, de acordo com o seguinte cronograma:

1. Levantamento dos materiais necessários/disponíveis;
2. Testes de bancada do design proposto para o atuador;
3. Montagem da estrutura metálica passiva e implementação dos atuadores na estrutura;
4. Montagem dos circuitos e módulos de controle da estrutura;
5. Configuração dos atuadores;
6. Comunicação com PC;
7. Testes de movimentação.

Neste capítulo detalha-se uma destas etapas de desenvolvimento, iniciando pelos materiais e métodos utilizados para construção e desenvolvimento do sistema de controle da estrutura.

4.1. Materiais utilizados

Para construção deste protótipo, diversos aspectos construtivos previstos no projeto original foram adaptados para possibilitar a construção do equipamento com a disponibilidade de espaço, materiais e ferramentas disponíveis no Laboratório de Automação e Robótica da Universidade Federal de Uberlândia.

Estrutura passiva

A estrutura que comporta os elementos ativos do protótipo foi construída a partir de perfis de alumínio extrudado, de acordo com a lista de materiais apresentada na Tabela 4.

Tabela 4 – Lista de materiais aplicados na estrutura passiva.

Item	Dimensões	Quantidade
Rodízios articulados com trava	40 mm	4
Perfil estrutural de alumínio – canal 8	30x30x1500 mm	3
Perfil estrutural de Alumínio – canal 8	30x30x1500 mm	3
Perfil quadrado de alumínio – 5 mm de espessura	25,4x25,4x300 mm	10

Chapa de alumínio para montagem da plataforma móvel – 10 mm de espessura		1
Selim para monociclo, com conexão adaptada		1
Molas helicoidais em aço – capacidade 70 kg		3
Cabo de aço 4 mm	500 mm	3

Módulo dos atuadores

Os módulos de atuação da estrutura também foram construídos com base na disponibilidade de materiais do Laboratório de Automação e Robótica, sendo que os componentes estruturais foram usinados na oficina mecânica da Faculdade de Engenharia Mecânica. As peças usinadas foram elaboradas de acordo com a planificação dos modelos CAD, disponíveis no Apêndice.

A lista de materiais aplicados nestes componentes é apresentada na Tabela 5.

Tabela 5 – Lista de materiais aplicados no Módulo dos atuadores.

Item	Dimensões	Quantidade
motor DC 24V Bosch F006WM0310	46W, 51 rpm, 3Nm, redução 63:1, IP 44	5
Encoder Rotativo HEDS 5540	500 pulsos	1
Encoder Rotativo NPN LPD3806-600BM-G5-24C	600 pulsos	4
Conjunto de peças usinadas para suportar o motor DC	De acordo com Apêndice	5
Perfil quadrado de alumínio – 5 mm de espessura	25,4x25,4x300 mm	5
Perfil quadrado de alumínio – 5 mm de espessura	25,4x25,4x250 mm	5
Acoplador flexível para transmissão de torque	5mm x 5mm	5
Acoplador flexível para conexão do encoder	5mm x 5mm	5
Guia linear com cursor	300 mm	5
Sensor de fim de curso com haste	-	10

Sistema de controle

O sistema de controle foi construído em duas etapas, primeiro utilizando um jogo de protoboards e jumpers para prototipação, migrando para uma PCI com as trilhas definitivas posteriormente.

Os materiais utilizados também foram selecionados em função de disponibilidade em laboratório e são listados na Tabela 6.

Tabela 6 – Lista de materiais aplicados no sistema de controle.

Item	Dimensões	Quantidade
Microcontrolador Arduino Mega 2560	-	1
Driver de tensão para motores DC Monster Moto Shield VNH2SP30	16V 30A	4
Módulos de prototipação – Protoboard	400 pontos	3
Conectores “jumper” – macho/macho	-	90
Fonte DC Variável 0 a 30V	-	1
Placa de Circuito Impresso em fenolite	30x50mm	1
<i>Pin headers</i> fêmea	2,54mm	90

Ressalta-se que o microcontrolador selecionado para o desenvolvimento do controle da estrutura carece de um módulo *Quadrature Encoder Interface (QEI)*. Dessa forma, o controle de posição dos motores foi realizado baseado em interrupções e temporizadores. Isso compromete o determinismo do sistema pois o microcontrolador possui apenas seis temporizadores e dois deles foram utilizados para controlar parte da lógica do processo.

Apesar das condições adversas, a escolha do microcontrolador justifica-se pela facilidade de se implementar o código, disponibilidade de módulos compatíveis e levando em consideração de o foco deste trabalho ser o design mecânico proposto e a integração com o jogo sério.

O processo de desenvolvimento de cada uma das partes do protótipo é detalhado a seguir.

4.2. Testes de bancada do atuador

Inicialmente, construiu-se uma bancada de testes com apenas um dos módulos, conforme apresentado na Fig. 49. Nesta bancada, utilizou-se todos os componentes necessários para a construção de um módulo de atuação do suporte ativo de peso. O motor DC e o encoder da bancada foram conectados a um dos drivers de corrente e ao microcontrolador.

Com esta bancada, foram realizados testes iniciais na arquitetura proposta para o atuador, tais como:

- Teste de acionamento do motor DC pelo driver de corrente;
- Teste de funcionamento do encoder;
- Teste de comunicação entre atuador e microcontrolador;
- Construção de malha de controle fechada de teste, utilizando um controlador Proporcional Integral Derivativo (PID);
- Identificação de divergências significantes com o projeto original.

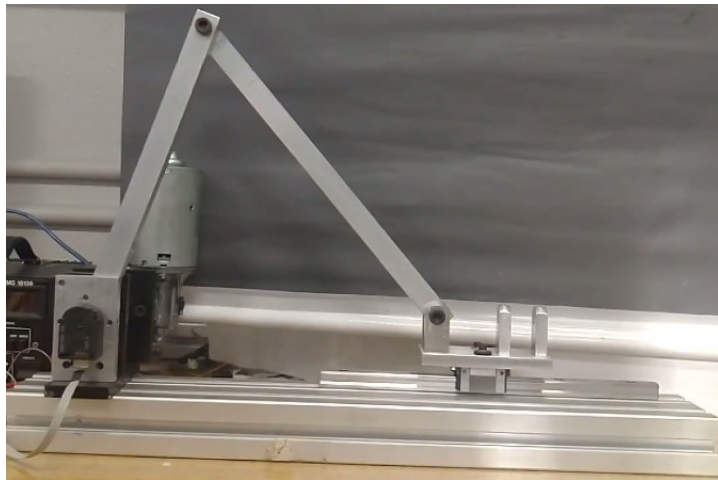


Figura 49 – Protótipo construído para o atuador do Módulo de Suporte Ativo de Peso Corporal.

A construção do protótipo foi limitada à disponibilidade de equipamentos e materiais do laboratório, foi necessário configurar os motores para operar em 16 V, abaixo do nominal, para não danificar os drivers utilizados devido a correntes de pico no acionamento.

Apesar da ponte H fornecida pelos drivers implementados suportar correntes de até 30 A, não foi possível implementar uma de controle para regular o abastecimento dos motores devido a indisponibilidade de temporizadores para controle de malha fechada no microcontrolador adotado. Dessa forma, o sistema foi apenas limitado para exceder a corrente nominal dos circuitos diretamente nos potenciômetros da fonte utilizada.

Apesar disso, os testes iniciais na bancada demonstraram que a movimentação do cursor estava de acordo com o especificado na seção 3.3, atendendo aos critérios de velocidade. A capacidade de carga do módulo foi avaliada conforme descrito na seção 6.3 e discutido nos resultados.

Durante os testes, utilizou-se a posição relativa do encoder para se estimar a posição do cursor e contagem de pulsos por milissegundo para se estimar a velocidade do atuador.

Através do uso da bancada, também foi possível verificar que o funcionamento dos encoders estava de acordo com o esperado, ocorrendo variações na resolução de cada atuador, que permaneceu entre 1 e 3 pulsos por milímetro e atingindo velocidades de até 1600 pulsos por segundo, o que equivale a 1,2 m/s do deslocamento do cursor.

Estima-se que a variação na quantidade de pulsos por milímetro observada na bancada esteja ligada as folgas mecânicas na construção da bancada e as limitações do microcontrolador utilizado.

A utilização dos pinos nos microcontroladores Arduino® para leitura de pulsos de encoder necessita da utilização de pelo menos um pino habilitado para interrupções de acordo com o fabricante (ARDUINO, 2022).

Com a bancada, foi possível escrever a primeira versão do código no microcontrolador enviando comandos específicos para o motor, além de testar as bibliotecas nativas do Arduino para criação de controladores PID.

Esta etapa do processo também deixou clara a necessidade da utilização de sensores de fim de curso nas extremidades da guia linear para evitar a fuga do cursor ou sobrecarga do motor em caso de perda da referência do encoder, falta de energia ou mal funcionamento de quaisquer elementos do atuador.

4.3. Montagem da estrutura passiva

A próxima etapa foi a construção da estrutura metálica que sustenta os atuadores e a plataforma móvel do robô. A estrutura é composta basicamente de perfis metálicos em alumínio com trilhos para passagem de elementos de fixação como presilhas, parafusos e porcas-martelo.

A estrutura foi construída sobre quatro rodízios articulados para facilitar a movimentação da estrutura no Laboratório. A base da estrutura foi projetada para comportar as esteiras abaixo dos pés do paciente, fornecer o suporte adequado ao seu peso passivo e posicionar adequadamente os módulos de atuação de acordo com a configuração projetada.

Durante esta etapa, verificou-se a necessidade de acrescentar um reforço estrutural nas barras dos atuadores 1 e 5. Portanto, foram adicionadas duas barras adicionais abaixo das extremidades de ambos. Estas mesmas barras foram prolongadas posteriormente para se adicionar a uma barra superior central para ser usada como suporte para a mola central.

Também se verificou que as posições dos módulos de atuação poderiam ser alteradas para diminuir o espaço total ocupado pela estrutura e evitar colisões entre as barras de cada atuador. Sendo assim, os atuadores 2 e 4 foram montados com guias e mecanismos na parte interna da estrutura, e o atuador 3 foi montado invertido, com o mecanismo na parte inferior de sua barra de suporte, conforme Figura 50.

A montagem parcial da estrutura é representada na Figura 50.

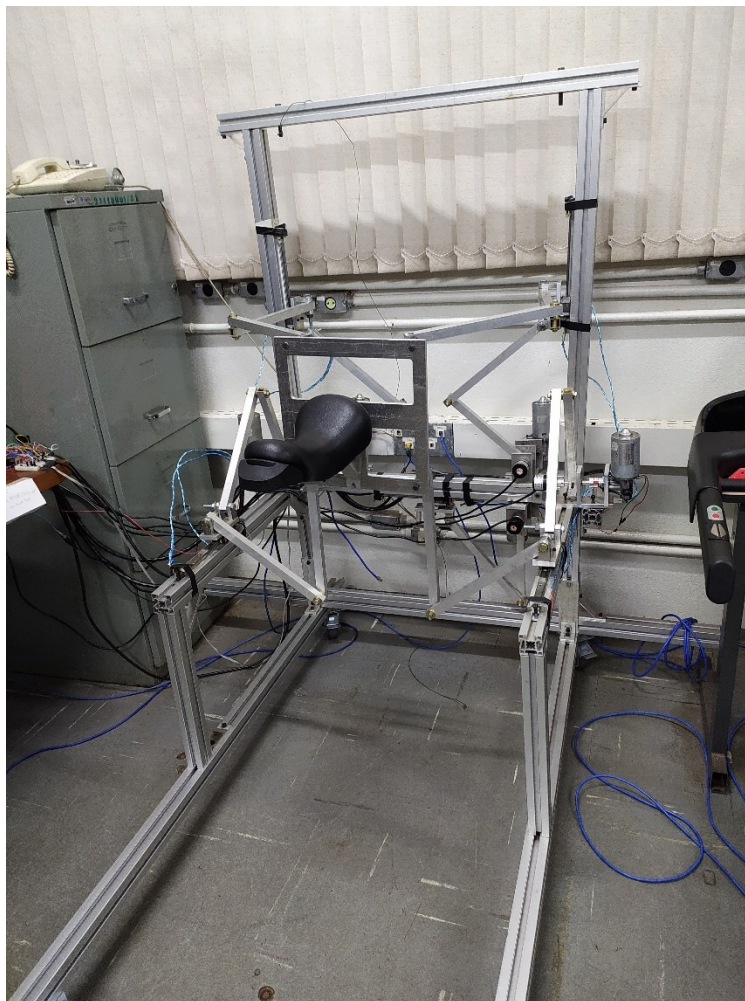


Figura 50 – Montagem parcial do módulo de suporte ativo durante a elaboração do sistema de controle.

Os módulos dos motores DC foram instalados utilizando retalhos menores dos perfis estruturais, priorizando o alinhamento das guias lineares, conforme Figura 51.

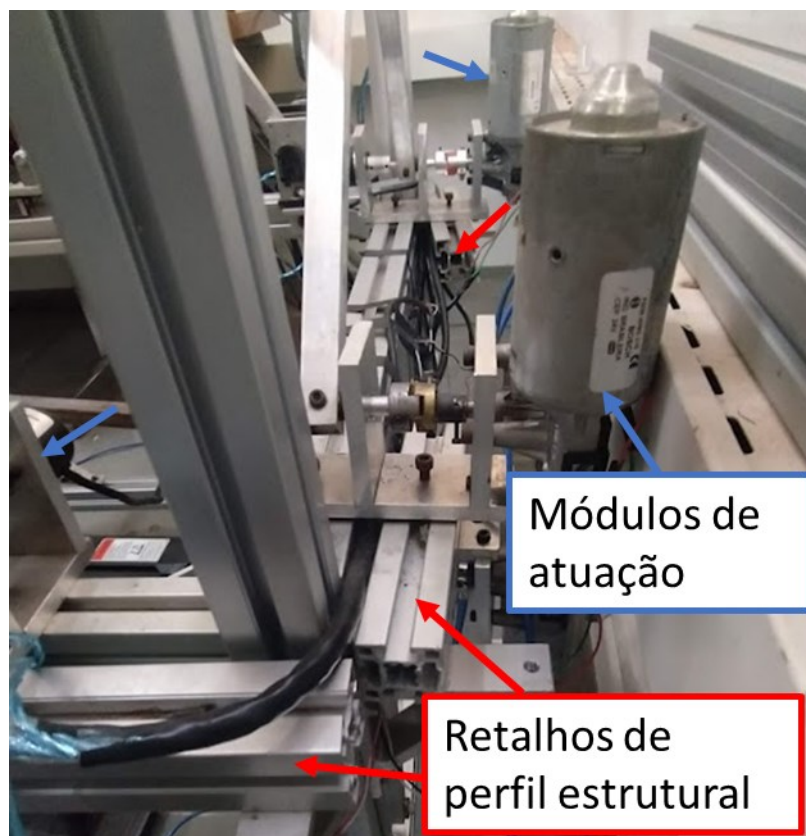


Figura 51 – Detalhe da instalação dos módulos de atuação, utilizando retalhos para posicionamento do suporte do motor DC e priorizando o alinhamento das guias lineares em relação à estrutura completa.

A plataforma móvel foi construída utilizando uma chapa de alumínio estrutural de 5 mm de espessura, seguindo as mesmas medidas utilizadas na modelagem matemática apresentada no Capítulo III.

Foram realizados recortes internos na chapa para reduzir o seu peso passivo, e acrescentadas as furações necessárias para instalar o selim próximo ao ponto P, Figura 34, e conectar as juntas universais previstas para a estrutura. Esta conexão foi realizada com o auxílio de cantoneiras, de forma a manter a configuração especificada na modelagem, conforme apresentado na Figura 52.

Por fim, as molas foram adicionadas na estrutura, realizando-se a furação necessária nos perfis estruturais, e utilizando parafusos como pinos de suporte. As molas foram posicionadas próximas às guias lineares dos motores 2 e 4, e uma acima da plataforma móvel, sendo conectadas à estrutura passiva através de cabos de aço com abraçadeiras rosqueadas tipo U. Dessa forma, é possível ajustar o nível das molas através das abraçadeiras para se adaptar a configuração da estrutura a diferentes alturas.

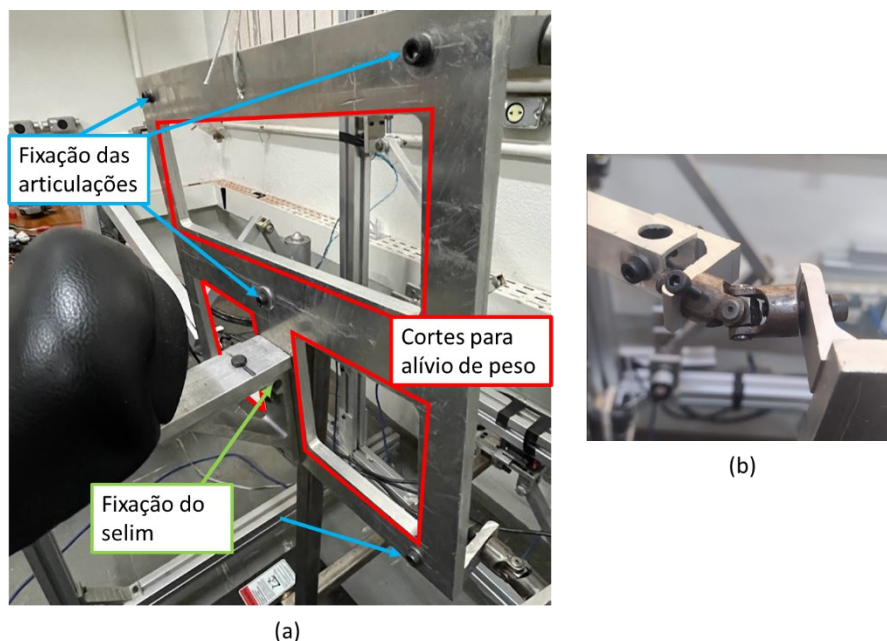


Figura 52 – Detalhe da construção da plataforma móvel. (a) Destaque do posicionamento das articulações, selim e cortes para alívio de peso; e (b) detalhe da construção de uma das articulações utilizando junta universal.

4.4. Montagem dos circuitos e módulos de controle da estrutura

Com a montagem da estrutura passiva e dos módulos de atuação completos, a próxima etapa foi a montagem da central de controle do suporte de peso ativo. Esta central foi projetada para integrar todos os sinais elétricos de entrada e saída da estrutura, os drivers de tensão para os motores DC e o microcontrolador.

Inicialmente, durante a realização dos testes realizados na etapa 4.2, a montagem elétrica foi realizada de forma reduzida e experimental, incluindo apenas um motor DC, um driver de tensão, um encoder e dois sensores de fim de curso, além do microcontrolador conectado a um computador via conexão serial USB.

Esta montagem inicial foi utilizada como base para expandir o projeto para os demais atuadores. Esta expansão foi projetada utilizando o software CAD/CAE para circuitos elétricos Fritzing[®], conforme apresentado na Figura 53. A partir do modelo CAD obtido, construiu-se o circuito inicial utilizando jumpers e protoboards para se realizar as conexões. Os sinais de entrada e saída dos componentes foram transmitidos por meio de cabos multivias.

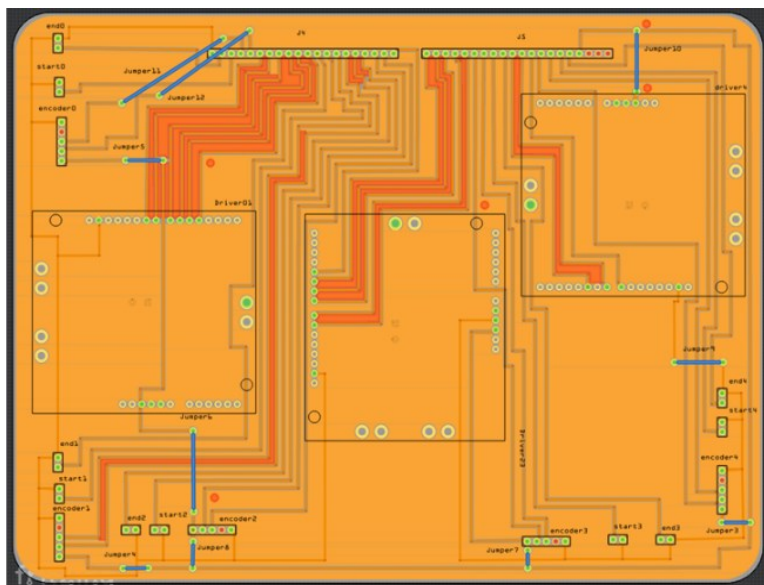


Figura 53 – Projeto CAD/CAE criado para a central de controle da estrutura para PCI.

A partir desta montagem inicial, o código-base para operação dos sensores e atuadores da estrutura foi elaborado, focado na movimentação básica de cada um dos motores e possibilitando a identificação de falhas na montagem.

Após a validação operacional do circuito montado sobre os Protoboards, elaborou-se o esquema para uma PCI a partir do projeto CAD construído anteriormente. Os protoboards e jumpers foram, então, substituídos pela placa confeccionada, apresentada na Figura 54.

Para facilitar manobras e manutenção no circuito projetado, e possibilitar a confecção da PCI utilizando uma placa de fenolite de face simples, a conexão entre os componentes de placa e o microcontrolador foi mantida com o uso de jumpers.

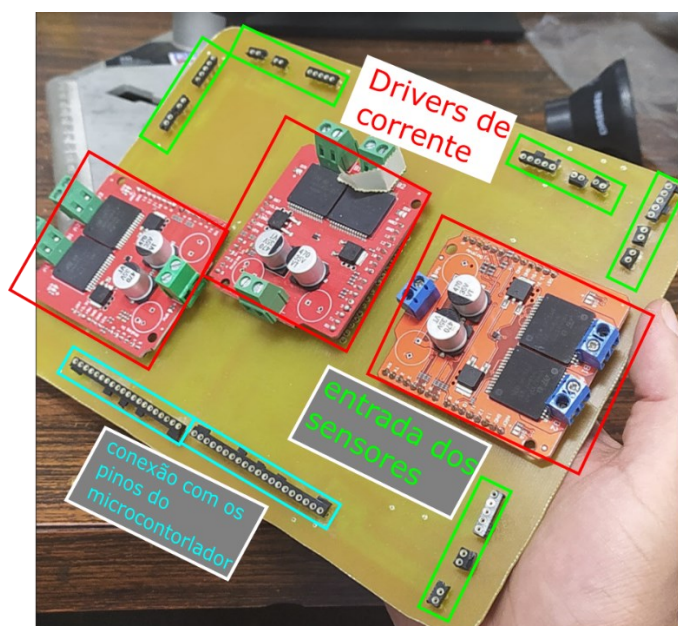


Figura 54 –PCI da central de controle do suporte de peso ativo, já com pin headers e drivers de tensão instalados.

4.5. Configuração dos atuadores

Com os circuitos elétricos em funcionamento, os controladores para os atuadores foram implementados. O esquema de controle de cada um dos atuadores é representado na Figura 55.

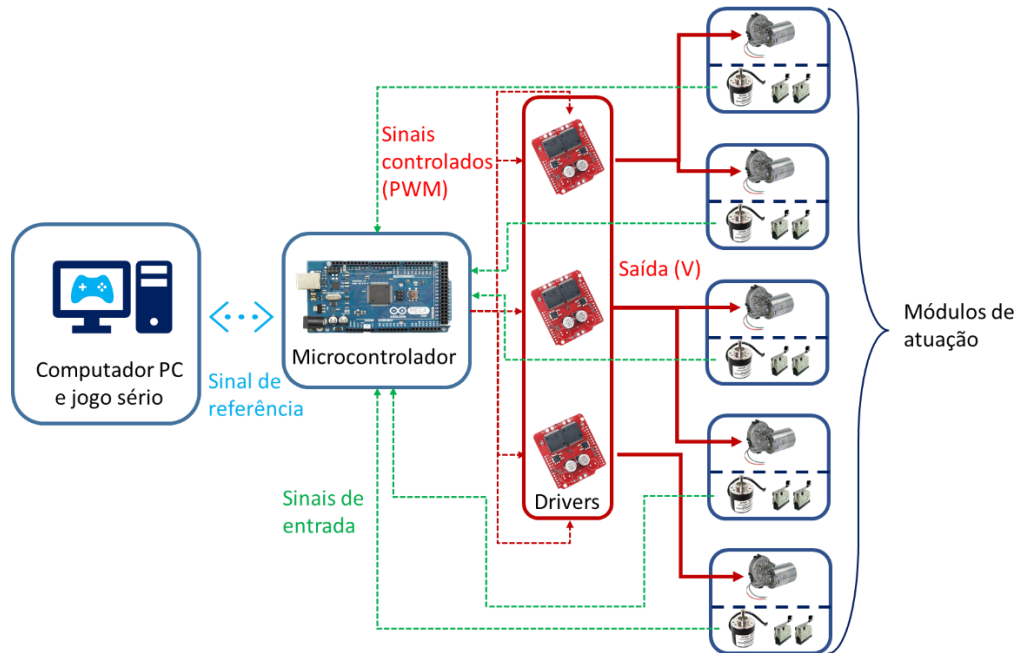


Figura 55 – Esquema de controle aplicado no protótipo de suporte de peso ativo.

Para cada módulo, foi especificado um controlador dividido em duas partes: um controlador de posição do tipo PID, afinado usando um sistema de otimização baseado em um algoritmo de evolução diferencial; e um controlador “*Assist-as-needed*” (AAN) para suprimir a ação do PID de forma dinâmica, aplicando-a apenas quando necessitado pelo paciente.

Para cada encoder aplicado, utilizou-se um pino habilitado para interrupção e de uso geral. Conforme descrito pela documentação oficial do microcontrolador, o cenário ideal para controle determinístico deve contar com dois pinos com interrupção para cada canal de cada encoder. Devido a indisponibilidade de pinos suficiente, o controle utilizou uma configuração não-determinística, onde a contagem dos pulsos está sujeita a perdas de pulsos.

Funcionamento básico de um PID

Um controlador PID é um sistema digital que manipula a saída para os atuadores, que neste caso é a tensão elétrica de alimentação dos motores enviada pelos drivers de corrente (FADALI; VISIOLI, 2013). Esta saída é modificada a partir do sinal de referência em função de 3 termos: um proporcional (K_p); um integral (K_I); e outro derivativo (K_D), conforme descrito na Eq. (33).

$$u[kT] = K_p e[kT] + K_I \int_0^T e[kT] dT + K_d \frac{d}{dT} e[kT] \quad (33)$$

Onde $u[kT]$ representa o sinal discreto de saída do controlador PID em função do tempo de amostragem T e $e[kT]$ é o erro do sistema calculado em função da referência do sistema. Com o uso de controlador, é possível fazer com que o atuador atinja requisitos de projeto, como, por exemplo, erro de posição máximo ϵ , tempo de acomodação t_s , overshoot máximo etc. (FADALI; VISIOLI, 2013).

Contudo, devido à falta de determinismo implicada pelas limitações do microcontrolador utilizado, a lei de controle da Eq. (33) está sujeita a variações no período de amostragem T . Dessa forma, o uso de um otimizador por evolução diferencial foi necessário para garantir parâmetros estáveis que compensassem este problema.

No caso do suporte ativo de peso, o controle dos atuadores foi elaborado seguindo os seguintes critérios:

- Tempo de acomodação t_s menor que 0,2s.
- Overshoot próximo de zero.
- Erro estacionário zero.

Primeiro, o PID foi especificado para cada atuador individualmente. Como o protótipo foi construído de acordo com a disponibilidade de materiais, seu sistema elétrico e mecânico não possui parâmetros e requisitos funcionais necessários para se estimar uma função transferência de forma precisa.

Logo, para se estimar os parâmetros K_p, K_I e K_D de forma eficiente, utilizou um sistema otimização meta-heurística baseado em um algoritmo de evolução diferencial (GONÇALVES; CARVALHO; LOBATO, 2016a, 2016b). Para utilizar este sistema, foram realizados diversos ensaios de forma automática no sistema, analisando a resposta do sistema a uma entrada fixa conhecida e colhendo a telemetria de resposta correspondente de cada execução.

O algoritmo aplicado, de forma resumida, funciona associando os parâmetros de cada PID de forma aleatória, simulando o princípio da Seleção Natural, onde os valores parâmetros funcionam como “genes”, as combinações destes valores são os “indivíduos” e são avaliados em função de uma Função Objetivo (FO), que determina a aptidão de cada indivíduo ao meta-ambiente (GONÇALVES; CARVALHO; LOBATO, 2016a, 2016b). Os indivíduos são combinados de forma aleatória, simulando o processo de reprodução, onde os mais aptos possuem maior chance de se reproduzir. Um fator aleatório de mutação também é aplicado a cada combinação para aumentar a aleatoriedade do processo.

Os indivíduos utilizados neste algoritmo possuíam 3 genes, onde cada um representa uma das constantes do controlador (K_p, K_I e K_D), os quais foram cruzados utilizando *binomial crossover* aleatório com probabilidade de cruzamento (CR) igual a 80% e taxa de perturbação igual a 0,5%.

O fluxograma generalizado deste método de otimização é apresentado na Figura 56.

API de otimização

Para aplicar esta solução diretamente nos módulos aplicados, foi necessário integrar o código-base do algoritmo de evolução diferencial, implementado no software MATLAB®, com o código embarcado no microcontrolador, desenvolvido com o Arduino IDE, através de uma Interface de Comunicação entre Aplicações (API).

A API desenvolvida foi desenvolvida para operar usando a comunicação serial USB, onde o microcontrolador funciona como servidor e o código MATLAB como cliente.

O programa cliente envia para o servidor como entrada os parâmetros PID de um dado indivíduo. O microcontrolador, então, aplica os dados recebidos ao controlador sendo otimizado e realiza um deslocamento de teste no atuador, enquanto coleta dados dos sensores. Os dados dos sensores são avaliados em função dos requisitos de projeto de tempo de acomodação, erro e Overshoot, gerando uma pontuação. Este valor é então retornado pela API para o algoritmo de evolução diferencial, que o associa ao respectivo indivíduo.

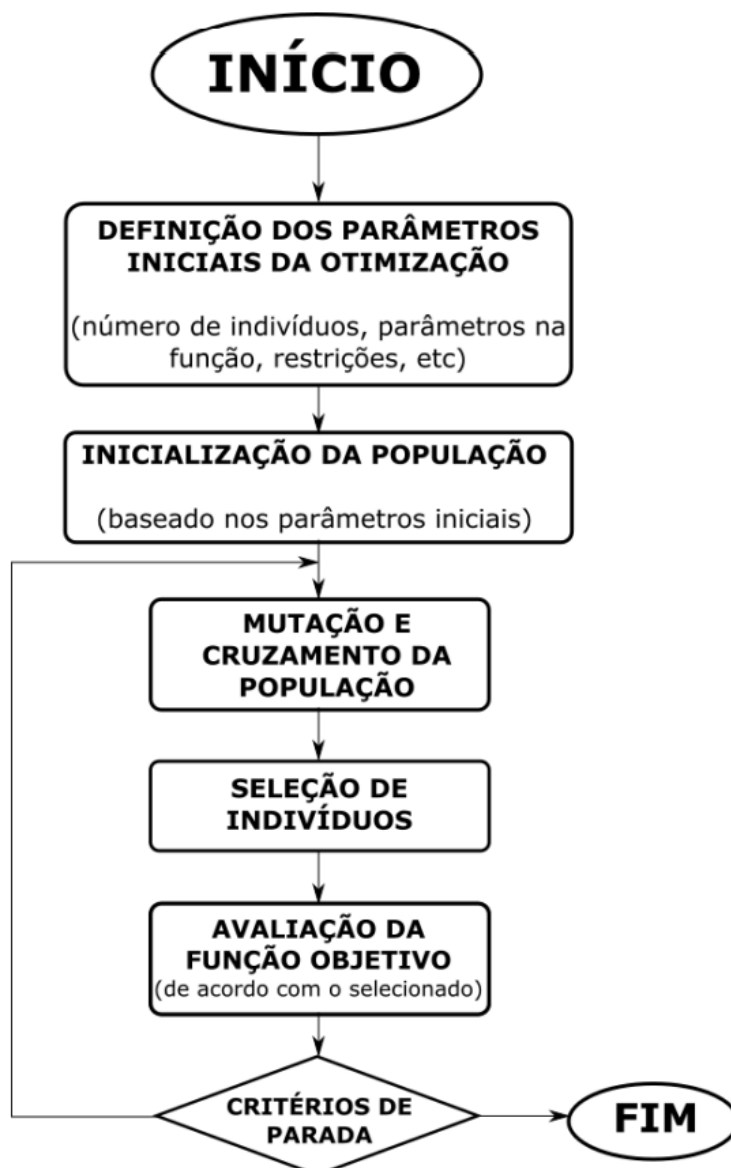


Figura 56 – Fluxograma detalhando o funcionamento do algoritmo de evolução diferencial (RODRIGUES, 2017).

A arquitetura da API descrita acima é representada na Figura 57.

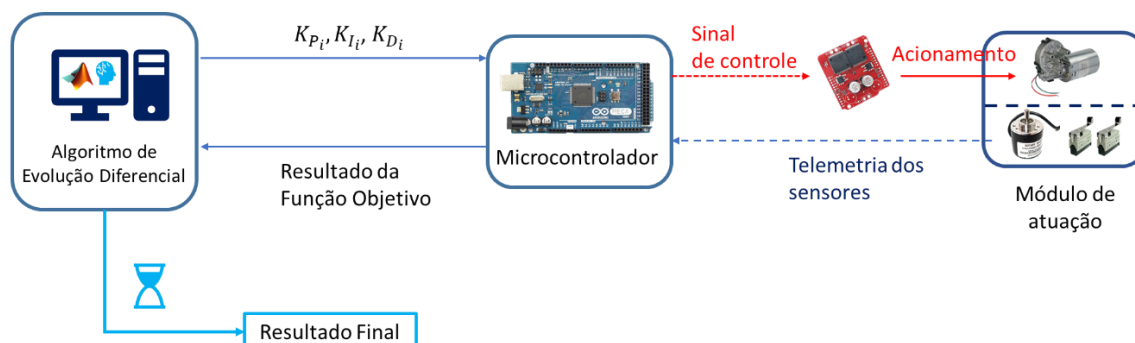


Figura 57 – Arquitetura desenvolvida para a API de otimização dos atuadores.

Controladores PID obtidos

O processo de otimização dos controladores foi realizado para cada controlador separadamente, seguindo os seguintes parâmetros:

- Tamanho da população (número de indivíduos): 30.
- Número máximo de iterações (máximo de gerações): 30
- Taxa de mutação: 0.5%
- Probabilidade de cruzamento: 80%
- Limites laterais das variáveis:
 - K_P : de 0,1 até 4,0.
 - K_I : de 0,0 até 1,0.
 - K_D : de 0,0 até 2,0.

Após otimizados, obteve-se os resultados apresentados na Tabela 7. Todos os códigos utilizados neste processo estão disponíveis no Apêndice.

Tabela 7 – Resultados obtidos para os parâmetros dos controladores após a otimização.

Motor	K_P	K_I	K_D
1	2,78460	0,00000	0,12855
2	1,68340	0,00000	0,06866
3	3,07820	0,00000	0,08302
4	1,45780	0,38957	0,09438
5	4,00000	0,00000	0,10293

Adicionando o componente AAN

Para implementar o controle AAN, aplicou-se uma parcela adicional a cada controlador de acordo com o diagrama de blocos apresentado na Figura 58, capaz de gerar a uma zona onde sua ação é suprimida, denominada “*dead-zone*” (ASL; NARIKIYO; KAWANISHI, 2019). Esta zona consiste em uma faixa de valores do erro do sistema e é definida empiricamente em função das características do sistema controlado, que no caso da reabilitação assistida, representa a região onde o paciente exercerá ação motora sem a necessidade do controlador.

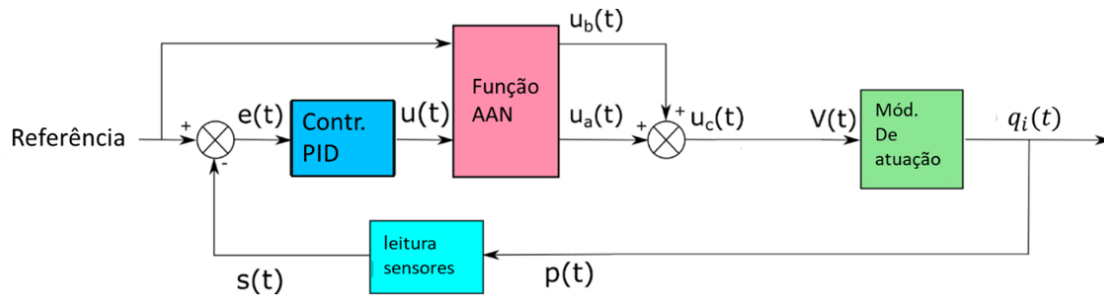


Figura 58 – Malha de controle aplicada nos atuadores, incluindo um controlador PID e uma função de AAN.

Em contrapartida, sempre que o erro do sistema exceder a *dead-zone*, a ação de controle do controlador será novamente aplicada, representando a ausência de ação motora do paciente e implicando a necessidade de intervenção da estrutura.

A implementação da *dead-zone* é feita através de uma combinação da ação do controlador com uma função tangente hiperbólica. A escolha dessa função garante a transição suave entre a *dead-zone* e as regiões de ação do controlador, e ponderam a ação do mesmo para evitar movimentos bruscos durante a sessão de treinamento. Para evitar instabilidades no sistema quando o erro estiver dentro da *dead-zone*, parte da referência do sistema é passada diretamente à ação de controle, ação conhecida como *feedforward*. Dessa forma, a ação de controle do sistema é descrita na Eq. (34) e representada graficamente na Figura 59.

$$u_a[kT] = K_p \tanh(\gamma u[kT]^3) + K_f \tanh(\gamma r[kT]^3) \quad (34)$$

Onde K_p representa o ganho proporcional do sistema, $u[kT]$ é o sinal de controle do controlador PID, γ é a constante utilizada para delimitar a *dead-zone*, K_f é o ganho associado a parcela em *feedforward* do sistema, e $r[kT]$ é a referência de posição do sistema.

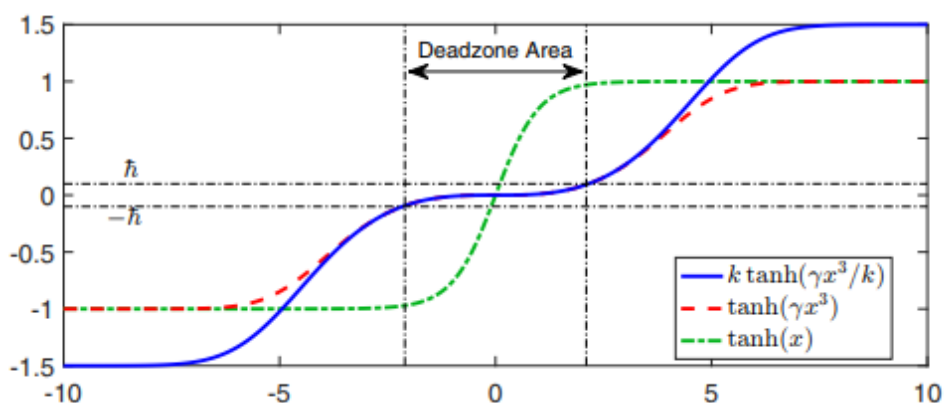


Figura 59 – Representação da *dead-zone* gerada pela função tangente hiperbólica: Em verde, a função aplicada diretamente; em vermelho a função combinada com uma função cúbica; em azul, a função combinada com uma constante para ganho (ASL; NARIKIYO; KAWANISHI, 2019).

4.6. Comunicação com PC

Ao concluir o passo descrito na seção anterior, o sistema está com os controladores definidos. Agora, o software do microcontrolador foi atualizado para comportar outra API, projetada para integrar as entradas e saídas da central de controle com o jogo sério, descrito no Capítulo V.

Para esta nova API, construiu-se dois modos de funcionamento: o modo operacional, que será utilizado em conjunto com o jogo sério; e o modo de manutenção, que será utilizado com um terminal serial para se verificar a integridade dos sinais dos sensores e do comportamento dos motores. Um diagrama destes modos de funcionamento é apresentado na Figura 60.

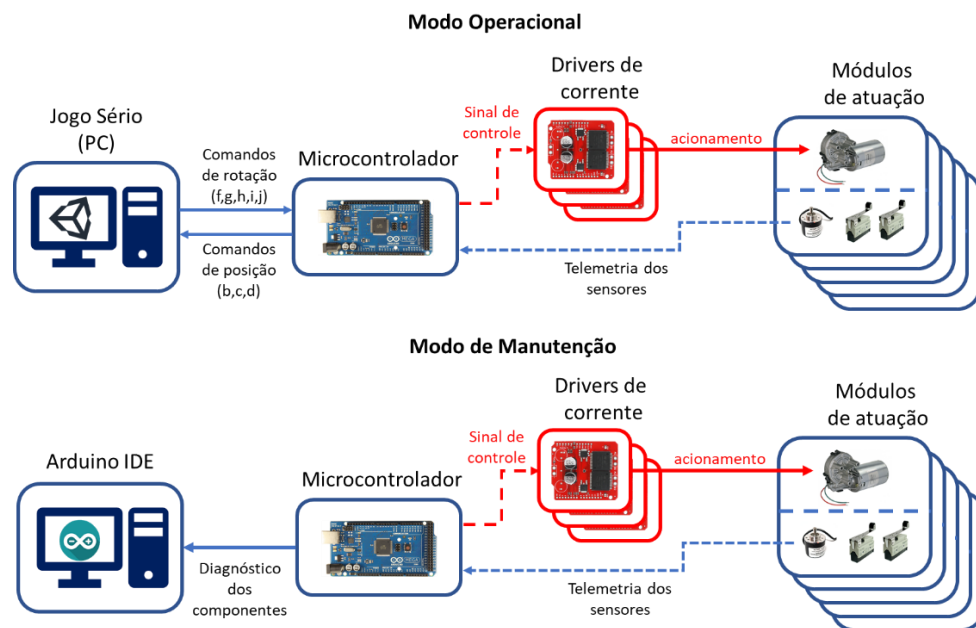


Figura 60 – Diagrama de funcionamento dos modos da API de comunicação da central de controle.

Durante o modo operacional, a estrutura recebe comandos simplificados para realizar rotações na plataforma, para simular desníveis de forma controlada, e enviará ao jogo sério dados da posição do paciente, que serão replicados para o ambiente virtual.

Além dos comandos de entrada, quando configurado neste modo, o microcontrolador analisa o deslocamento ao longo do eixo Y, monitorado pelo encoder do motor 3, e envia comandos ao jogo sério, também codificado por caracteres, de forma a posicionar o personagem principal do jogo na mesma região física do paciente.

Os comandos utilizados em ambas as aplicações foram definidos conforme a Tabela 8.

Tabela 8 – Comandos de entrada e saída da API de operação das ações de rotação da estrutura.

Comando	Caractere utilizado	Origem	Destino
---------	---------------------	--------	---------

Rotação esquerda no plano coronal	“g”	Jogo Sério	Central de controle
Rotação direita no plano coronal	“h”	Jogo Sério	Central de controle
Rotação esquerda no plano transversal	“i”	Jogo Sério	Central de controle
Rotação direita no plano transversal	“j”	Jogo Sério	Central de controle
Posição neutra	“f”	Jogo Sério	Central de controle
Posicionar à esquerda	“b”	Central de controle	Jogo Sério
Posicionar ao meio	“c”	Central de controle	Jogo Sério
Posicionar à direita	“d”	Central de controle	Jogo Sério

Quando ajustado para o modo de manutenção, o microcontrolador executa uma rotina de diagnóstico, utilizada para identificar possíveis falhas nos sensores de fim de curso, na contagem de pulsos dos encoders, na direção de movimentação dos motores e possíveis controladores PID descalibrados.

Nesta rotina, cada componente é analisado de forma independente, seguindo uma ordem priorizando a segurança e integridade do equipamento. Primeiro, a estrutura monitora os sensores de início e fim de curso, onde o operador deve acioná-los manualmente, e verificar se houve resposta do sistema.

Depois, o operador deverá manipular manualmente os atuadores enquanto o microcontrolador verifica se houve sinal adequado dos encoders. Com todos os sensores funcionando normalmente, o microcontrolador aciona os motores para executar movimentos básicos nos módulos e verifica se os sensores responderam de forma esperada. Quando tudo está funcionando de forma correta, o microcontrolador retorna para o modo operacional.

4.7. Testes de movimentação

Após seguir todos os passos descritos nas seções anteriores, a estrutura está pronta para executar testes de movimentação. Primeiramente, testou-se executar as movimentações propostas na simulação apresentada na Figura 46. Foi possível verificar que a mobilidade proposta foi atingida, conforme observado na Figura 61.

Antes de realizar os testes, a central de controle foi configurada para o modo de manutenção, e todos os sensores e motores foram verificados. Após isso, a estrutura entrou em modo operacional, e os comandos de movimentação foram enviados manualmente utilizando o

monitor serial do software Arduino IDE, também utilizado para o desenvolvimento do código-fonte do microcontrolador.

Com a movimentação validada, utilizou-se um boneco antropométrico de madeira, com 1,80 m de altura, para se testar a movimentação da plataforma móvel com a influência da carga de uma pessoa. Primeiro, posicionou-se uma esteira abaixo da plataforma, para se ajustar a altura da plataforma e a tensão das molas de suporte de acordo com a altura do quadril do boneco quando de pé sobre a esteira.

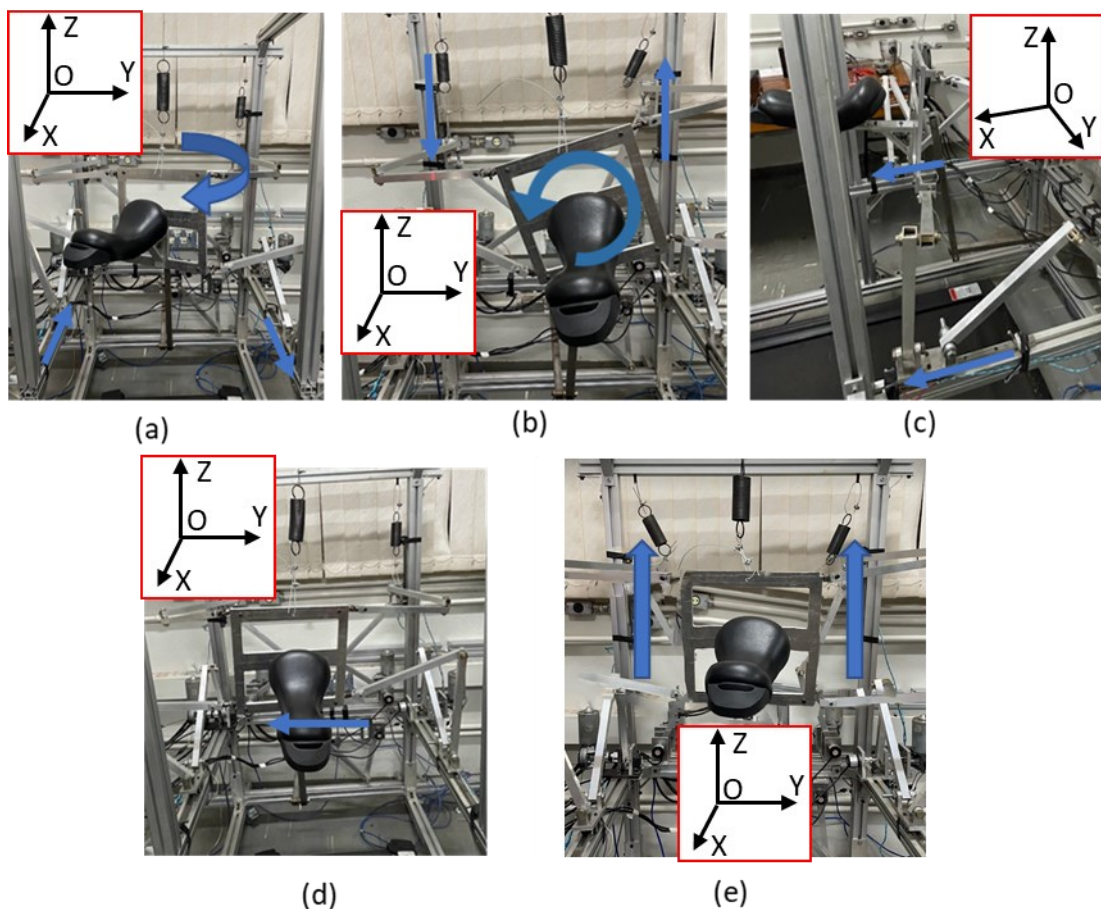


Figura 61 – Testes de movimentação da plataforma móvel do protótipo de suporte ativo de peso. (a) Rotação no plano transversal (em torno do eixo OZ); (b) Rotação no plano frontal (em torno do eixo OX); (c) Translação no eixo OX; (d) Translação no eixo OY; e (e) Translação no eixo OZ.

Depois disso, o boneco foi fixado na plataforma com auxílio de cabos de aço e parafusos. Este suporte adicional será posteriormente substituído por um colete apropriado a pacientes para evitar quedas durante a operação do equipamento.

Com o boneco posicionado conforme apresentado na Figura 62, realizou-se então um novo teste de movimentação da estrutura. Foi possível observar uma maior dificuldade da

movimentação da plataforma com o boneco fixado a ela, entretanto todas as manobras solicitadas a central de controle foram executadas corretamente.

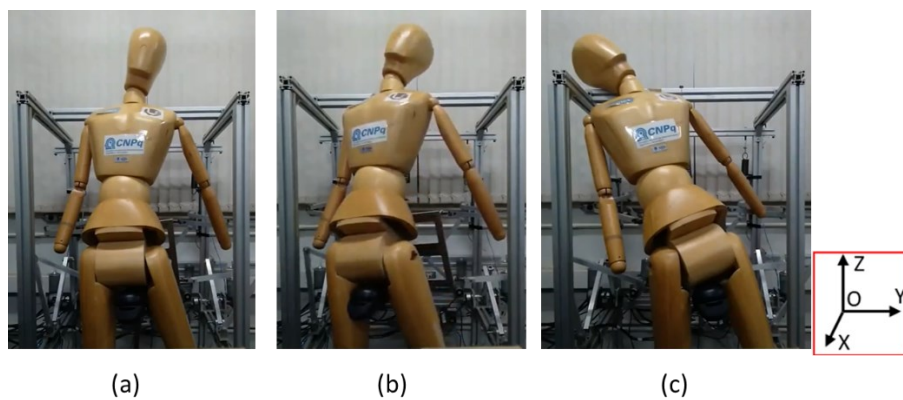


Figura 62 – Boneco antropométrico de madeira, posicionado sobre uma esteira e fixado à plataforma móvel da estrutura para testes de movimentação: (a) posição neutra; (b) rotação em torno do eixo Z; (c) rotação em torno do eixo X.

CAPÍTULO V PROTÓTIPO PARTE 2: JOGO ELETRÔNICO

Para a criação do jogo, primeiramente considerou-se quais deveriam ser os requisitos presente neste de modo que atendesse as necessidades terapêuticas do paciente, mas também fosse funcional e divertido. Desta forma, o desenvolvimento do projeto busca incentivar os movimentos necessários para a reabilitação do paciente, garantir a interação plena com a estrutura HOPE-G e garantir a segurança do usuário.

Assim, foi decidido pelo desenvolvimento de um jogo 3D no estilo “*Endless Runner*”, em que o personagem se move para frente de forma automática e o usuário controla os movimentos laterais ao se deslocar lateralmente sobre a estrutura. Para o desenvolvimento deste jogo, utilizou-se a plataforma de criação de jogos Unity 3D devido a maior necessidade de se simular uma movimentação espacial compatível com o treino da marcha humana na estrutura.

5.1. Criação do cenário e obstáculos

Primeiramente, foi necessário a programação de um script que gerasse uma pista infinita para que o personagem corra. Para isso, dividiu-se a plataforma de corrida em diversos segmentos, de modo que inicialmente apenas uma certa quantidade de segmentos pré-determinada é criada na tela. Conforme o jogador avança no mapa o script automaticamente exclui os segmentos que não estão mais visíveis e para garantir que o jogo não sobrecarregue a memória do sistema com o passar do tempo acrescenta novos segmentos a uma taxa constante ao final do segmento. Dessa forma a pista permanece indefinidamente enquanto a sessão durar.

Além disso, um efeito de iluminação foi aplicado ao fundo da pista para que o usuário não perceba o acréscimo de segmentos que acontece constantemente. Este efeito é ilustrado na Figura 63, onde o item (a) o fim da pista é evidente, enquanto no item (b) isso não é possível, devido à névoa.

Nesse contexto, os segmentos foram previamente criados e salvos como recursos prontos, de forma que cada um tem seu estilo e obstáculos próprios. Assim, o script mencionado anteriormente fica também responsável por selecionar aleatoriamente um desses recursos e colocá-lo no caminho do personagem garantindo um cenário diversificado durante a jogabilidade.



(a)



(b)

Figura 63 – Comparativo: imagem (a) sem o efeito de névoa ao fundo e imagem (b) com o efeito.

O script responsável por gerenciar os segmentos também controla o surgimento de cubos bônus. Estes, por sua vez, são pequenos cubos verdes que caso o jogador consiga pegá-los concedem uma pontuação extra ao total. Assim, o programa garante que os objetos não apareçam no caminho com uma frequência alta de forma que existe um espaçamento padrão entre eles, além de garantir o embaralhamento dos recursos pré-fabricados entre eles evitando percursos repetidos.

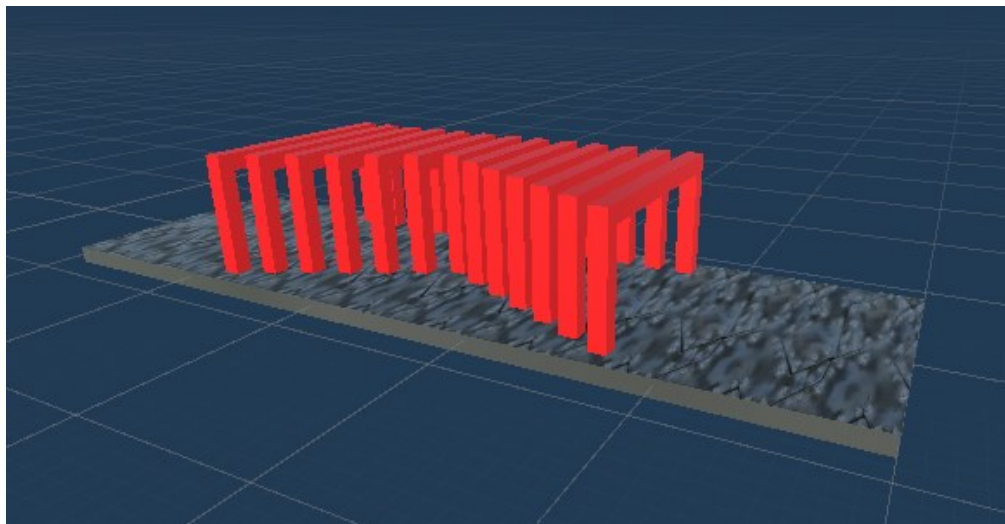


Figura 64 – Exemplo de recurso de segmento da pista.

5.2. Sistema de pontuação

O sistema de pontuação principal do jogo é proporcional ao tempo de sessão incrementado de forma contínua. Além dessa pontuação base foram implementadas pontuações extras que podem ser obtidas ao se coletar um item de bônus que aparece aleatoriamente no cenário e sistema de multiplicação da pontuação base que é acionado sempre que uma rotação da plataforma/cenário ocorre.

A principal fonte de pontuação extra do jogo está nos multiplicadores. Sempre que a plataforma sofre uma rotação, a pontuação é multiplicada por um fator que vai de duas até quatro vezes a taxa base de pontuação. O fator é incrementado a cada rotação e se mantém enquanto o jogador não colidir com os obstáculos do cenário.

Os cubos de bônus são pequenos objetos flutuantes, conforme apresentado na Figura 65. Eles aparecem sempre em uma das rotas que o personagem pode correr, porém nem sempre estará facilmente acessível podendo estar muito próximo de obstáculos. Dessa forma, é criada uma situação desafiadora onde o jogador deverá decidir se irá arriscar pegá-lo ou não. Além disso, os cubos são criados apenas depois de uma certa distância do início de forma que o usuário terá tempo para se familiarizar com os controles antes de ter a oportunidade de pegar um bônus.

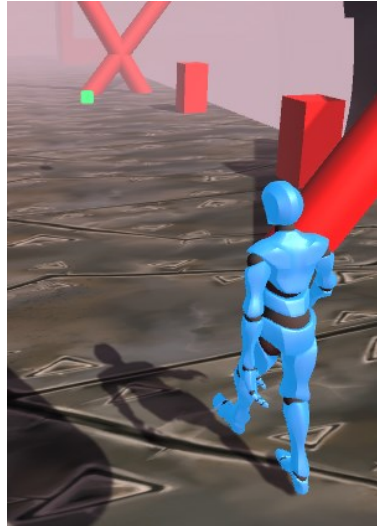


Figura 65 – Cubo bônus no cenário.

5.3. Personagem e câmera

Para o personagem principal, foi escolhido um modelo humanoide, Figura 66, e foram feitas nele as animações necessárias como a da marcha humana em linha reta, deslocamento lateral, espera e outras auxiliares.



Figura 66 – Personagem principal.

Em seguida, foi feita a programação da câmera do jogo que será o ponto de vista do jogador. Para isso criou-se um script para que a câmera siga o personagem a uma distância pré-determinada proporcionando uma visão em terceira pessoa ao usuário.

Além disso, um outro elemento de jogabilidade foi adicionado onde a plataforma móvel rotaciona em torno do eixo O_x ou O_z simulando uma condição adversa para treinar o equilíbrio do paciente. As rotações são controladas por um script no jogo e são acionadas aleatoriamente dentro de um intervalo entre 60 e 100 segundos. A câmera que acompanha o personagem principal realiza a mesma rotação enviada à estrutura para aumentar a sensação de imersão no jogo e um ícone animado surge no centro da tela para reforçar a ocorrência. Um exemplo da perspectiva do cenário durante a rotação é apresentado na Figura 67.



Figura 67 – Visão do jogador durante a rotação do cenário, destacando o diagrama de interface.

5.4. Interface com o usuário

O jogo possui três telas principais sendo elas o menu principal, a tela de sessão e a tela de encerramento. O menu principal contém o título do jogo, um botão de início da sessão, um de saída da aplicação, e uma caixa de texto onde pode ser inserido um tempo de sessão, conforme Figura 68. Caso a caixa seja deixada em branco, será assumido um tempo de sessão de 10 minutos. Ao clicar no botão de início, o jogo ativa a tela de sessão, onde uma contagem regressiva permanece no canto superior esquerdo da tela para que o paciente e o profissional da saúde possam acompanhar tempo restante. No canto superior direito fica disponível a pontuação atual do jogo.



Figura 68 – Menu principal do jogo.

Há também outros elementos na interface nesta tela como o ícone que indica para qual lado o cenário está rotacionando, quando esta situação acontece, e o fator multiplicador da pontuação. Foi adicionado também um botão de parada para permitir que o profissional que conduz a sessão pare o jogo e o equipamento antes do final da sessão, caso seja necessário. A interface principal do jogo é apresentada na Figura 67.

Por fim, ao se encerrar a sessão, o personagem principal encerra seu movimento e é exibida uma tela de encerramento com a pontuação final e a duração total da sessão. Nesta tela há um botão de encerrar a aplicação e outro para retornar ao menu principal, onde uma nova sessão pode ser iniciada.

5.5. Comunicação com a estrutura

Para interagir com a estrutura utilizou-se um script de controle para a interface entre a movimentação do personagem e o usuário. O script inicia a comunicação serial entre o Unity e o Arduino ao início da sessão através de uma porta USB, de modo que a cada 500 milissegundos o microcontrolador envia um caractere representando a posição ao longo do eixo O_y , medida utilizando o encoder rotativo do atuador q_3 .

Cada encoder da estrutura possuem de 500 a 600 pulsos por volta, com um espaço de trabalho de entre 150 e 200 pulsos. Definiu-se 3 faixas para movimentação do personagem, sendo uma à direita, uma ao centro e outra à esquerda da pista infinita.

Para evitar problemas de desalinhamento da referência da estrutura e do jogo, o controle do personagem, realizado através do monitoramento do módulo q_3 , é realizado de forma relativa.

Dessa forma, o microcontrolador interpreta apenas a velocidade \dot{q}_3 , sem levar em consideração a posição absoluta do módulo.

Caso o paciente esteja posicionado próximo às extremidades do cursor e seja bloqueado durante uma movimentação, o acionamento dos sensores de fim de curso de q_3 também irão enviar comandos de deslocamento ao jogo de acordo com a direção desejada.

Além de receber o sinal de controle de posicionamento do personagem, o script de interface também envia ao microcontrolador os seguintes comandos para alterar a rotação da câmera.

Sempre que a câmera sofre uma rotação, um sinal é enviado ao microcontrolador indicando como deve ser a rotação a ser aplicada na plataforma móvel. A movimentação é executada pelos módulos de controle, utilizando também uma função similar a apresentada na Equação (34), para garantir que caso o paciente consiga manter a posição por si próprio, não seja executada uma ação ativa dos motores.

Além disso, há outra ação assistiva implementada no jogo. O script de interface monitora o número de colisões por minuto que o personagem principal sofre com obstáculos. Caso esse número seja maior que 4 colisões por minuto, o jogo poderá enviar um comando para ativar a ação ativa do atuador q_3 para auxiliar o posicionamento do paciente na pista correta, caso seja necessário deslocar o personagem.

Esta funcionalidade já está implementada no jogo e está prevista futuras atualizações na central de controle da estrutura.

CAPÍTULO VI **PROTÓTIPO PARTE 3: TESTES PILOTO E RESULTADOS**

Após concluir a construção e configuração do protótipo do suporte ativo de peso e o desenvolvimento do jogo sério a ser integrado com a estrutura foi realizado testes experimentais para validar o funcionamento das funcionalidades previstas.

Os testes foram realizados no Laboratório de Automação e Robótica da Universidade Federal de Uberlândia, de acordo com as etapas listadas abaixo:

1. Testes de validação da estrutura;
2. Teste de integração jogo-estrutura utilizando boneco antropométrico;
3. Teste com participantes saudáveis;
4. Análise preliminar dos resultados.

Para realizar os testes com a estrutura desenvolvida foi necessário solicitar autorização para realização com testes em seres humanos junto ao Comitê de Ética em Pesquisa (CEP) da Universidade Federal de Uberlândia. A solicitação foi submetida e aprovada por meio do projeto de pesquisa 01305318.2.0000.5152, disponível nos Apêndices.

6.1. Testes de validação da estrutura

Para garantir que os requisitos determinados para a estrutura durante a fase de modelagem foram atendidos, propôs-se a realização de 4 experimentos. Nestes, os dados coletados devem trazer evidências da conformidade ou não-conformidade com os parâmetros críticos do projeto.

Estes testes são especialmente importantes, uma vez que diversos elementos do projeto elétrico e mecânico da estrutura necessitaram de adaptações orientadas à disponibilidade de recursos do Laboratório de Automação e Robótica.

Os quatro experimentos realizados foram:

- Verificação dos requisitos de controle;
- Validação da resolução dos módulos;
- Validação da Amplitude angular da plataforma móvel;
- Comparação da resposta do Modelo CAD e Protótipo.

Verificação dos requisitos de controle

O primeiro teste realizado buscou validar se o uso do algoritmo de evolução diferencial foi capaz de obter as constantes K_P , K_I e K_D capazes de atender os requisitos propostos no Capítulo IV. Este teste tornou-se especialmente importante devido a necessidade de se utilizar configurações não recomendadas do microcontrolador e de alimentação dos motores DC, o que poderia comprometer o correto funcionamento da estrutura.

Dessa forma, o experimento realizado consistiu em salvar os dados registrados no Arduino de entrada, saída e de tempo ao se realizar um deslocamento de 200 pulsos, e analisá-los utilizando de bibliotecas abertas de Ciência de Dados em Python e Jupyter Notebooks.

Os resultados obtidos são representados visualmente através de histogramas na Figura 69.

Especificações de controle foram atendidas

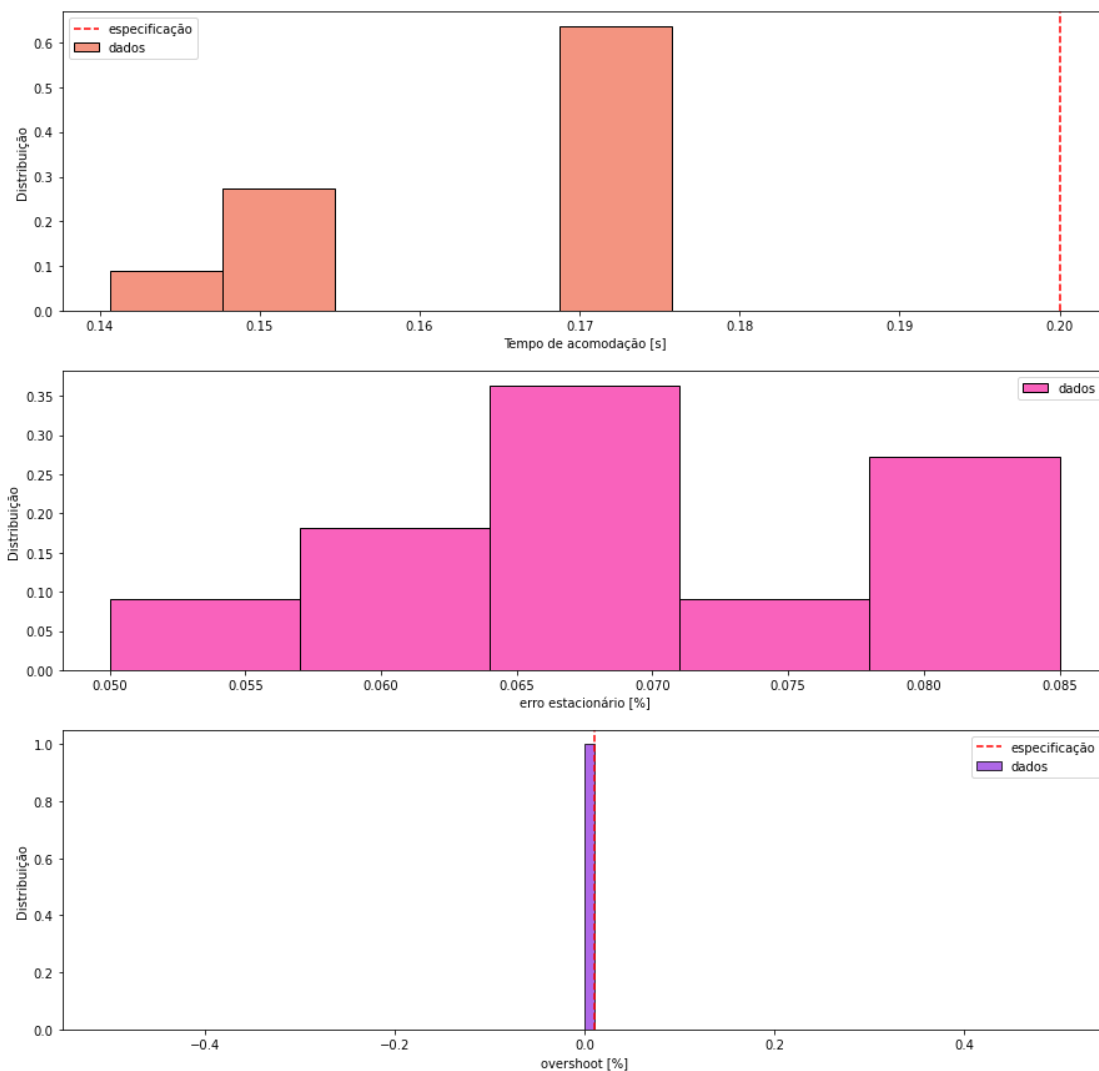


Figura 69 – Histograma representando a distribuição dos resultados obtidos nas 11 repetições do experimento de validação dos requisitos de controle. De cima para baixo, Resultados da análise do tempo de acomodação, erro estacionário percentual e Overshoot.

Através da análise visual dos dados coletados, foi possível notar que o tempo de acomodação especificado foi atendido de forma consistente, garantindo que sistema apresenta uma resposta adequada.

Embora o erro estacionário tenha sido especificado como nulo e as características mecânicas do sistema garantam uma função transferência com *Type Number* 1, o erro não pode ser eliminado completamente. Este erro precisará ser investigado em detalhes em trabalhos

futuros e pode estar ligado às limitações do microcontrolador e aos problemas construtivos da estrutura.

Contudo, considerando os aspectos construtivos da estrutura, o erro estacionário da estrutura ainda pode ser considerado tolerável e não causará problemas significativos na movimentação da plataforma, garantindo a funcionalidade do equipamento.

Além disso, o sistema não apresentou Overshoot em nenhuma das amostras. Isso também garante que os movimentos especificados pela estrutura não irão expor os pacientes a movimentos com amplitudes excessivas e evitará vibrações ao atingir a posição desejada.

Validação da resolução dos módulos

Outro experimento realizado buscou validar a relação entre o número de pulsos medidos nos encoders. Para realizar este experimento, aplicou-se um deslocamento de 200 pulsos em cada um dos motores com 10 repetições e mediu-se o deslocamento do cursor linear com o auxílio de uma régua fixada abaixo do trilho do cursor.

Os resultados obtidos são representados graficamente na Figura 70

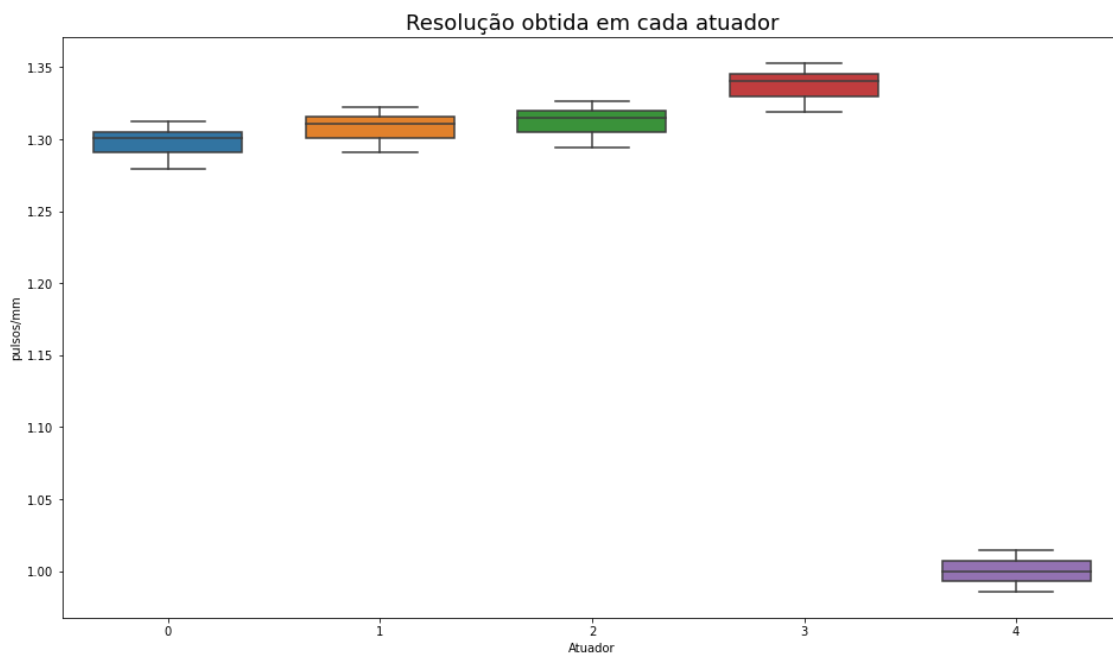


Figura 70 – Gráfico boxplot da resolução calculada para cada atuador, em pulsos/mm.

Destaca-se que como um dos atuadores utilizou um encoder de 500 pulsos por revolução, este obteve uma resolução inferior aos demais. A resolução apresentou uma variância considerável em todos os casos a qual está conectada com as limitações mecânicas e eletrônicas do protótipo.

Contudo, foi possível observar que as resoluções se mantêm próximas entre si nos atuadores com encoders de 600 pulsos, e dentro do valor esperado de acordo com o modelo matemático obtido no Capítulo III.

De posse de um valor estimado para a resolução, foi possível estimar também a velocidade linear máxima atingida pelos módulos, igual a aproximadamente 1,29 m/s.

Validação da amplitude angular da plataforma móvel

A seguir, foram realizadas medições utilizando um inclinômetro digital para definir o espaço de trabalho real da estrutura e compará-lo com o especificado no modelo matemático.

Neste experimento também foram realizadas 10 repetições para cada grau de liberdade angular medido. No caso dos graus de liberdade lineares, utilizou-se as medidas nominais das guias lineares aplicadas na estrutura como espaço de trabalho.

Os resultados são apresentados na Tabela 9.

Tabela 9 – Amplitude da plataforma móvel medida experimentalmente.

Grau de Liberdade	Amplitude da pélvis	Especificação	Experimental
Translação em X	-	$\pm 100 \text{ mm}$	$\pm 200 \text{ mm}$
Translação em Y	-	$\pm 200 \text{ mm}$	$\pm 200 \text{ mm}$
Translação em Z	-	$\pm 150 \text{ mm}$	$\pm 200 \text{ mm}$
Rotação em X	$\pm 5^\circ$	$\pm 5^\circ$	$-33,39^\circ + 35,45^\circ$
Rotação em Z	$\pm 25^\circ$	$\pm 25^\circ$	$-36,53^\circ + 32,55^\circ$

Ao observar os dados obtidos acima, é possível notar que o espaço de trabalho obtido no protótipo atente as especificações propostas considerando a mobilidade da articulação pélvica. Nota-se também que o design selecionado possibilita ainda amplitudes angulares acima da mínima, o que permitirá que a estrutura seja utilizada para realizar movimentos controlados para treino de equilíbrio.

Comparação da resposta do Modelo CAD e Protótipo

Por fim, realizou-se uma comparação com os pontos obtidos para um dos motores da estrutura em relação ao mesmo movimento realizado pelo modelo CAD desenvolvido. Neste experimento, realizou-se uma simulação do sistema mecânico deslocando 150 mm em 0,4 s e os pontos obtidos foram salvos. Depois, o mesmo movimento foi realizado 10 vezes em um dos motores da estrutura e a telemetria coletada.

A comparação entre os resultados simulados e experimentais é representada na Figura 71. Nesta, é possível notar que o protótipo possui uma inércia maior que a prevista no modelo CAD, o que faz com que o tempo de resposta inicial seja maior. Há também uma pequena oscilação no

final da trajetória e erro estacionário conforme observado no primeiro experimento. Porém, a forma das curvas experimentais se mantém satisfatoriamente próximas da curva simulada, sem desvios significantes.

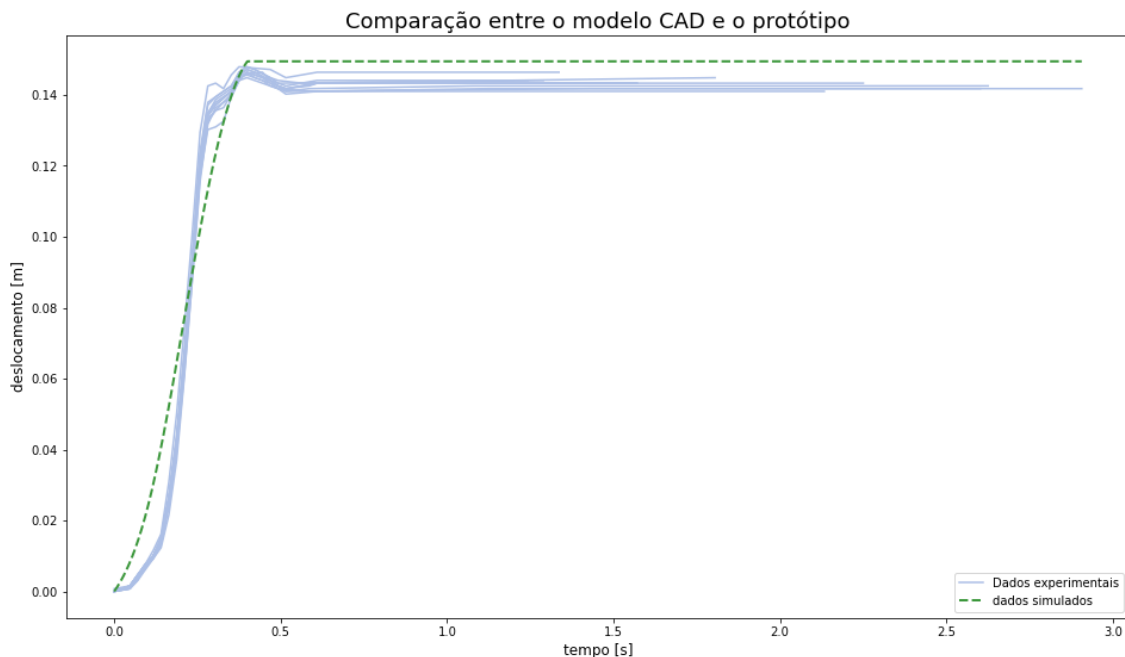


Figura 71 – Comparação entre os dados experimentais medidos em um dos atuadores do protótipo e os dados simulados em CAD.

Logo, com base nos resultados obtidos nos experimentos aqui descritos, pode-se concluir que a estrutura está válida para os testes práticos previstos com o boneco antropométrico e pacientes saudáveis.

6.2. Teste de integração jogo-estrutura utilizando boneco antropométrico

O primeiro teste realizado na estrutura já integrada com os jogos foi realizado com o auxílio do boneco antropométrico. Neste teste, o boneco foi movimentado manualmente com a estrutura e jogo em funcionamento e a esteira desligada. Os objetivos deste teste são:

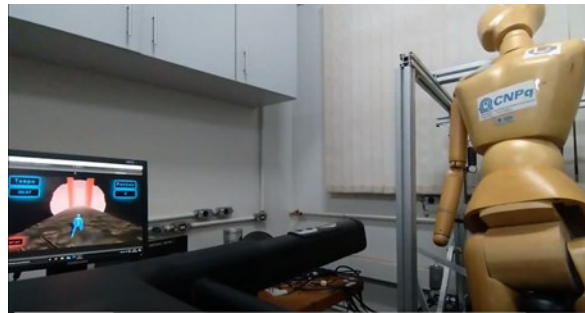
- Verificar a capacidade de controlar o personagem principal do jogo com a movimentação passiva da plataforma;
- Verificar a mobilidade da plataforma durante a execução do jogo;
- Avaliar a dificuldade do jogo;
- Identificar *bugs* e problemas na interface do jogo.

O teste foi então realizado em três sessões com durações entre 3 e 5 minutos. Uma fotografia realizada durante a execução de uma das sessões é apresentada na Figura 72.

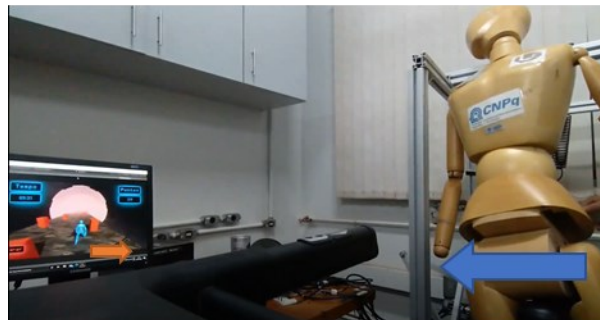
Após a realização das sessões de movimentação, o boneco foi retirado da estrutura.

Durante os testes verificou-se que era possível controlar a posição do personagem principal com movimentos laterais suaves, possibilitando percorrer o trajeto no jogo e desviar da maioria dos obstáculos sem muitas dificuldades.

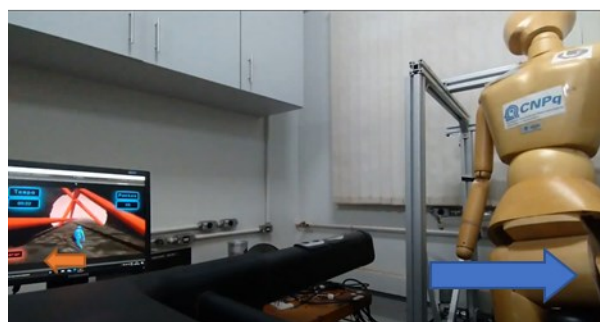
Notou-se que, apesar da mudança de posição do personagem também ocorrer ao se acionar os sensores de fim de curso, era necessário centralizar o atuador q_3 antes do início de cada sessão para evitar travamentos na plataforma ao longo do jogo e melhorar a experiência.



(a)



(b)



(c)

Figura 72 – Teste de integração jogo-estrutura utilizando boneco antropométrico: (a) posição central; (b) movimentação para deslocar o personagem para a direita da pista do jogo; (c) movimentação para deslocar o personagem para a esquerda da pista do jogo;

Foi possível notar também que a inércia imposta pelo atuador dificultava o movimento do cursor próximo às extremidades da guia. Dessa forma, ao se centralizar o atuador antes do início da sessão esse efeito era minimizado na jogabilidade.

Durante a execução do jogo, a plataforma foi acionada corretamente nos momentos de inclinação conforme previsto. Foi possível notar que durante as rotações da plataforma no plano frontal a jogabilidade foi pouco afetada, e durante as rotações no plano transversal, muito afetada, sendo necessário fazer movimentos mais amplos para movimentar o personagem, além de requerer um maior esforço para se manter o boneco equilibrado.

A maior parte dos obstáculos gerados durante as três sessões foram adequados, sendo que alguns precisaram ser remodelados, pois o tempo de ação para desvio estava muito curto, impossibilitando o jogador de desviar a tempo apenas com o movimento da plataforma.

Não houve instabilidades ou erros relevantes notados durante esta etapa de testes.

6.3. Teste com participantes saudáveis

Após a aprovação dos testes em seres humanos pelo CEP, convocou-se voluntários para utilizar a estrutura em funcionamento pleno, ou seja, com o jogo integrado e uma esteira em funcionamento.

Durante esta etapa inicial, foram realizados testes em dois voluntários saudáveis maiores de 18 anos, sendo um do sexo masculino e outro do sexo feminino, atendendo aos requisitos especificados para o Grupo de Pessoas Saudáveis (GPS) do Projeto de pesquisa aprovado no CEP, conforme descrito nos Apêndices.

Os objetivos propostos para esta rodada de testes são:

- Verificar a operação da estrutura em conjunto com a esteira em funcionamento, com a marcha humana em execução;
- Verificar o controle da posição do personagem principal do jogo, durante a execução da marcha humana;
- Avaliar a jogabilidade e dificuldade durante o uso real;
- Identificar falhas elétricas e/ou mecânicas durante a operação.

Foram executadas duas sessões de teste com cerca de 5 minutos de duração. Uma fotografia de uma das sessões de teste é apresentada na Figura 73.



Figura 73 – Sessão de teste com participantes saudáveis.

Antes da execução de cada teste, os participantes foram solicitados para ler e assinar um Termo de Consentimento Livre e Esclarecido (TCLE) conforme modelo apresentado nos Apêndices. Este termo descreve todos os procedimentos a serem executados durante os testes, os dados a serem coletados e os riscos envolvidos.

Após isso, a posição neutra da plataforma foi ajustada de acordo com a altura de cada participante através de regulagem nos cabos de aço das molas e abraçadeiras em U. Em seguida, os participantes se acomodaram na estrutura e a esteira foi ligada.

Os participantes caminharam sobre a esteira por alguns minutos sem o jogo em funcionamento para verificar se a velocidade estava adequada e o conforto da plataforma. Após confirmar as condições de início com os participantes, o jogo foi iniciado e a sessão de teste foi gravada em vídeo.

Ao final da sessão, as gravações foram interrompidas e os voluntários apresentaram as principais impressões sobre a estrutura e o jogo.

O primeiro feedback apresentado pelos participantes foi em relação a execução da marcha sobre a esteira com a plataforma. Ambos os participantes reportaram conseguir caminhar normalmente, com pouca ou nenhuma influência do selim sobre os movimentos realizados.

Durante a execução do jogo, os participantes reportaram dificuldades em movimentar adequadamente o personagem apenas movimentando-se para esquerda ou direita da esteira. As principais causas para este problema foram o espaço limitado para deslocamento lateral e a dificuldade de movimentar o atuador q_3 , apesar da atenuação da ação de controle pela função AAN.

A dificuldade na movimentação lateral também incorreu em desconforto ao tentar movimentar o personagem de acordo com o solicitado durante o jogo.

Quanto a jogabilidade do jogo, os participantes contribuíram para a correção da posição de alguns obstáculos e do cubo de bônus. A dificuldade observada em concluir o jogo foi associada aos problemas de acionamento citados anteriormente.

Nas ocorrências de inclinações da plataforma, os participantes reportaram que o deslocamento foi perceptível, sendo necessário ajustar os movimentos da marcha para a posição imposta. Durante os testes, observou-se que as rotações da plataforma foram menores em amplitude. Os principais fatores associados a esta ocorrência foram a tensão aplicada aos motores DC, que está abaixo da nominal, e a falhas na fixação dos acoplamentos flexíveis dos motores, indicando possível subdimensionamento da transmissão de torque.

Um trecho gravado dos testes em voluntários saudáveis pode ser acessado em <https://tinyurl.com/hope-g-tests>.

Sendo assim, com base nos resultados observados nos testes finais, foi possível obter importantes contribuições para melhorias na estrutura antes da execução das próximas etapas de testes, que incluem testes clínicos.

CAPÍTULO VII CONCLUSÕES

Nesta tese apresentou-se os procedimentos executados para o desenvolvimento da HOPE-G, uma estrutura de reabilitação da marcha humana baseada em três módulos distintos, onde o Módulo de Controle e Monitoramento Sem Marcadores foi desenvolvido em trabalhos anteriores. Os módulos da HOPE-G combinam a movimentação passiva dos membros inferiores por meio de esteiras bipartidas basculantes, e o suporte ativo do peso corporal utilizando uma nova estrutura paralela de 5 GDL para atuar sobre os graus de liberdade da articulação pélvica, e promover variações controladas na posição do corpo para treinar o equilíbrio durante a marcha.

Realizou-se uma ampla revisão bibliográfica das ciências associadas ao desenvolvimento da estrutura, incluindo os fundamentos da marcha humana, o atual estado da arte em estruturas de reabilitação da marcha humana, e a metodologia utilizada para síntese de estruturas paralelas.

Com base neste estudo, optou em realizar o desenvolvimento da HOPE-G seguindo princípios observados na estrutura MIT-Skywalker, utilizando um princípio de acionamento diferente para a esteira bipartida, utilizando cilindros pneumáticos, reduzindo sua altura em relação ao solo. Também se optou em mover o treinamento de equilíbrio para um suporte de peso ativo, desenvolvido como uma estrutura paralela da família de mecanismos *multipteron*.

No modelo matemático do Módulo de Movimentação por Esteira Bipartida foi possível aplicar uma solução que atende as melhorias especificadas pelos trabalhos anteriores do MIT-Skywalker (SUSKO, 2015).

A modelagem matemática do Módulo de Suporte Ativo de Peso Corporal a partir da arquitetura e a metodologia selecionadas possibilitaram a obtenção de equações com baixo nível de acoplamento e a análise de singularidades demonstrou um espaço de trabalho sem posições singulares. Dessa forma, a estrutura poderá ser controlada de forma mais simples e com atuadores mais independentes uns dos outros, apesar da configuração de cadeia fechada, e a otimização das dimensões foi pouco restringida quanto a posições singulares, o que melhora a eficiência dos materiais aplicados.

Cabe ressaltar também que a inovação associada ao tipo de estrutura aplicada no Módulo de Suporte Ativo de Peso Corporal possibilitou o depósito de uma patente de propriedade intelectual juntamente à Agência de Intelecto da UFU com número BR 10 2018 0703986.

Em seguida, as simulações computacionais apresentaram as primeiras evidências da validade dos modelos matemáticos construídos, uma vez que estes mostram de forma visual a mobilidade de cada um dos módulos, além de representar facilmente o funcionamento dos dispositivos para pesquisadores de outras áreas do conhecimento.

O protótipo foi desenvolvido para o Módulo de Suporte Ativo de Peso Corporal, onde foi necessário realizar adaptações para adequar o projeto à disponibilidade de recursos e materiais. Contudo, a montagem do protótipo foi finalizada e a mobilidade de todos os GDL da estrutura

pôde ser verificada e validada, de acordo com os resultados observados nas simulações computacionais.

De posse de um protótipo funcional, realizou-se o controle dos atuadores da estrutura, utilizando o mesmo processo de otimização por evolução diferencial para se obter os melhores parâmetros de controle para cada atuador. Dessa forma, foi possível compensar não-linearidades e fatores não previstos durante o processo de construção dos módulos atuadores e controlar os motores. Além disso, foi aplicada uma função AAN para possibilitar a ação motora do paciente sem a interferência do equipamento quando não necessário.

O protótipo também foi integrado a um jogo sério, desenvolvido especificamente para funcionar em conjunto com a estrutura proposta. O jogo integra a movimentação do paciente com os eventos ocorridos no jogo, enviando ações de controle para os motores através de uma API, e recebendo comandos baseados na telemetria dos sensores da estrutura.

Realizou-se também os primeiros testes experimentais no Módulo de Suporte Ativo de Peso, integrado ao jogo desenvolvido. Os testes foram realizados primeiramente em um boneco antropométrico manipulado externamente de forma manual, e posteriormente com voluntários saudáveis.

Os primeiros testes evidenciaram o funcionamento correto da estrutura em conjunto com o jogo proposto e a sua operação em conjunto com uma esteira simples. Os testes também demonstraram diversos aspectos construtivos a serem melhorados na estrutura para oferecer uma melhor experiência durante as sessões de treinamento.

Baseando-se nos testes realizados com os voluntários saudáveis, foi possível obter recomendações de melhorias para sistema de controle do personagem principal, montagem da estrutura física, montagem dos atuadores e ajustes no jogo sério.

Para o controle do personagem através da telemetria da estrutura, recomenda-se substituir a leitura da posição do atuador q_3 pela leitura de um sensor inercial a ser acoplado diretamente na plataforma móvel ou utilizar a posição do sistema de visão robótica já desenvolvido. Uma alternativa é aumentar o espaço de trabalho da estrutura ao longo do eixo OY e a largura das esteiras utilizadas. Dessa forma, o paciente poderá realizar os movimentos laterais com mais clareza.

Para o controle da estrutura, recomenda-se utilizar, em trabalhos futuros, *Machine Learning* para obter e avaliar o perfil de trajetória da marcha humana saudável na telemetria da estrutura, e controlar a *dead-zone* da função AAN em função de desvios observados neste perfil. Caso os sensores da estrutura sejam insuficientes, o uso de um sensor inercial na plataforma móvel poderá prover os dados necessários.

Durante os testes, também foi evidente que o material utilizado para confecção dos módulos de atuação não está adequado. Recomenda-se substituir os motores DC por modelos de 12 V, ou os drivers de corrente por outros capazes de operar em 24 V, ou ainda mudar o

acionamento do sistema para motores de passo. Os acoplamentos flexíveis também precisam ser substituídos por modelos capazes de transmitir torques maiores.

Quanto aos demais aspectos construtivos, recomenda-se substituir as articulações de rotação da estrutura por modelos comerciais para reduzir as folgas observadas na estrutura, o que poderá melhorar a transmissão de forças e o controle da plataforma móvel.

O sistema de alívio do peso passivo do corpo também pode ser melhorado. As molas aplicadas podem ser substituídas por um sistema de contrapesos e polias, o que permitirá um melhor controle da supressão de peso e do ajuste da posição da plataforma móvel para cada paciente.

Por fim, com base nas recomendações apresentadas e nos ajustes propostos, a estrutura poderá ser utilizada para testes clínicos a serem realizados em pacientes pós-AVE, seguindo os procedimentos de pesquisa já aprovados juntamente ao CEP, e obter um conjunto de dados em volume adequado para se avaliar impactos do uso da estrutura no treinamento da marcha humana, tanto quanto a execução dos movimentos, quanto ao equilíbrio e adaptação em situações adversas.

Além disso, para que o equipamento possa ser utilizado com segurança dentro de um ambiente hospitalar, será necessário adequá-lo às resoluções normativas da ANVISA e orientadas no Manual Para Regularização De Equipamentos Médicos (ANVISA, 2021).

Os testes a serem executados com o Módulo de Suporte Ativo de Peso Corporal poderão ainda ser expandidos para incluir um novo protótipo do Módulo de Movimentação por Esteira Bipartida e integrados por meio do Módulo de Controle e Monitoramento Sem Marcadores, promovendo a validação completa do sistema proposto.

Esta tese traz como principais contribuições o desenvolvimento de uma estrutura paralela para suporte ativo do peso corporal baseado em uma estrutura paralela com poucas configurações singulares, melhorias nos aspectos construtivos da estrutura MIT- Skywalker, a implementação de um jogo sério totalmente integrado com o a nova estrutura e os primeiros resultados de testes piloto em participantes saudáveis.

A realização deste trabalho possibilitou direta e indiretamente a produção de uma patente e dois registros de programa de computador, dois capítulos de livros técnicos, três apresentações em congressos locais e nacionais, duas publicações em revistas internacionais de alto impacto e a orientação de dois alunos de iniciação científica voluntária.

Publicações da tese

GONÇALVES, R. S.; RODRIGUES, L. A. O. Development of a Novel Parallel Structure for Gait Rehabilitation. In: **Handbook of Research on Advanced Mechatronic Systems and Intelligent Robotics**. Cap. 3, p. 42–81, 2019.

GONÇALVES, R. S.; RODRIGUES, L. A. O., GASPAR, L. A., Serious Games Integrated with Rehabilitation Structures and Assist-as-need Techniques. **Handbook of Research**

on Promoting Economic and Social Development Through Serious Games. Aceito para publicação. 2022.

RODRIGUES, L.A.O., GONÇALVES, R.S., Development of a Parallel Robotic Body Weight Support for Human Gait Rehabilitation, In: **Brazilian Congress in Biomedical Engineering (CBEB 2020)**, Vitoria, Brazil, 2020.

RODRIGUES, L.A.O., GONÇALVES, R.S., Desenvolvimento da Estrutura de Reabilitação da Marcha Humana HOPE-G, **28º Simpósio do Programa de Pós-graduação em Engenharia Mecânica POSMEC 2018**. Uberlândia, Brasil. 2018.

RODRIGUES, L.A.O., MORAES, V.P., GONÇALVES, R.S., Dispositivo de Reabilitação Combinada para Aplicação em Pacientes de Acidente Vascular Encefálico, **29º Simpósio do Programa de Pós-graduação em Engenharia Mecânica POSMEC 2019**. Uberlândia, Brasil. 2019.

RODRIGUES, L.A.O., GONÇALVES, R.S. **DISPOSITIVO PARALELO ATIVO PARA MOVIMENTAÇÃO DA PÉLVIS E SUPORTE DE PESO CORPORAL**, Depositante: Universidade Federal de Uberlândia. BR n° 102018070398-6 A2, depositado em 03/10/2018, concedido em 22/04/2020.

RODRIGUES, L.A.O., GONÇALVES, R.S., Development of nonmotorized mechanisms for lower limb rehabilitation, In: **Robotica**, v.40(1) p.102-119, 2021, DOI: 10.1017/S0263574721000412

RODRIGUES, L.A.O., MORAES, V.P., GONÇALVES, R.S., ReGear: an upper and lower limb simultaneous system for stroke rehabilitation, **Journal of the Brazilian Society of Mechanical Sciences and Engineering**, v.43, p.1-18, 2021, DOI: 10.1007/s40430-021-03199-8

REFERÊNCIAS

ALASHRAM, A. R.; ANNINO, G.; PADUA, E. **Robot-assisted gait training in individuals with spinal cord injury: A systematic review for the clinical effectiveness of Lokomat.** *Journal of Clinical Neuroscience*, v. 91, p. 260-269, set. 2021. <https://doi.org/10.1016/j.jocn.2021.07.019>

ALCHALCABI, A. E.; EDDIN, A. N.; SHIRMOHAMMADI, S. **More attention, less deficit: Wearable EEG-based serious game for focus improvement.** 2017 IEEE 5th International Conference on Serious Games and Applications for Health (SeGAH). Anais...IEEE, abr. 2017. <https://doi.org/10.1109/SeGAH.2017.7939288>

ALTUZARRA, O. et al. **A practical procedure to analyze singular configurations in closed kinematic chains.** *IEEE Transactions on Robotics*, v. 20, n. 6, p. 929-940, 2004. <https://doi.org/10.1109/TRO.2004.832798>

ALVES, T.; CHAVES D'CARVALHO, M.; GONÇALVES, R. S. **Assist-as-needed control in a cable-actuated robot for human joints rehabilitation.** *Journal of Mechanical Engineering and Biomechanics*, v. 3, n. 5, p. 57-62, 2019. <https://doi.org/10.24243/JMEB/3.5.214>

ANVISA, MANUAL PARA REGULARIZAÇÃO DE EQUIPAMENTOS MÉDICOS NA ANVISA, p.175, 2021. disponível em: <https://www.gov.br/anvisa/pt-br/centraisdeconteudo/publicacoes/produtos-para-a-saude/manuais/manual-para-regularizacao-de-equipamentos-medicos-na-anvisa.pdf>. Acessado em 15/03/2022.

ARDUINO. **Digital Pins With Interrupts.** Disponível em: <https://www.arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/>. Acessado em: 15/03/2022

ASL, H. J.; NARIKIYO, T.; KAWANISHI, M. **An assist-as-needed control scheme for robot-assisted rehabilitation.** *Proceedings of the American Control Conference*, p. 198-203, 2019. <https://doi.org/10.1109/IROS.2018.8594244>

AZIZI, SH. et al. **The impact of an anti-gravity treadmill (AlterG) training on walking capacity and corticospinal tract structure in children with cerebral palsy.** 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). 2017. DOI: <https://doi.org/10.1109/EMBC.2017.8037033>

BAGGI, R. P. A.; RODRIGUES, E. P.; CARAMÊZ, R. **Acidente Vascular Encefálico.** *Revista Unilus Ensino e Pesquisa*, v. 11, n. 25, p. 88-89, 2014. Disponível em: <http://revista.unilus.edu.br/index.php/ruep/article/view/209>. Acessado em 15/03/2022.

BARBOSA, A. M. **Desenvolvimento de uma estrutura robótica paralela atuada por cabos para reabilitação dos movimentos do ombro.** VI Congresso Nacional de Engenharia Mecânica. 2013. Disponível em: <https://www.abcm.org.br/anais/conem/2010/PDF/CON10-0607.pdf>. Acessado em 15/03/2022.

BARONCHELLI, F. et al. **The Effect of Robotic Assisted Gait Training With Lokomat® on Balance Control After Stroke: Systematic Review and Meta-Analysis.** *Frontiers in Neurology*, v. 12, 6 jul. 2021. <https://doi.org/10.3389/fneur.2021.661815>

BAUR, K. et al. **Robot-Supported Multiplayer Rehabilitation: Feasibility Study of Haptically Linked Patient-Spouse Training.** 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). *Anais...IEEE*, out. 2018a. <https://doi.org/10.1109/IROS.2018.8593769>

BAUR, K. et al. **Trends in robot-assisted and virtual reality-assisted neuromuscular therapy: a systematic review of health-related multiplayer games.** *Journal of NeuroEngineering and Rehabilitation*, v. 15, n. 1, p. 107, dez. 2018b. <https://doi.org/10.1186/s12984-018-0449-9>

BLAYA, J. A.; HERR, H. **Adaptive Control of a Variable-Impedance Ankle-Foot Orthosis to Assist Drop-Foot Gait.** *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, v. 12, n. 1, p. 24-31, 2004. <https://doi.org/10.1109/TNSRE.2003.823266>

BONEV, I. **General Terminology Related to Parallel Mechanisms.** Disponível em: <https://www.parallelemic.org/Terminology/General.html>. Acessado em: 15/03/2022

CAMARGOS, A. C. R. et al. **Relação entre independência funcional e qualidade de vida na paralisia cerebral.** *Relationship between functional independence and quality of life in cerebral palsy.* v. 25, n. 1, p. 83-92, 2012. <https://doi.org/10.1590/S0103-51502012000100009>

CLAFLIN, E. S.; KRISHNAN, C.; KHOT, S. P. **Emerging Treatments for Motor Rehabilitation After Stroke.** *The Neurohospitalist*, v. 5, n. 2, p. 77-88, abr. 2015. <https://doi.org/10.1177/1941874414561023>

DIAZ, I.; GIL, J. J.; SANCHEZ, E. **Lower-Limb Robotic Rehabilitation: Literature Review and Challenges.** *Journal of Robotics*, v. 2011, n. 11, p. e759764, 2011. <https://doi.org/10.1155/2011/759764>

DING, H. et al. **Computer-aided structural synthesis of 5-DOF parallel mechanisms and the establishment of kinematic structure databases.** *Mechanism and Machine Theory*, v. 83, p. 14-30, 2015. <https://doi.org/10.1016/j.mechmachtheory.2014.08.013>

DOMÍNGUEZ-TÉLLEZ, P. et al. **Game-Based Virtual Reality Interventions to Improve Upper Limb Motor Function and Quality of Life After Stroke: Systematic Review and Meta-analysis.** *Games for Health Journal*, v. 9, n. 1, p. 1-10, 1 fev. 2020. <https://doi.org/10.1089/g4h.2019.0043>

DYER, J.-O. et al. **Changes in activation timing of knee and ankle extensors during gait are related to changes in heteronymous spinal pathways after stroke.** *Journal of NeuroEngineering and Rehabilitation*, v. 11, n. 1, p. 148, 2014. <https://doi.org/10.1186/1743-0003-11-148>

FADALI, M.; VISIOLI, A. **Digital control engineering: analysis and design**. Academic Press. ISBN 978-0-12-374498-2. 536p. 2009. <https://doi.org/10.1016/B978-0-12-374498-2.00016-3>

FONG, J.; TAVAKOLI, M. **Kinesthetic teaching of a therapist's behavior to a rehabilitation robot**. 2018 International Symposium on Medical Robotics (ISMR). Anais...IEEE, mar. 2018. <https://doi.org/10.1109/ISMR.2018.8333285>

GONÇALVES, R. S. et al. **MIT-Skywalker: Evaluating Comfort of Bicycle / Saddle Seat**. 2017 International Conference on Rehabilitation Robotics (ICORR). Anais...London: 2017. <https://doi.org/10.1109/ICORR.2017.8009300>

GONÇALVES, R. S.; CARVALHO, J. C. M.; LOBATO, F. S. **Design of a robotic device actuated by cables for human lower limb rehabilitation using self-adaptive differential evolution and robust optimization**. Bioscience Journal, p. 1689-1702, 2016a. <https://doi.org/10.14393/BJ-v32n1a2016-32436>

GONÇALVES, R. S.; CARVALHO, J. C. M.; LOBATO, F. S. **Dispositivo robótico atuado por cabos para reabilitação do membro inferior humano utilizando evolução diferencial auto-adaptável e otimização robusta**. Bioscience Journal, v. 32, n. 6, p. 1689-1702, 2016b. <https://doi.org/10.14393/BJ-v32n1a2016-32436>

GONÇALVES, R. S.; HAMILTON, T.; KREBS, H. I. **MIT-Skywalker: On the use of a markerless system**. IEEE International Conference on Rehabilitation Robotics. Anais...2017. <https://doi.org/10.1109/ICORR.2017.8009247>

GOSELIN, C. **Compact dynamic models for the tripteron and quadruperon parallel manipulators**. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, v. 223, n. 1, p. 1-12, 1 fev. 2009. <https://doi.org/10.1243/09596518JSCE605>

GOSELIN, C.; LALIBERTÉ, T.; VEILLETTE, A. **Singularity-Free Kinematically Redundant Planar Parallel Mechanisms With Unlimited Rotational Capability**. IEEE Transactions on Robotics, v. 31, n. 2, p. 457-467, 2015. <https://doi.org/10.1109/TRO.2015.2409433>

GOSELIN, C. M. et al. **Parallel Mechanisms of the Multipteron Family: Kinematic Architectures and Benchmarking**. Proceedings 2007 IEEE International Conference on Robotics and Automation. Anais...IEEE, abr. 2007. <https://doi.org/10.1109/ROBOT.2007.363045>

GUTIÉRREZ, Á. et al. **Serious Game Platform with Haptic Feedback and EMG Monitoring for Upper Limb Rehabilitation and Smoothness Quantification on Spinal Cord Injury Patients**. Applied Sciences, v. 10, n. 3, p. 963, 2 fev. 2020. <https://doi.org/10.3390/app10030963>

HERNÁNDEZ, A. et al. **Kinematic analysis of mechanisms via a velocity equation based in a geometric matrix.** *Mechanism and Machine Theory*, v. 38, n. 12, p. 1413-1429, 2003. [https://doi.org/10.1016/S0094-114X\(03\)00095-8](https://doi.org/10.1016/S0094-114X(03)00095-8)

HESSE, S. et al. **Upper and lower extremity robotic devices for rehabilitation and for studying motor control** *Current Opinion in Neurology*. Lippincott Williams & Wilkins, 2003. <https://doi.org/10.1097/00019052-200312000-00010>

HOBBS, B.; ARTEMIADIS, P. **A Review of Robot-Assisted Lower-Limb Stroke Therapy: Unexplored Paths and Future Directions in Gait Rehabilitation.** *Frontiers in Neurorobotics*, v. 14, 15 abr. 2020. <https://doi.org/10.3389/fnbot.2020.00019>

HOGAN, N. **Impedance Control: An Approach to Manipulation.** *American Control Conference*. <https://doi.org/10.23919/ACC.1984.4788393>

HONG, W.; ANIL KUMAR, N.; HUR, P. **A Phase-Shifting Based Human Gait Phase Estimation for Powered Transfemoral Prostheses.** *IEEE Robotics and Automation Letters*, v. 6, n. 3, p. 5113-5120, jul. 2021. <https://doi.org/10.1109/LRA.2021.3068907>

HUANG, Y. et al. **A comparison of the rehabilitation effectiveness of neuromuscular electrical stimulation robotic hand training and pure robotic hand training after stroke: A randomized controlled trial.** *Biomedical Signal Processing and Control*, v. 56, p. 101723, fev. 2020. <https://doi.org/10.1016/j.bspc.2019.101723>

HUSSAIN, S.; XIE, S. Q.; JAMWAL, P. K. **Adaptive impedance control of a robotic orthosis for gait rehabilitation.** *IEEE Transactions on Cybernetics*, v. 43, n. 3, p. 1025-1034, 2013. <https://doi.org/10.1109/TSMCB.2012.2222374>

I. N., MEDVEDEV. **Place and Possibilities of the Robotic System Lokomat in the Rehabilitation of Patients After Ischemic Stroke.** *Biomedical and Pharmacology Journal*, v. 12, n. 1, p. 131-140, 28 mar. 2019. <https://doi.org/10.13005/bpj/1621>

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **Antropometria e Estado Nutricional no Brasil 2008-2009.** <https://doi.org/10.13005/bpj/1621IBGE>

IJSSELSTEIJN, W. et al. **Measuring the Experience of Digital Game Enjoyment.** *Proceedings of Measuring Behavior 2008*.

JAMWAL, P. K. et al. Adaptive Impedance Control of Parallel Ankle Rehabilitation Robot. **Journal of Dynamic Systems, Measurement, and Control**, v. 139, n. 11, 1 nov. 2017. <https://doi.org/10.1115/1.4036560>

JEZERNIK, S.; COLOMBO, G.; MORARI, M. Automatic gait-pattern adaptation algorithms for rehabilitation with a 4-DOF robotic orthosis. **IEEE Transactions on Robotics and Automation**, v. 20, n. 3, p. 574–582, 2004. <https://doi.org/10.1109/TRA.2004.825515>

JIN, X. et al. A class of novel 4-DOF and 5-DOF generalized parallel mechanisms with high performance. **Mechanism and Machine Theory**, v. 120, p. 57–72, 2018. <https://doi.org/10.1016/j.mechmachtheory.2017.09.015>

KAISER, M. S. et al. Healthcare Robots to Combat COVID-19. In: SANTOSH, K. C.; JOSHI, A. (Eds.). . **COVID-19: Prediction, Decision-Making, and its Impacts**. Singapore: Springer Singapore, 2021. p. 83–97. https://doi.org/10.1007/978-981-15-9682-7_10

KATAOKA, N. et al. **Effects of partial body-weight support and functional electrical stimulation on gait characteristics during treadmill locomotion: Pros and cons of saddle-seat-type body-weight support**. 2017 International Conference on Rehabilitation Robotics (ICORR). **Anais...IEEE**, jul. 2017. <https://doi.org/10.1109/ICORR.2017.8009277>

KHARB, A. et al. **A REVIEW OF GAIT CYCLE AND ITS PARAMETERS**. IJCEM International Journal of Computational Engineering & Management. 2011

KONG, X.; GOSSELIN, C. M. Type synthesis of 5-DOF parallel manipulators based on screw theory. **Journal of Robotic Systems**, v. 22, n. 10, p. 535–547, 2005. <https://doi.org/10.1002/rob.20084>

LAURETTI, C. et al. Learning by Demonstration for Planning Activities of Daily Living in Rehabilitation and Assistive Robotics. **IEEE Robotics and Automation Letters**, v. 2, n. 3, p. 1375–1382, jul. 2017. <https://doi.org/10.1109/LRA.2017.2669369>

LAVADOS, P. M. et al. Stroke epidemiology, prevention, and management strategies at a regional level: Latin America and the Caribbean. **The Lancet Neurology**, v. 6, n. 4, p. 362–372, 2007. [https://doi.org/10.1016/S1474-4422\(07\)70003-0](https://doi.org/10.1016/S1474-4422(07)70003-0)

LEFEBER, N.; SWINNEN, E.; KERCKHOFS, E. **The immediate effects of robot-assistance on energy consumption and cardiorespiratory load during walking compared to walking without robot-assistance: a systematic review**. Disability and Rehabilitation: Assistive Technology, 2017. https://doi.org/10.1007/978-3-319-46669-9_200

LI, J.; CHEN, D.; FAN, Y. An Open-Structure Treadmill Gait Trainer: From Research to Application. **Journal of Healthcare Engineering**, v. 2017, p. 1–12, 2017. <https://doi.org/10.1155/2017/9053630>

LIANG, B. W. et al. Pelvic step: The contribution of horizontal pelvis rotation to step length in young healthy adults walking on a treadmill. **Gait and Posture**, v. 39, n. 1, p. 105–110, 2014. <https://doi.org/10.1016/j.gaitpost.2013.06.006>

LOBO DA COSTA, P. H.; SERRÃO, F. VI. **Movimento articular: Aspectos Morfológicos E Funcionais (Membro Inferior)**. 1. ed. São Paulo: Malone, 2010. v. 2

LOBO, P. G. G. A. et al. Epidemiologia do acidente vascular cerebral isquêmico no Brasil no ano de 2019, uma análise sob a perspectiva da faixa etária / Epidemiology of the ischemic cerebrovascular accident in Brazil in the year of 2019, an analysis from an age group perspective.

Brazilian Journal of Health Review, v. 4, n. 1, p. 3498–3505, 2021. <https://doi.org/10.34119/bjhrv4n1-272>

LU, Y. et al. Kinematics and statics analysis of a novel 5-DoF parallel manipulator with two composite rotational/linear active legs. **Robotics and Computer-Integrated Manufacturing**, v. 30, n. 1, p. 25–33, 2014. <https://doi.org/10.1016/j.rcim.2013.07.003>

LUM, P. et al. Robotic Devices for Movement Therapy After Stroke: Current Status and Challenges to Clinical Acceptance. **Topics in Stroke Rehabilitation**, v. 8, n. 4, p. 40–53, 2002. <https://doi.org/10.1310/9KFM-KF81-P9A4-5WW0>

LUNA, L. et al. **EMG-Based Kinematic Impedance Control of a Lower-Limb Exoskeleton**. VIII Latin American Conference on Biomedical Engineering and XLII National Conference on Biomedical Engineering. CLAIB 2019. IFMBE Proceedings, vol 75. Springer, Cham. https://doi.org/10.1007/978-3-030-30648-9_192

MACLENNAN, A. H.; THOMPSON, S. C.; GECZ, J. Cerebral palsy: causes, pathways, and the role of genetic variants. **American Journal of Obstetrics and Gynecology**, v. 213, n. 6, p. 779–788, dez. 2015. <https://doi.org/10.1016/j.ajog.2015.05.034>

MAGGIO, M. G. et al. Effects of robotic neurorehabilitation through lokomat plus virtual reality on cognitive function in patients with traumatic brain injury: A retrospective case-control study. **International Journal of Neuroscience**, v. 130, n. 2, p. 117–123, 1 fev. 2020. <https://doi.org/10.1080/00207454.2019.1664519>

MICHAEL-ASALU, A. et al. Cerebral Palsy. **Advances in Pediatrics**, v. 66, p. 189–208, ago. 2019. <https://doi.org/10.1016/j.yapd.2019.04.002>

MINELLI, C.; FEN, L. F.; MINELLI, D. P. C. Stroke incidence, prognosis, 30-day, and 1-year case fatality rates in Matão, Brazil: A population-based prospective study. **Stroke**, v. 38, n. 11, p. 2906–2911, 2007. <https://doi.org/10.1161/STROKEAHA.107.484139>

MOTEVALLI, B.; ZOHOOR, H.; SOHRABPOUR, S. Structural synthesis of 5 DoFs 3T2R parallel manipulators with prismatic actuators on the base. **Robotics and Autonomous Systems**, v. 58, n. 3, p. 307–321, 2010. <https://doi.org/10.1016/j.robot.2009.10.001>

MUBIN, O. et al. Exoskeletons With Virtual Reality, Augmented Reality, and Gamification for Stroke Patients' Rehabilitation: Systematic Review. **JMIR Rehabilitation and Assistive Technologies**, v. 6, n. 2, p. e12010, set. 2019. <https://doi.org/10.2196/12010>

MUBIN, O. et al. Exploring serious games for stroke rehabilitation: a scoping review. **Disability and Rehabilitation: Assistive Technology**, p. 1–7, jun. 2020. <https://doi.org/10.1080/17483107.2020.1768309>

NACKE, L. E. et al. Playability and Player Experience Research. **Proceedings of the IEEE**, p. 1–11, 2009. <http://doi.org/10.3390/s20143958>

NUNES, W. M. et al. **Cable-based parallel manipulator for rehabilitation of shoulder and elbow movements**. IEEE International Conference on Rehabilitation Robotics. *Anais...*2011. <https://doi.org/10.1109/ICORR.2011.5975503>

NUNES, W. M. **Desenvolvimento de uma estrutura robótica atuada por cabos para reabilitação/recuperação dos movimentos do ombro humano**. Universidade Federal De Uberlândia, 2012. <https://doi.org/10.14393/ufu.di.2012.16>

NUNES, W. M. et al. **SISTEMA DE CONTROLE DO CaMaReS**. Conferência Brasileira de Dinâmica, Controle e Aplicações. 2012. <http://doi.org/10.5540/DINCON.2011.001.1.0145>

OCHOA, J.; STERNAD, D.; HOGAN, N. Treadmill vs. overground walking: different response to physical interaction. **Journal of Neurophysiology**, v. 118, n. 4, p. 2089–2102, 1 out. 2017. <https://doi.org/10.1152/jn.00176.2017>

O’SULLIVAN, S. B.; SCHIMITZ, T. J. **Fisioterapia: avaliação e tratamento**. 5. ed. MANOLE, 2010.

PAOLUCCI, S. et al. Post-stroke Depression Increases Disability More Than 15% in Ischemic Stroke Survivors: A Case-Control Study. **Frontiers in Neurology**, v. 10, 27 ago. 2019. <https://doi.org/10.3389/fneur.2019.00926>

PEREZ-IBARRA, J. C. et al. Adaptive Impedance Control Applied to Robot-Aided Neuro-Rehabilitation of the Ankle. **IEEE Robotics and Automation Letters**, v. 4, n. 2, p. 185–192, abr. 2019. <https://doi.org/10.1109/LRA.2018.2885165>

PINO, A.; MÚNERA, M.; CIFUENTES, C. A. Serious Games in Robot-Assisted Rehabilitation Therapy for Neurological Patients. In: **Interfacing Humans and Robots for Gait Assistance and Rehabilitation**. Cham: Springer International Publishing, 2022. p. 309–329. https://doi.org/10.1007/978-3-030-79630-3_12

POSTOLACHE, G. et al. Serious Games Based on Kinect and Leap Motion Controller for Upper Limbs Physical Rehabilitation. p. 147–177. https://doi.org/10.1007/978-3-319-99540-3_8

QIU, S. et al. Exoskeleton Online Learning and Estimation of Human Walking Intention Based on Dynamical Movement Primitives. **IEEE Transactions on Cognitive and Developmental Systems**, v. 13, n. 1, p. 67–79, mar. 2021. <https://doi.org/10.1109/TCDS.2020.2968845>

R, G. S.; BRODELL, J. Design of the Crank-Rocker Mechanism with Unit Time Ratio. **Journal of Mechanisms**, v. 5, n. 1, p. 1–4, 1970. [https://doi.org/10.1016/0022-2569\(70\)90047-9](https://doi.org/10.1016/0022-2569(70)90047-9)

REGO, P.; MOREIRA, P. M.; REIS, L. P. Serious Games for Rehabilitation: A survey and a classification towards a taxonomy. **Proceedings of the 5th Iberian Conference on Information Systems and Technologies, CISTI 2010**, n. July, p. 0–6, 2010.

REINKENSMEYER, D.; WYNNE, J. H.; HARKEMA, S. J. **A robotic tool for studying locomotor adaptation and rehabilitation**. Proceedings of the Second Joint 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society. Houston, TX, USA, 2002, pp. 2353-2354 vol.3, doi: 10.1109/IEMBS.2002.1053318.

ROBERTS, M. H. **A robot for gait rehabilitation**. Tese de Doutorado. Massachusetts Institute of Technology. 2004. <http://hdl.handle.net/1721.1/34562>

RODRIGUES, L. A. O. **Desenvolvimento de dispositivos não motorizados para reabilitação do membro inferior humano**. Dissertação de Mestrado. Universidade Federal de Uberlândia, 2017. <http://doi.org/10.14393/ufu.di.2017.270>

RODRIGUES, L. A. O.; GONÇALVES, R. S. Desenvolvimento De Jogos Aplicados a Reabilitação Do Membro Superior Humano. **XXIV Congresso Brasileiro de Engenharia Biomédica – CBEB**, p. 396–399, 2014.

RODRIGUES, L. A. O.; MORAES, V. P. DE; GONÇALVES, R. S. **Dispositivo de Reabilitação Combinada para Aplicação em Pacientes de Acidente Vascular Encefálico**. POSMEC 2019, 2019, Uberlândia. Anais do POSMEC 2019. p. 2017–2019, 2019.

ROSAMOND, W. et al. Heart disease and stroke statistics-2008 Update: A report from the American heart association statistics committee and stroke statistics subcommittee. **Circulation**, v. 117, n. 4, 2008. <https://doi.org/10.1161/CIRCULATIONAHA.107.187998>

SACCO, I. DE C. N.; TANAKA, C. • SACCO, I.C.N.; TANAKA C. **Cinesiologia e Biomecânica dos Complexos Articulares**. Guanabara Koogan, RJ, 2008. Rio de Janeiro: Guanabara Koogan, 2008.

SALIM, V. V. **DESENVOLVIMENTO DE UM SISTEMA SEM MARCADORES PARA CONTROLE DE UM DISPOSITIVO ROBÓTICO PARA REABILITAÇÃO DA MARCHA HUMANA**. Dissertação de Mestrado. Universidade Federal de Uberlândia, 2018. <http://dx.doi.org/10.14393/ufu.di.2018.1183>

SCIAVICCO, L.; SICILIANO, B. **Modelling and Control of Robot Manipulators**. 1. ed. Verlag: Springer. v. 1, 2005

SHI, D. et al. Assist-as-needed attitude control in three-dimensional space for robotic rehabilitation. **Mechanism and Machine Theory**, v. 154, p. 104044, dez. 2020. <https://doi.org/10.1016/j.mechmachtheory.2020.104044>

SRIVASTAVA, S. et al. Assist-as-Needed Robot-Aided Gait Training Improves Walking Function in Individuals Following Stroke. **IEEE Transactions on Neural Systems and Rehabilitation Engineering**, v. 23, n. 6, p. 956–963, nov. 2015. <https://doi.org/10.1109/TNSRE.2014.2360822>

SUSKO, T. G. **MIT Skywalker: A novel robot for gait rehabilitation of stroke and cerebral palsy patients**. Tese de doutorado. Massachusetts Institute of Technology. 2015 <http://hdl.handle.net/1721.1/97844>

SUSKO, T.; SWAMINATHAN, K.; KREBS, H. I. MIT-Skywalker: A Novel Gait Neurorehabilitation Robot for Stroke and Cerebral Palsy. **IEEE Transactions on Neural Systems and Rehabilitation Engineering**, v. 24, n. 10, p. 1089–1099, 2016. <https://doi.org/10.1109/TNSRE.2016.2533492>

SWINNEN, E. et al. Robot-assisted walking with the Lokomat: The influence of different levels of guidance force on thorax and pelvis kinematics. **Clinical Biomechanics**, v. 30, n. 3, p. 254–259, 2015a. <https://doi.org/10.1016/j.clinbiomech.2015.01.006>

SWINNEN, E. et al. Body weight support during robot-assisted walking: Influence on the trunk and pelvis kinematics. **NeuroRehabilitation**, 2015b. <https://doi.org/10.3233/NRE-141195>

SWINNEN, E. et al. Walking with robot assistance: the influence of body weight support on the trunk and pelvis kinematics. **Disability and Rehabilitation: Assistive Technology**, v. 10, n. 3, p. 252–257, 2015c. <https://doi.org/10.3109/17483107.2014.888487>

TRIOLA, M. **Introdução à Estatística (tradução)**. 7. ed. Rio de Janeiro: LTC, 1999. <https://doi.org/10.11606/issn.2359-5388.i3p7>

UICKER, J. J. et al. **Theory of Machines and Mechanisms** *Journal of Mechanical Design* Oxford Press, 2003. <https://doi.org/10.1115/1.1605769>

VENEMAN, J. F. et al. Design and evaluation of the LOPES exoskeleton robot for interactive gait rehabilitation. **IEEE Transactions on Neural Systems and Rehabilitation Engineering**, v. 15, n. 1, p. 379–386, 2007. <https://doi.org/10.1109/TNSRE.2007.903919>

VENEMAN, J. F. et al. Fixating the pelvis in the horizontal plane affects gait characteristics. **Gait and Posture**, v. 28, n. 1, p. 157–163, 2008. <https://doi.org/10.1016/j.gaitpost.2007.11.008>

VIRANI, S. S. et al. Heart Disease and Stroke Statistics—2021 Update. **Circulation**, v. 143, n. 8, 23 fev. 2021. <https://doi.org/10.1161/CIR.0000000000000950>

XIE, F. et al. Mobility, Singularity, and Kinematics Analyses of a Novel Spatial Parallel Mechanism. **Journal of Mechanisms and Robotics**, v. 8, n. 6, p. 061022, 2016. <https://doi.org/10.1115/1.4034886>

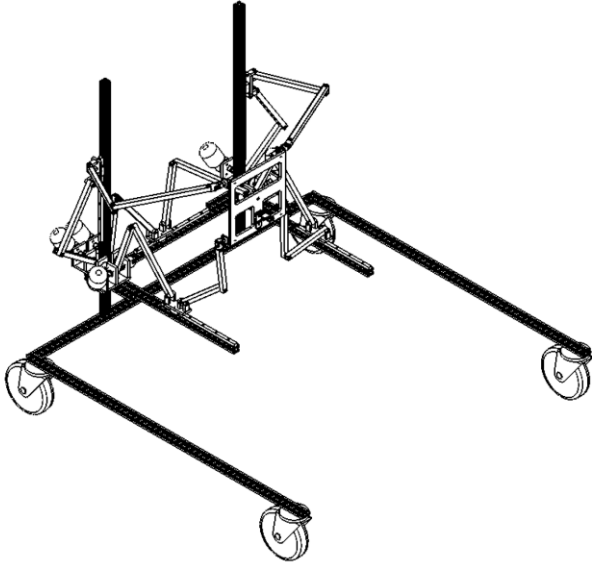
ZHU, S. J.; HUANG, Z.; ZHAO, M. Y. Singularity analysis for six practicable 5-DoF fully-symmetrical parallel manipulators. **Mechanism and Machine Theory**, v. 44, n. 4, p. 710–725, 2009. <https://doi.org/10.1016/j.mechmachtheory.2008.05.012>

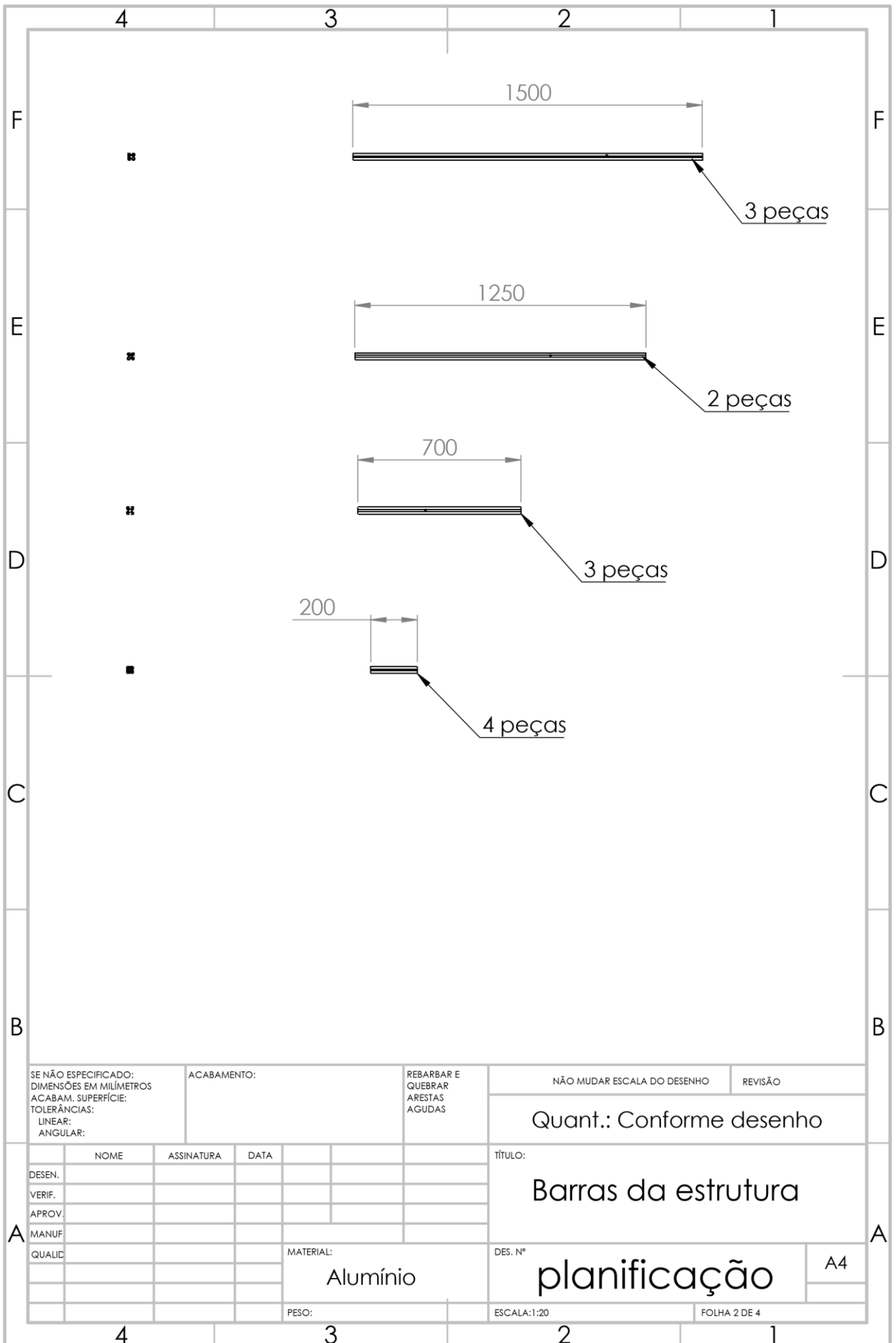
ZLATANOV, D. S. **GENERALIZED SINGULARITY ANALYSIS OF MECHANISMS**. Tese de doutorado. University of Toronto. 232p. 1998

APÊNDICES

Planificações do Módulo de Suporte Ativo de Peso Corporal

OBS: As quantidades podem divergir da descrição do texto da tese, pois a planificação referiu-se apenas às peças não disponíveis no laboratório no momento da construção do protótipo.

	4	3	2	1	
F					F
E					E
D					D
C					C
B					B
A					A
SE NÃO ESPECIFICADO: DIMENSÕES EM MILÍMETROS ACABAM. SUPERFÍCIE: TOLERÂNCIAS: LINEAR: ANGULAR:		ACABAMENTO:	REBARBAR E QUEBRAR ARESTAS AGUDAS	NÃO MUDAR ESCALA DO DESENHO	REVISÃO
DESEN.	NOME	ASSINATURA	DATA	TÍTULO: Vista Geral	
VERIF.					
APROV.					
MANUF.					
QUALIE				MATERIAL:	DES. Nº planificação
				PESO:	ESCALA:1:50
					FOLHA 1 DE 4
	4	3	2	1	



SE NÃO ESPECIFICADO:
DIMENSÕES EM MILÍMETROS
ACABAM. SUPERFÍCIE:
TOLERÂNCIAS:
LINEAR:
ANGULAR:

ACABAMENTO:

REBARBAR E
QUEBRAR
ARESTAS
AGUDAS

NÃO MUDAR ESCALA DO DESENHO

REVISÃO

Quant.: Conforme desenho

	NOME	ASSINATURA	DATA			
DESEN.						
VERIF.						
APROV.						
MANUF.						
QUALID.						

TÍTULO:

Barras da estrutura

DES. N°

planificação

A4

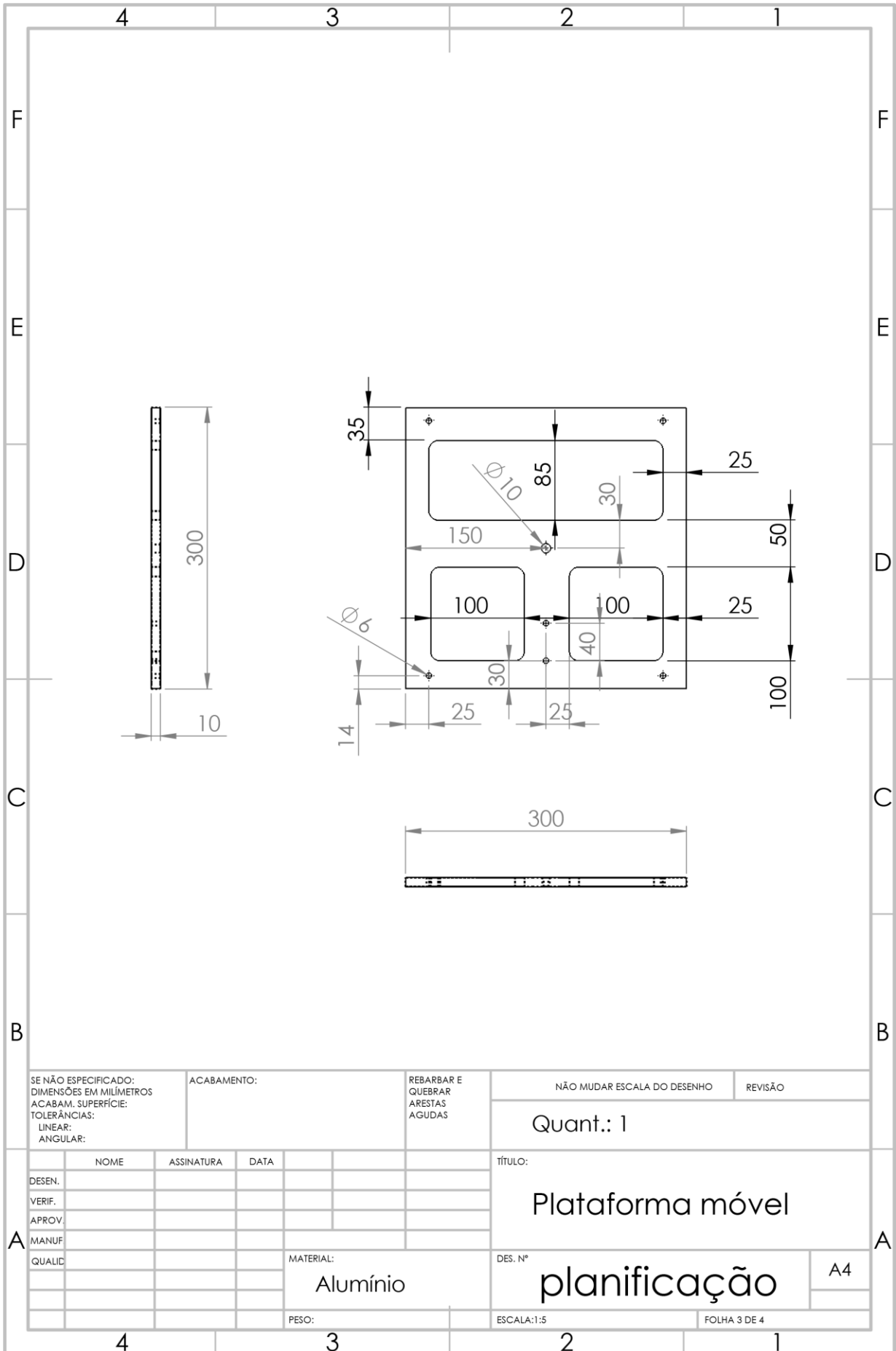
MATERIAL:

Alumínio

PESO:

ESCALA:1:20

FOLHA 2 DE 4



SE NÃO ESPECIFICADO:
DIMENSÕES EM MILÍMETROS
ACABAM. SUPERFÍCIE:
TOLERÂNCIAS:
LINEAR:
ANGULAR:

ACABAMENTO:

REBARBAR E
QUEBRAR
ARESTAS
AGUDAS

NÃO MUDAR ESCALA DO DESENHO

REVISÃO

Quant.: 1

	NOME	ASSINATURA	DATA
DESEN.			
VERIF.			
APROV.			
MANUF.			
QUALID.			

TÍTULO:

Plataforma móvel

DES. Nº

planificação

A4

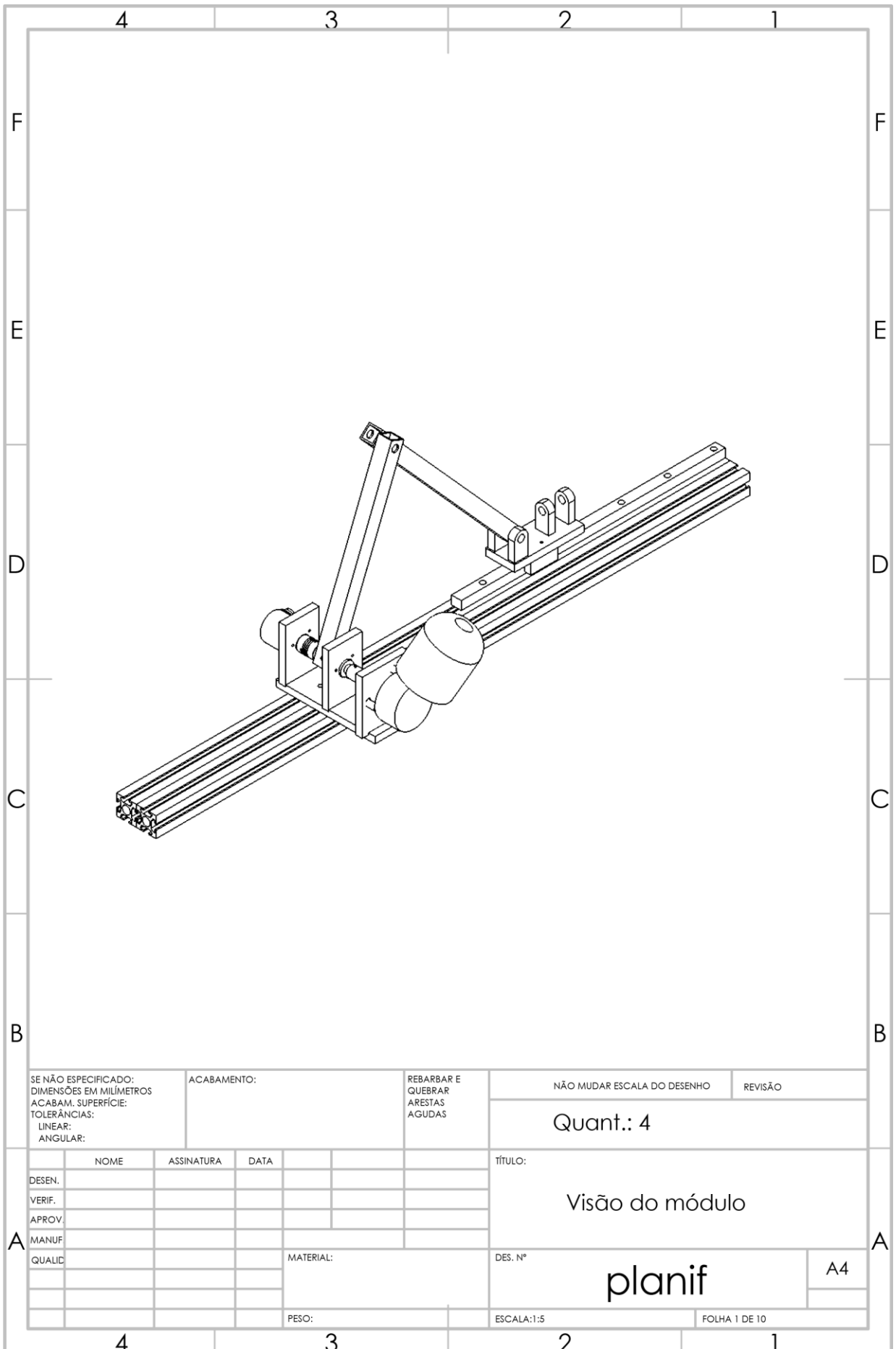
MATERIAL:

Alumínio

PESO:

ESCALA:1:5

FOLHA 3 DE 4



SE NÃO ESPECIFICADO:
DIMENSÕES EM MILÍMETROS
ACABAM. SUPERFÍCIE:
TOLERÂNCIAS:
LINEAR:
ANGULAR:

ACABAMENTO:

REBARBAR E
QUEBRAR
ARESTAS
AGUDAS

NÃO MUDAR ESCALA DO DESENHO

REVISÃO

Quant.: 4

	NOME	ASSINATURA	DATA		
DESEN.					
VERIF.					
APROV.					
MANUF					
QUALID					

TÍTULO:

Visão do módulo

MATERIAL:

DES. Nº

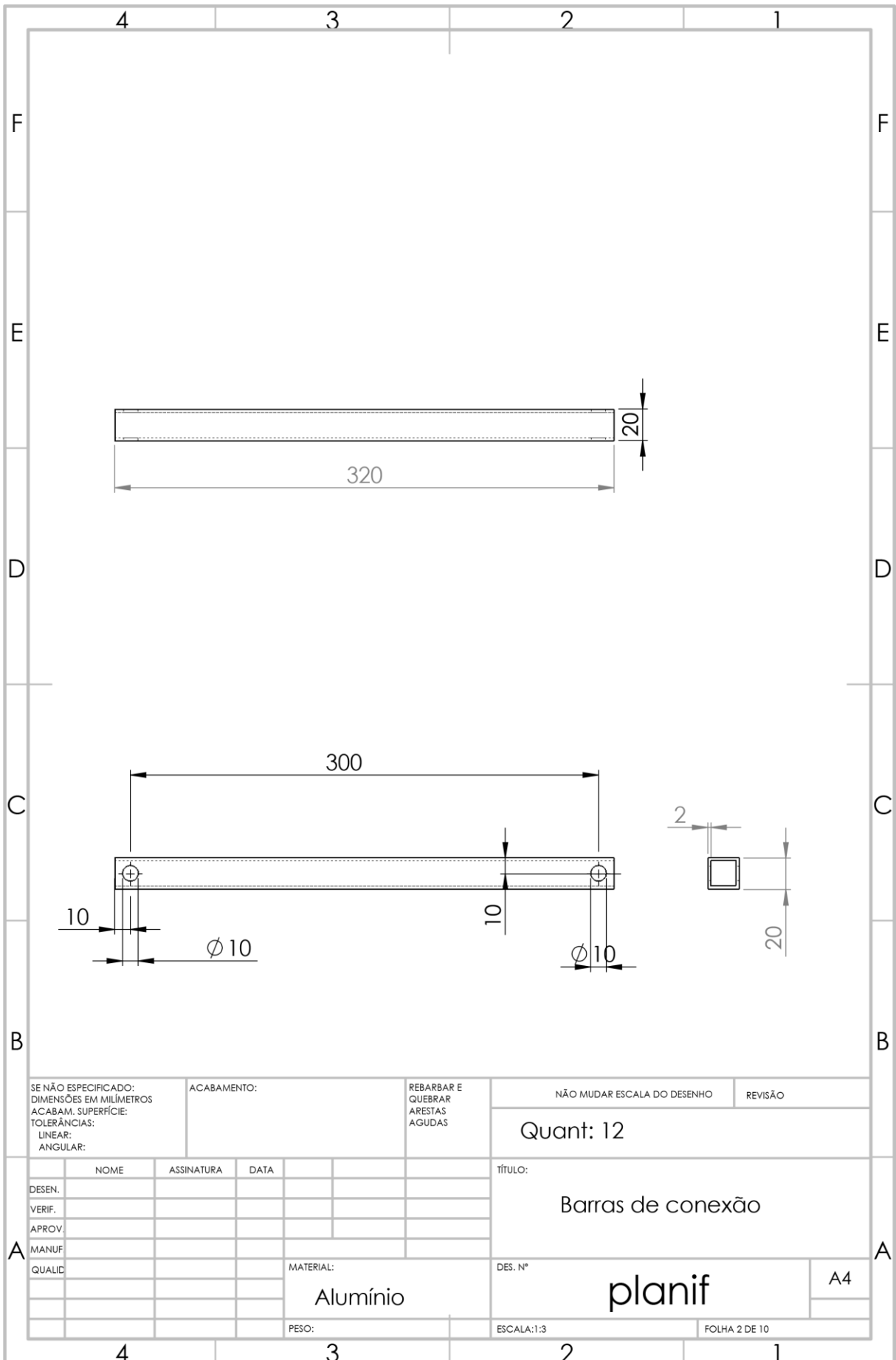
planif

A4

PESO:

ESCALA:1:5

FOLHA 1 DE 10



SE NÃO ESPECIFICADO:
DIMENSÕES EM MILÍMETROS
ACABAM. SUPERFÍCIE:
TOLERÂNCIAS:
LINEAR:
ANGULAR:

ACABAMENTO:

REBARBAR E
QUEBRAR
ARESTAS
AGUDAS

NÃO MUDAR ESCALA DO DESENHO

REVISÃO

Quant: 12

	NOME	ASSINATURA	DATA		
DESEN.					
VERIF.					
APROV.					
MANUF.					
QUALIC.					

TÍTULO:

Barras de conexão

MATERIAL:

Alumínio

DES. Nº

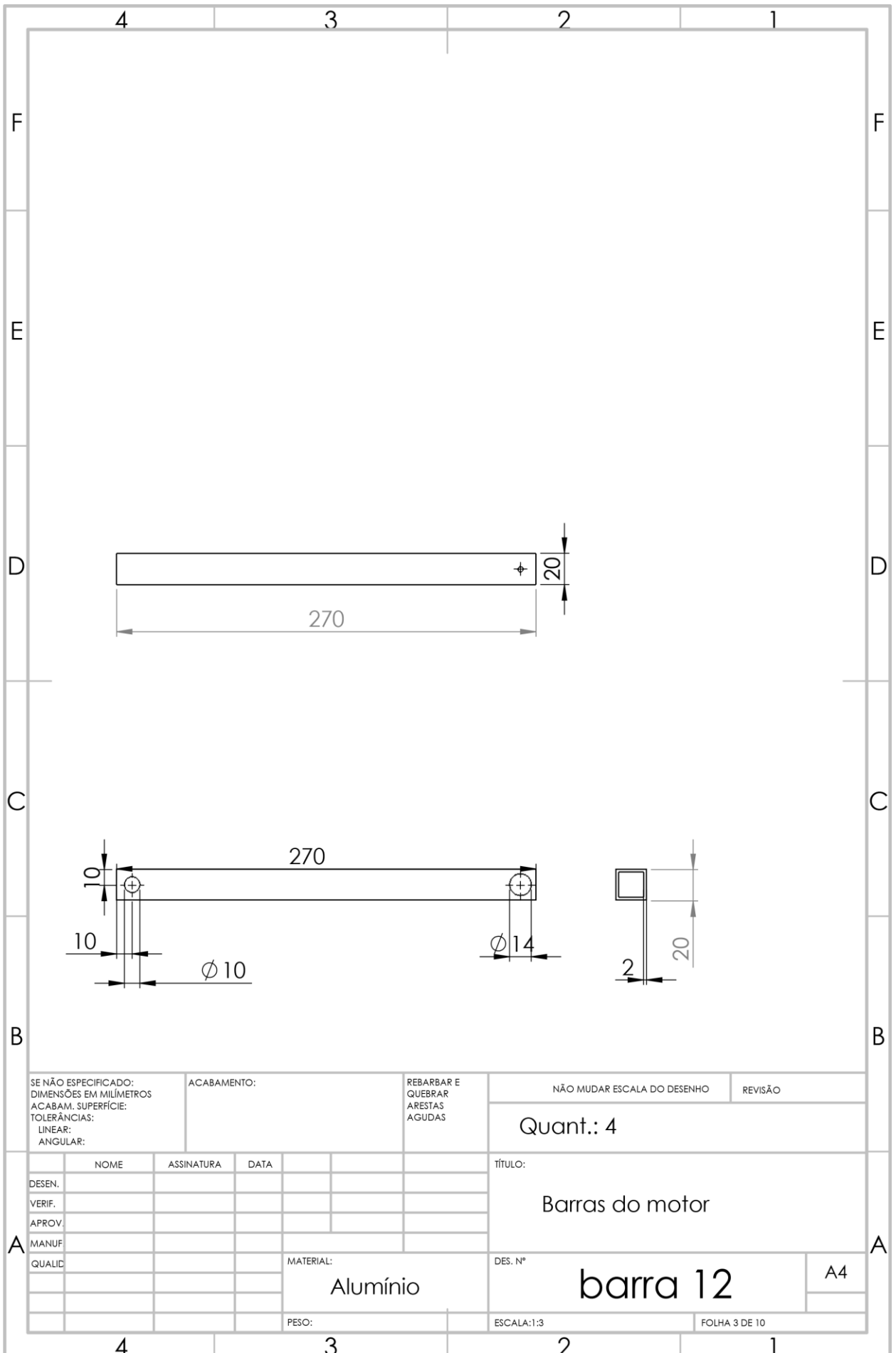
planif

A4

PESO:

ESCALA:1:3

FOLHA 2 DE 10



SE NÃO ESPECIFICADO: DIMENSÕES EM MILÍMETROS ACABAM. SUPERFÍCIE: TOLERÂNCIAS: LINEAR: ANGULAR:		ACABAMENTO:		REBARBAR E QUEBRAR ARESTAS AGUDAS		NÃO MUDAR ESCALA DO DESENHO		REVISÃO	
						Quant.: 4			
						TÍTULO: Barras do motor			
						DES. Nº		barra 12	
						MATERIAL: Alumínio		A4	
						PESO:		ESCALA:1:3	
								FOLHA 3 DE 10	

A

A

F

F

E

E

D

D

C

C

B

B

4

3

2

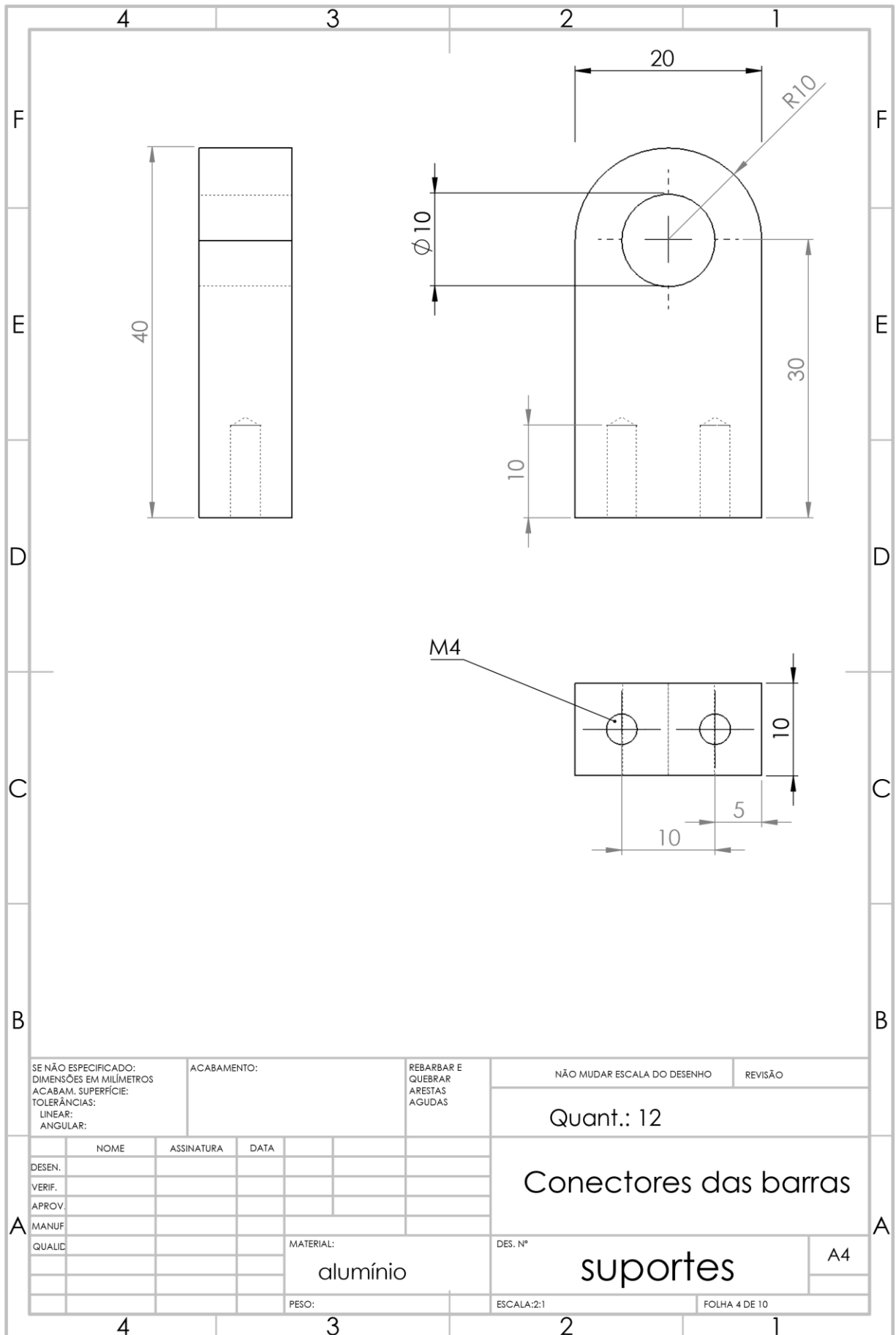
1

4

3

2

1



SE NÃO ESPECIFICADO:
DIMENSÕES EM MILÍMETROS
ACABAM. SUPERFÍCIE:
TOLERÂNCIAS:
LINEAR:
ANGULAR:

ACABAMENTO:

REBARBAR E
QUEBRAR
ARESTAS
AGUDAS

NÃO MUDAR ESCALA DO DESENHO

REVISÃO

Quant.: 12

Conectores das barras

	NOME	ASSINATURA	DATA
DESEN.			
VERIF.			
APROV.			
MANUF.			
QUALID.			

MATERIAL:
alumínio

DES. Nº

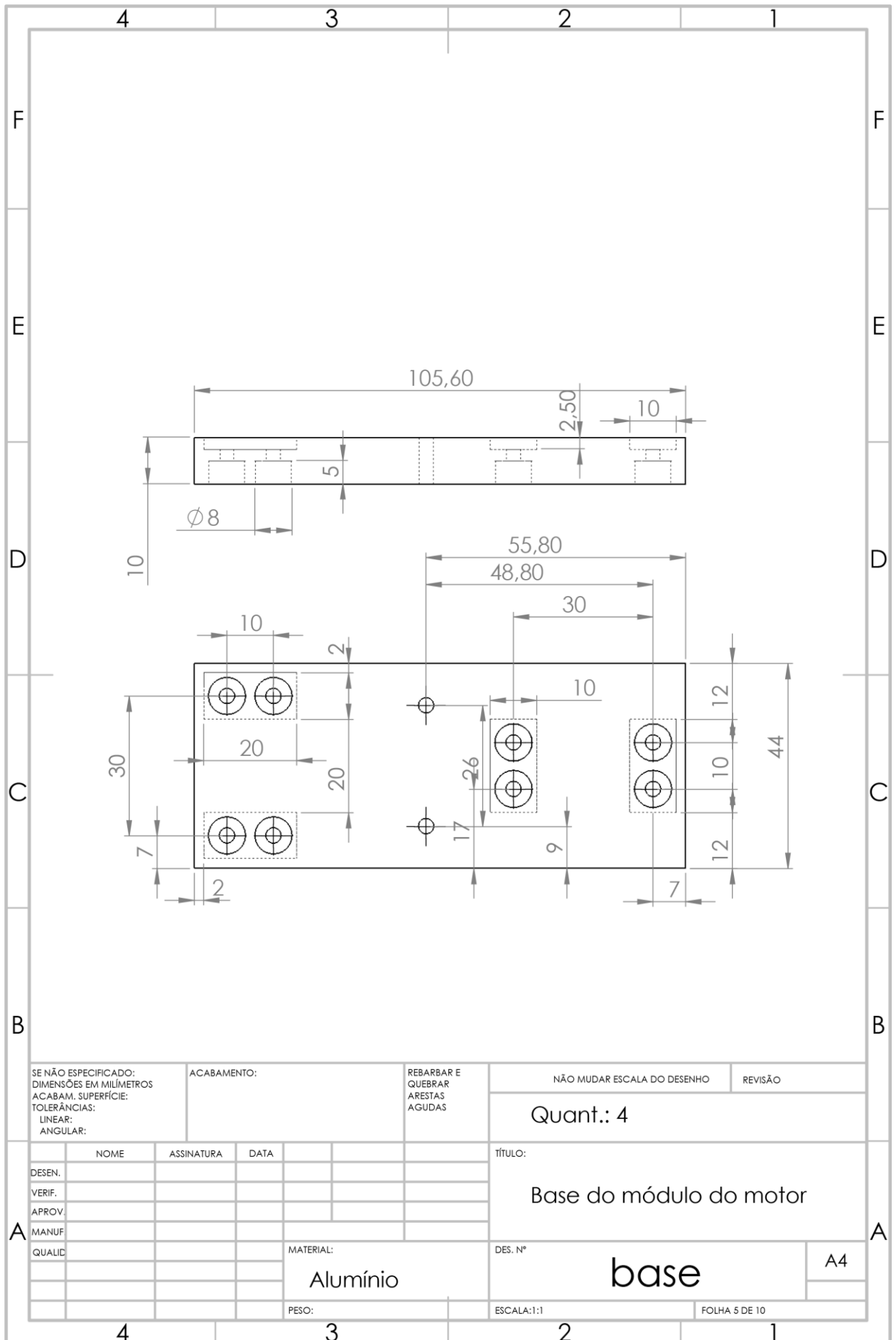
suportes

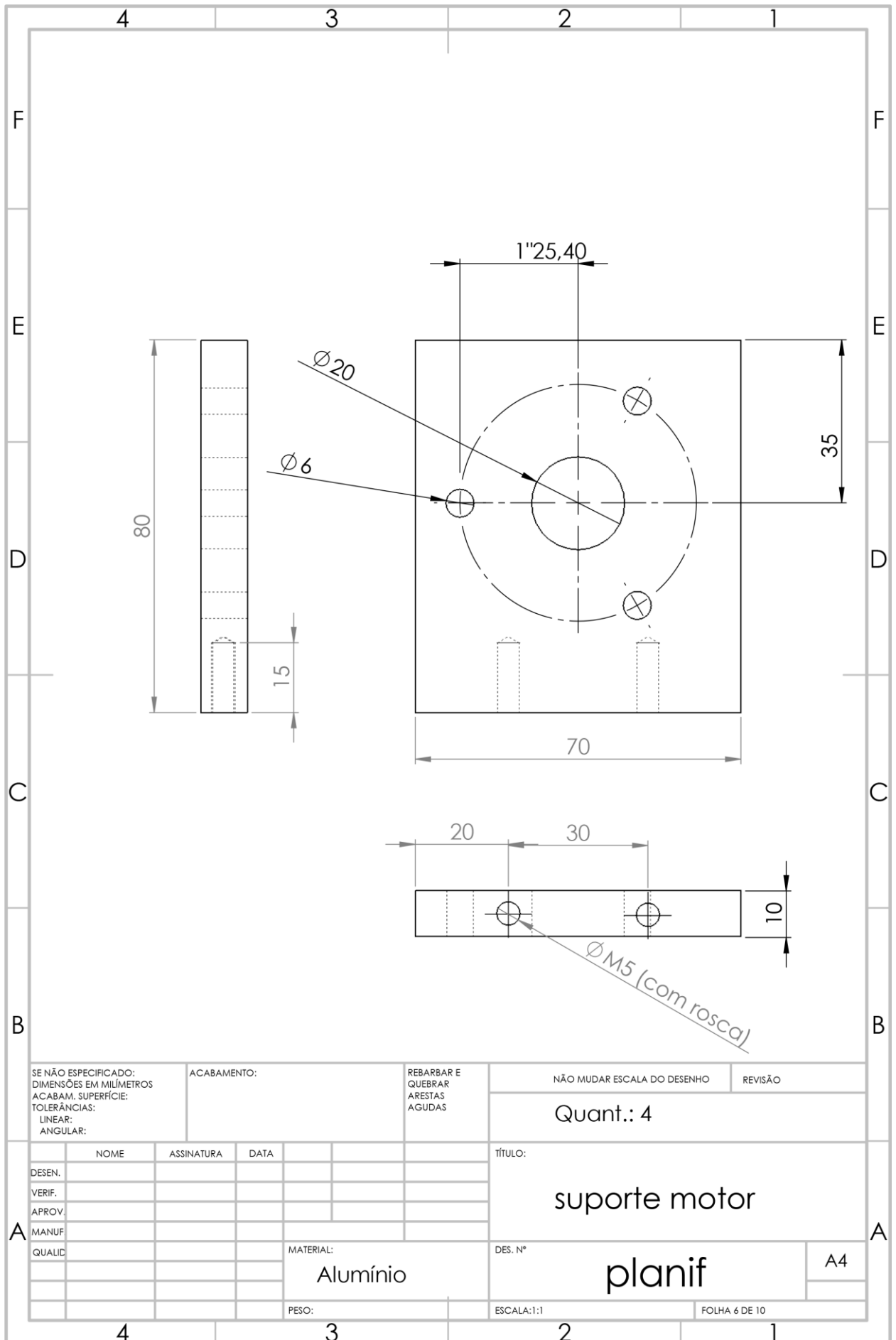
A4

PESO:

ESCALA:2:1

FOLHA 4 DE 10





SE NÃO ESPECIFICADO:
DIMENSÕES EM MILÍMETROS
ACABAM. SUPERFÍCIE:
TOLERÂNCIAS:
LINEAR:
ANGULAR:

ACABAMENTO:

REBARBAR E
QUEBRAR
ARESTAS
AGUDAS

NÃO MUDAR ESCALA DO DESENHO

REVISÃO

Quant.: 4

	NOME	ASSINATURA	DATA	
DESEN.				
VERIF.				
APROV.				
MANUF.				
QUALID.				

TÍTULO:

suporte motor

MATERIAL:

Alumínio

DES. N°

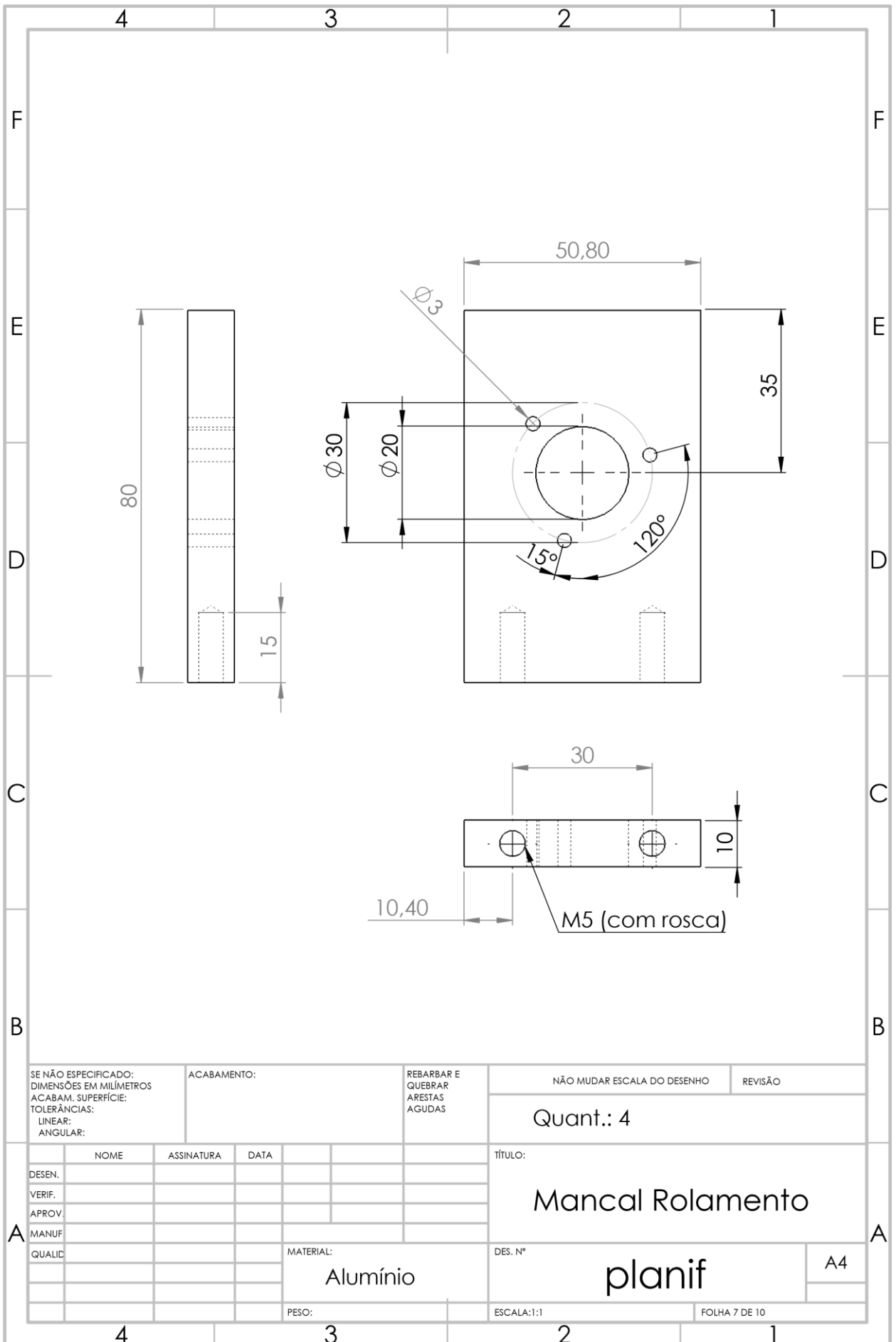
planif

A4

PESO:

ESCALA:1:1

FOLHA 6 DE 10



SE NÃO ESPECIFICADO:
DIMENSÕES EM MILÍMETROS
ACABAM. SUPERFÍCIE:
TOLERÂNCIAS:
LINEAR:
ANGULAR:

ACABAMENTO:

REBARBAR E
QUEBRAR
ARESTAS
AGUDAS

NÃO MUDAR ESCALA DO DESENHO

REVISÃO

Quant.: 4

	NOME	ASSINATURA	DATA	
DESEN.				
VERIF.				
APROV.				
MANUF.				
QUALID.				
				MATERIAL:
				Alumínio
				PESO:

TÍTULO:

Mancal Rolamento

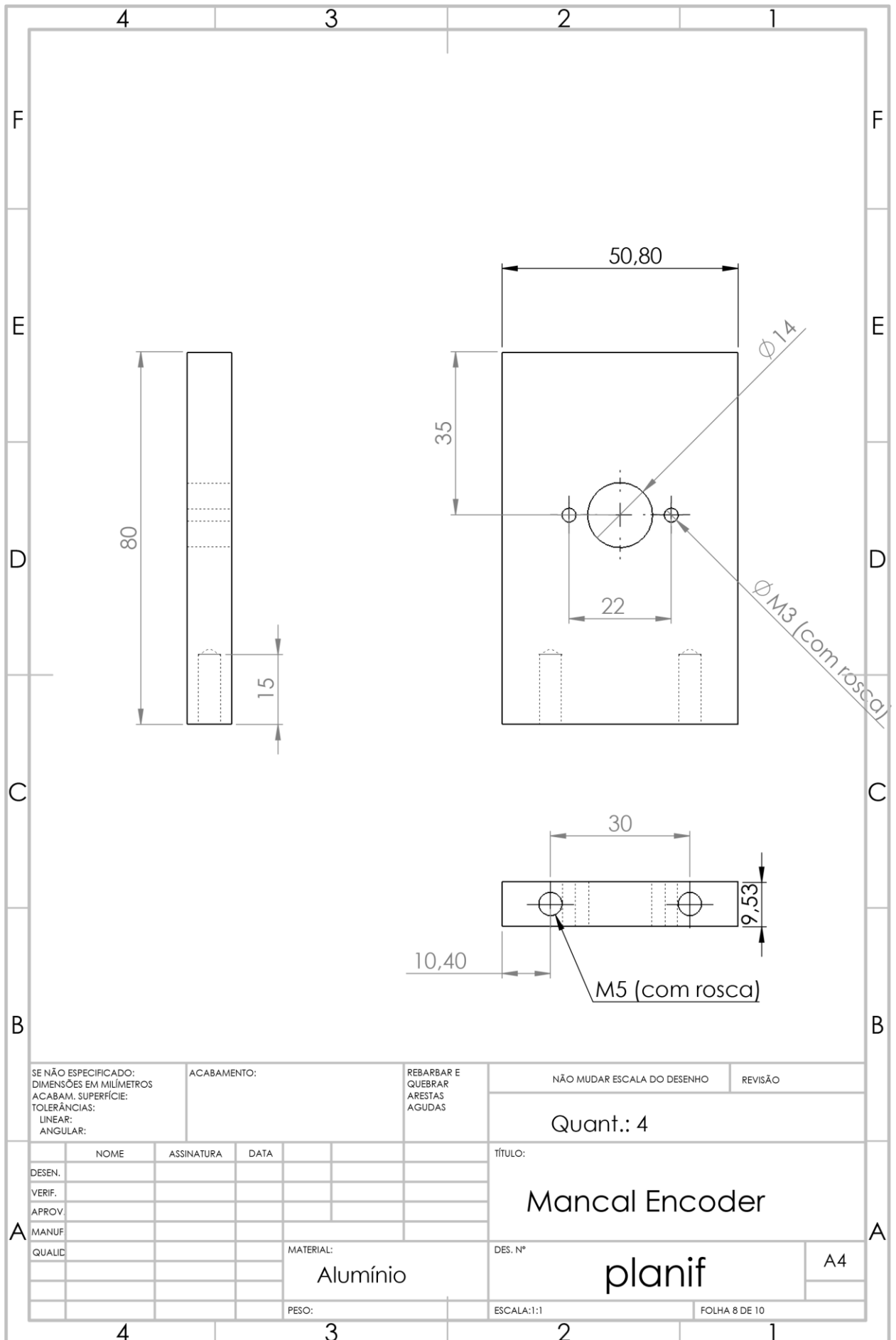
DES. N°

planif

A4

ESCALA:1:1

FOLHA 7 DE 10



SE NÃO ESPECIFICADO:
DIMENSÕES EM MILÍMETROS
ACABAM. SUPERFÍCIE:
TOLERÂNCIAS:
LINEAR:
ANGULAR:

ACABAMENTO:

REBARBAR E
QUEBRAR
ARESTAS
AGUDAS

NÃO MUDAR ESCALA DO DESENHO

REVISÃO

Quant.: 4

	NOME	ASSINATURA	DATA
DESEN.			
VERIF.			
APROV.			
MANUF.			
QUALID.			

TÍTULO:

Mancal Encoder

MATERIAL:

Alumínio

DES. Nº

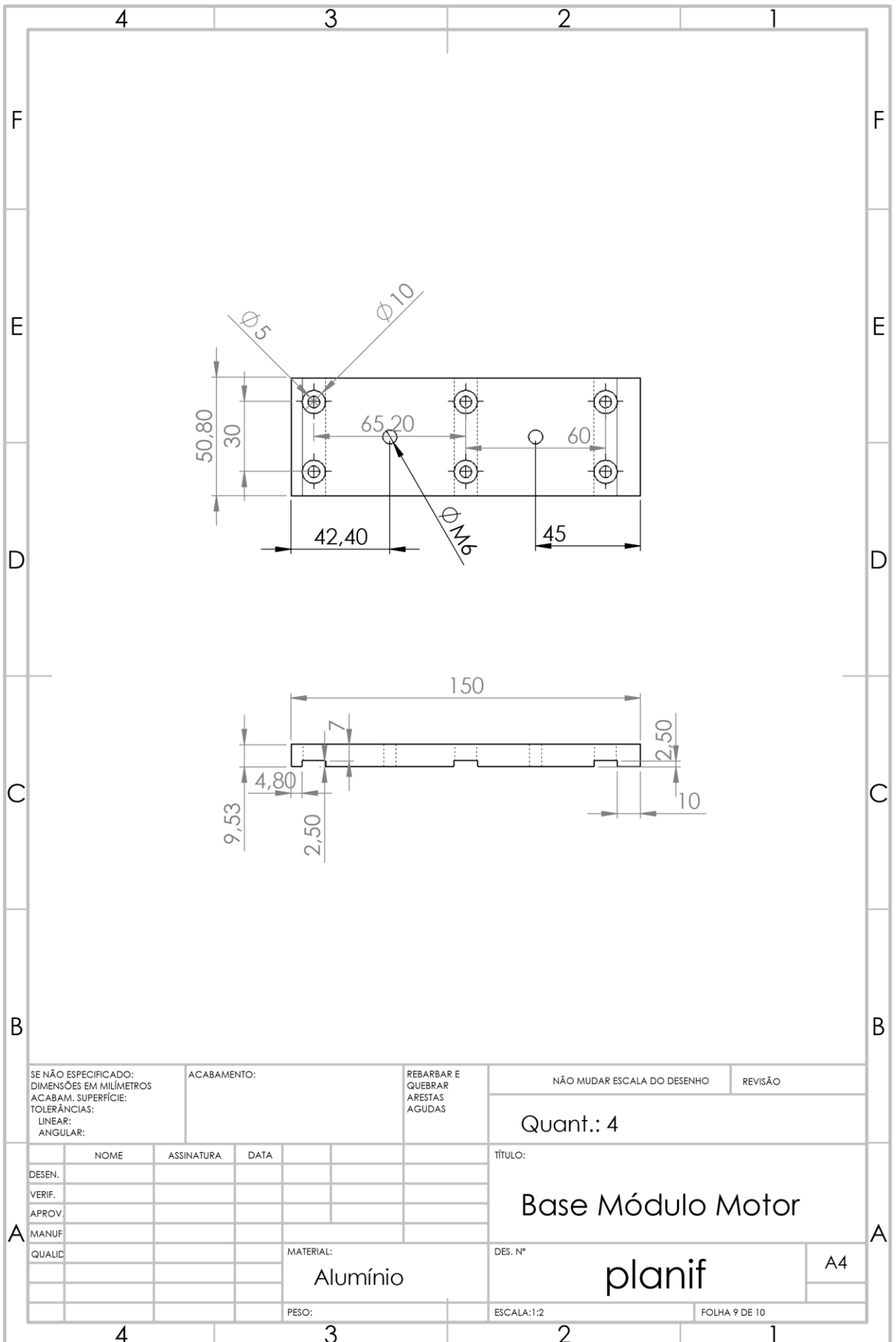
planif

A4

PESO:

ESCALA:1:1

FOLHA 8 DE 10



SE NÃO ESPECIFICADO:
DIMENSÕES EM MILÍMETROS
ACABAM. SUPERFÍCIE:
TOLERÂNCIAS:
LINEAR:
ANGULAR:

ACABAMENTO:

REBARBAR E
QUEBRAR
ARESTAS
AGUDAS

NÃO MUDAR ESCALA DO DESENHO

REVISÃO

Quant.: 4

TÍTULO:

Base Módulo Motor

DES. N°

planif

A4

MATERIAL:

Alumínio

PESO:

ESCALA:1:2

FOLHA 9 DE 10

B

B

C

C

D

D

E

E

F

F

4

3

2

1

4

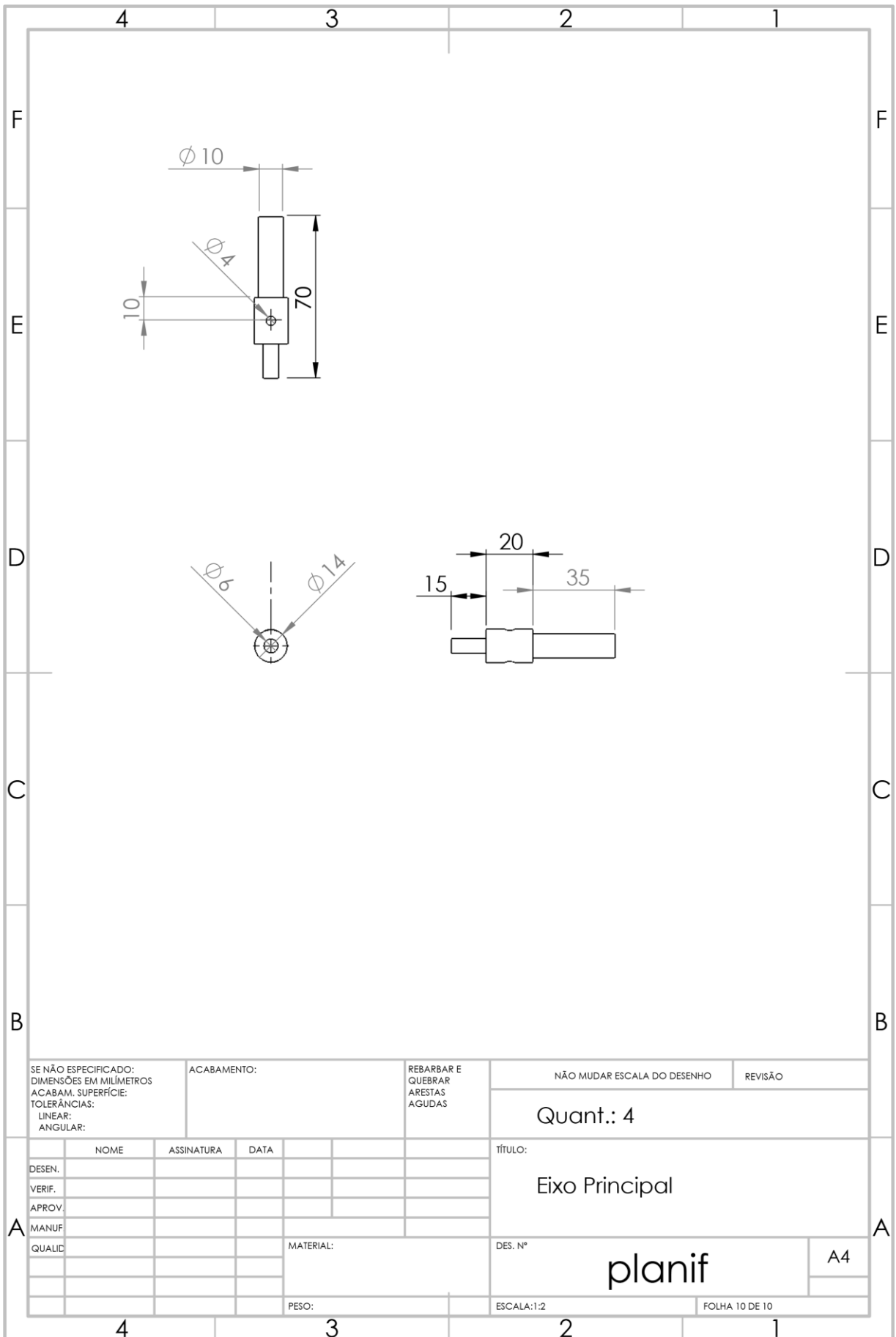
3

2

1

A

A



SE NÃO ESPECIFICADO:
DIMENSÕES EM MILÍMETROS
ACABAM. SUPERFÍCIE:
TOLERÂNCIAS:
LINEAR:
ANGULAR:

ACABAMENTO:

REBARBAR E
QUEBRAR
ARESTAS
AGUDAS

NÃO MUDAR ESCALA DO DESENHO

REVISÃO

Quant.: 4

	NOME	ASSINATURA	DATA		
DESEN.					
VERIF.					
APROV.					
MANUF.					
QUALID.					
				MATERIAL:	
				PESO:	

TÍTULO:

Eixo Principal

DES. Nº

planif

A4

ESCALA:1:2

FOLHA 10 DE 10

Códigos aplicados para modelagem dinâmica das esteiras bipartidas

```

%%%%%%%% Modelagem matemática - Skywalker Delta %%%%%%%%%
%%%%%%%% Lucas Antônio Oliveira Rodrigues, MSc.
%%%%%%%% Universidade Federal de Uberlândia - UFU

clear all
close all
clc

%% Definição das variáveis

L=0.95;           %Comprimento útil da esteira [m]
e=0.05;          %Espessura da esteira [m]
dmax= 0.0762;    %Altura máxima do drop desejada [m]
L_p= 0.1016;     %Comprimento do pistão pneumático (fechado)[m]
alpha=deg2rad(30); %ângulo de transmissão mínimo permitido [°]
P= 150;          %Massa máxima do usuário [kg]
p_e=26;         %Massa aproximada da esteira [kg]

H_p=L_p*sin(alpha)+e+dmax; %Altura da rótula do pistão

%% Variação da distância

d= @(x) H_p-(L_p+x)*sin(acos((L_p/(L_p+x))*cos(alpha)))-e;

lpm = fzero(d,0.1); %Extensão máxima do pistão pneumático [m]
tf = 0.2;          %tempo limite para a movimentação;
t=0:1e-3:tf;      %Variação do tempo [s]

beta=lpm/tf;
x=beta.*t;        %Variação de comprimento do pistão
pneumático[m]

d_plot=zeros(1,length(x));
for i=1:length(x)
    d_plot(i)=d(x(i))*100;
end

figure
plot(x*100,d_plot,'LineWidth',2)
grid on
title('Descida das esteiras', 'FontSize', 16)
xlabel('Abertura do pistão [cm]')
ylabel('distância esteira/chão [cm]')

%% Animação

```

```

D=L_p*cos(alpha)+L;

%pontos fixos
n1= [0 H_p];
n4= [D dmax];

x_a= [x fliplr(x)];
fig = figure;
filename = 'drop.gif';
pause on
for i=1:length(x_a)
    %pontos móveis
    n2= [L_p*(L_p/(L_p+x_a(i)))*cos(alpha) H_p-e-
L_p*sin(acos((L_p/(L_p+x_a(i)))*cos(alpha)))];
    n3= [L_p*cos(alpha) d(x_a(i))];

    plot([n1(1) n2(1)],[n1(2) n2(2)], 'r-o', [n2(1) n3(1)],[n2(2) n3(2)],
'g-o', [n3(1) n4(1)],[n3(2) n4(2)], 'b-o', 'LineWidth',2)
    axis([0 1.4 0 0.4]);
    title('Animação do movimento das esteiras','FontSize',16)
    grid on
    drawnow
    pause(0.015)
%     frame=getframe(fig);
%     im=frame2im(frame);
%     [imind,cm] = rgb2ind(im,256);
%     if i==1
%         imwrite(imind,cm,filename,'gif','Loopcount',inf);
%     else
%         imwrite(imind,cm,filename,'gif','WriteMode','append');
%     end
end
pause off

%% Carga Estática

%Considerando o caso onde a carga é suportado pelo vínculo

fr=1; %fator de redução de carga devido ao suporte;
Rp=(P+p_e)/sin(alpha)*9.81*fr/2;

disp(['Carga estática sob o cilindro: ' num2str(Rp) '[N]'])

%% Modelagem dinâmica

% Velocidade
v_t = @(t) -beta * sqrt(0.1e1 - L_p ^ 2 / (beta * t + L_p) ^ 2 *
cos(alpha) ^ 2) - 0.1e1 / (beta * t + L_p) ^ 2 * (0.1e1 - L_p ^ 2 / (beta

```



```

* t + L_p) ^ 2 * cos(alpha) ^ 2) ^ (-0.1e1 / 0.2e1) * L_p ^ 2 *
cos(alpha) ^ 2 * beta;

% Acelerao
a_t = @(t) beta ^ 2 * (0.1e1 - L_p ^ 2 / (beta * t + L_p) ^ 2 *
cos(alpha) ^ 2) ^ (-0.1e1 / 0.2e1) * L_p ^ 2 / (beta * t + L_p) ^ 3 *
cos(alpha) ^ 2 + 0.1e1 / (beta * t + L_p) ^ 5 * (0.1e1 - L_p ^ 2 / (beta
* t + L_p) ^ 2 * cos(alpha) ^ 2) ^ (-0.3e1 / 0.2e1) * L_p ^ 4 *
cos(alpha) ^ 4 * beta ^ 2;

v_plot=zeros(1,length(t));
a_plot=zeros(1,length(t));
for i=1:length(x)
    v_plot(i)=v_t(t(i));
    a_plot(i)=a_t(t(i));
end

figure
subplot(3,1,1)
plot(t,d_plot,'b','LineWidth',2)
grid on
title('Deslocamento da descida','FontSize', 14)
xlabel('tempo [s]')
ylabel('distancia [cm]')
subplot(3,1,2)
plot(t,v_plot,'g','LineWidth',2)
grid on
title('Velocidade da descida', 'FontSize', 14)
xlabel('tempo [s]')
ylabel('Velocidade [m/s]')
subplot(3,1,3)
plot(t,a_plot,'r','LineWidth',2)
grid on
title('Acelerao da descida', 'FontSize', 14)
xlabel('tempo [s]')
ylabel('Acelerao [m/s^2]')

```

Otimização das barras do Módulo de Suporte Ativo de Peso Corporal baseada na modelagem matemática

Função principal

```
function principal

clc
close all
format long

VTR = 1.e-20;
D =12; % número de variáveis de projeto (problema)

XVmin = [.2 .2 .25*ones(1,10)]; % vetor que define o limite inferior
para o vetor de variáveis de projeto
XVmax = [.5 .5 .45*ones(1,10)]; % vetor que define o limite superior
para o vetor de variáveis de projeto
%
y=0.4; % vetor de parâmetros de entrada (no modelo a ser otimizado)
NP = 50; % tamanho da população
itermax = 200; % número máximo de gerações
F = 0.5; % taxa de perturbação
CR = 0.8; % probabilidade de cruzamento
strategy=7; % estratégia a ser utilizada (default)
refresh=10; % nível de impressão na tela - não precisa alterar esse
valor ...

[X,FO,NF]=differential_evolution('BWS_eval',VTR,D,XVmin,XVmax,y,NP,iterma
x,F,CR,strategy,refresh)
% X o projeto final obtido
% FO o valor da função objetivo
% NF o número de avaliações da função objetivo
```

Função Pseudo-Objetivo

```
function Ft=BWS_eval(R,y)
%Módulo de Elasticidade - Alumínio
%E= 69e9;

%limite de escoamento - Alumínio
L_al=150; % [MPa]

%Perfil: tubo retangular

esp=5; %espessura da barra
%A= (1e-3*25.4)*(1e-3*25.4)-(1e-3*(25.4-esp))*(1e-3*(25.4-esp));
```

```

I= (1e-3*25.4)*(1e-3*25.4)^3/12-((1e-3*(25.4-esp)*(1e-3*(25.4-
esp))^3)/12);
g=0;

%1. Parâmetros do problema:
t_c=0.4; %1.1
Tempo de ciclo [s];
Mp=150; %1.2
Massa crítica do sistema [kg];
d_1= 0.2;
%d = [d_1*sin(deg2rad(25)) 0.15 0.2 0.15 d_1*sin(deg2rad(25))]; %1.3
Trajetória das articulações [m];

%2. Acelerações [m/s^2]
a= [[1.25000000000000 0.93750000000000 0.625000000000000
0.937500000000000 1.25000000000000]];

%3. Forças ativas [N]
f= [Mp/2*a(1) Mp/2*(g+a(2)) Mp*a(3) Mp/2*(g+a(4)) Mp/2*a(5)];

%% Cálculo do torque necessário
%barras r_2 e r_3 do mecanismo de acionamento:
offset=y; %distância entre o atuador e o
trilho do atuador;
r_2=R(1);
r_3=R(2);

ptos= 100; %resolução adotada;
t= linspace(0,t_c/2,ptos);
x=zeros(5,ptos);
for i=1:5
    x(i,:)=a(i).*t.^2/2;
end

q=zeros(5,ptos);
T=zeros(5,ptos);
Tmax=zeros(5,1);

for i=1:5
    q(i,:)=real(acos((-
r_3^2+r_2^2+(x(i,)+offset).^2)./(2*r_2.*(x(i,)+offset))));
    T(i,:)=f(i)*r_2*sin(q(i,:));
    Tmax(i)=max(abs(T(i,:)));
end

%% Cálculo da Velocidade angular de r_2

%velocidade máxima na trajetória de cada atuador:
v=zeros(5,1);

```

```

omega_2=zeros(5,1);
for i=1:5
    v(i)=sqrt(2*a(i)*(x(i,end)));
    omega_2(i)=(v(i)/(r_2*sin(q(i,end))))*...
        ((sin(pi-asin((r_2/r_3)*cos(q(i,end)))))/sqrt(1-
((r_2^2*cos(q(i,end))^2)/r_3^2)))^-1;
end

RPM=omega_2.*60/(2*pi);

%% Potência do motor elétrico

P=Tmax.*RPM/9550;

%% Análise das tensões - tração/compressão

%comprimentos das barras

r = R(3:12);
%barras    11  12  21  22  31  32  41  42  51  52
%Cálculo das tensões
f_barras=[];
sigma=zeros(1,length(r));

for i=1:5
    f_barras= [f_barras f(i) f(i)];
end
for i=1:length(r)
    sigma(i)=1e-6*f_barras(i)*(r(i)/2)*0.0254/(2*I);
end

%Restrições:
g=zeros(1,length(sigma)+1+length(q)+size(q,1)); %desigualdade
h=zeros(1,2); %igualdade

%geométricas:
g(1)=(r_2+r_3)/(max(max(x))+offset)-1;

%singularidades;
for i=1:size(q,1)
    for j=1:size(q,2)
        if (abs(r_2*cos(q(i,j))-r_3*sqrt(1-
((r_2^2*cos(q(i,j))^2)/(r_3^2))))<=1e-3)
            h(1)=1e12;
            break;
        end
    end
end
end
end

```

```

if max(abs(imag(P)))>=1e-6
    h(2)=1e12;
end

%limite de escoamento das tensões:
for i=1:length(sigma)
    g(i+1)=sigma(i)/L_al;
end

%ângulo de transmissão.
for i=1:length(q)
    g(length(sigma)+1+i)=-q(i)/degtorad(30)+1;
end

if max(abs(imag(g)))>1e-6
    h(2)=1e12;
end

%Resolução do atuador
delta_q=max(rad2deg(q),[],2)-min(rad2deg(q),[],2);
delta_s=max(x,[],2);

reso=(delta_s*1e3)./delta_q;

for i=1:size(q,1)
    g(length(sigma)+1+length(q)+i)=reso(i)/5-1;
end

%função objetivo;

F0= max(P);

%fator Rp:
Rp=1e2;

%função pseudo-objetivo:
Ft=F0+h(1)+max(1e-10,h(2))^2;
for i=1:length(g)
    Ft=Ft+Rp*(max(1e-10,g(i))^2);
end

end

```

Algoritmo de evolução-diferencial

```

function [bestmem,bestval,nfeval] =
de(fname,VTR,D,XVmin,XVmax,y,NP,itermax,F,CR,strategy,refresh);
% minimization of a user-supplied function with respect to x(1:D),

```

```

% using the differential evolution (DE) algorithm of Rainer Storn
% (http://www.icsi.berkeley.edu/~storn/code.html)
%
% Special thanks go to Ken Price (kprice@solano.community.net) and
% Arnold Neumaier (http://solon.cma.univie.ac.at/~neum/) for their
% valuable contributions to improve the code.
%
% Strategies with exponential crossover, further input variable
% tests, and arbitrary function name implemented by Jim Van Zandt
% <jrv@vanzandt.mv.com>, 12/97.
%
% Output arguments:
% -----
% bestmem      parameter vector with best solution
% bestval      best objective function value
% nfeval       number of function evaluations
%
% Input arguments:
% -----
%
% fname        string naming a function f(x,y) to minimize
% VTR          "Value To Reach". devc3 will stop its minimization
%              if either the maximum number of iterations "itermax"
%              is reached or the best parameter vector "bestmem"
%              has found a value f(bestmem,y) <= VTR.
% D           number of parameters of the objective function
% XVmin       vector of lower bounds XVmin(1) ... XVmin(D)
%              of initial population
%              *** note: these are not bound constraints!! ***
% XVmax       vector of upper bounds XVmax(1) ... XVmax(D)
%              of initial population
% y           problem data vector (must remain fixed during the
%              minimization)
% NP          number of population members
% itermax     maximum number of iterations (generations)
% F           DE-stepsize F from interval [0, 2]
% CR          crossover probability constant from interval [0, 1]
% strategy    1 --> DE/best/1/exp          6 --> DE/best/1/bin
%              2 --> DE/rand/1/exp         7 --> DE/rand/1/bin
%              3 --> DE/rand-to-best/1/exp 8 --> DE/rand-to-
best/1/bin
%              4 --> DE/best/2/exp          9 --> DE/best/2/bin
%              5 --> DE/rand/2/exp         else DE/rand/2/bin
%              Experiments suggest that /bin likes to have a slightly
%              larger CR than /exp.
% refresh     intermediate output will be produced after "refresh"
%              iterations. No intermediate output will be produced
%              if refresh is < 1
%

```

```

% The first four arguments are essential (though they have
% default values, too). In particular, the algorithm seems to
% work well only if [XVmin,XVmax] covers the region where the
% global minimum is expected. DE is also somewhat sensitive to
% the choice of the stepsize F. A good initial guess is to
% choose F from interval [0.5, 1], e.g. 0.8. CR, the crossover
% probability constant from interval [0, 1] helps to maintain
% the diversity of the population and is rather uncritical. The
% number of population members NP is also not very critical. A
% good initial guess is 10*D. Depending on the difficulty of the
% problem NP can be lower than 10*D or must be higher than 10*D
% to achieve convergence.
% If the parameters are correlated, high values of CR work better.
% The reverse is true for no correlation.
%
% default values in case of missing input arguments:
% VTR = 1.e-6;
% D = 2;
% XVmin = [-2 -2];
% XVmax = [2 2];
% y=[];
% NP = 10*D;
% itermax = 200;
% F = 0.8;
% CR = 0.5;
% strategy = 7;
% refresh = 10;
%
% Cost function: function result = f(x,y);
% has to be defined by the user and is minimized
% w.r. to x(1:D).
%
% Example to find the minimum of the Rosenbrock saddle:
% -----
% Define f.m as:
% function result = f(x,y);
% result = 100*(x(2)-x(1)^2)^2+(1-x(1))^2;
% end
% Then type:
%
% VTR = 1.e-6;
% D = 2;
% XVmin = [-2 -2];
% XVmax = [2 2];
% [bestmem,bestval,nfeval] = devec3("f",VTR,D,XVmin,XVmax);
%
% The same example with a more complete argument list is handled in
% run1.m
%

```

```

% About devec3.m
% -----
% Differential Evolution for MATLAB
% Copyright (C) 1996, 1997 R. Storn
% International Computer Science Institute (ICSI)
% 1947 Center Street, Suite 600
% Berkeley, CA 94704
% E-mail: storn@icsi.berkeley.edu
% WWW:    http://http.icsi.berkeley.edu/~storn
%
% devec is a vectorized variant of DE which, however, has a
% property which differs from the original version of DE:
% 1) The random selection of vectors is performed by shuffling the
%    population array. Hence a certain vector can't be chosen twice
%    in the same term of the perturbation expression.
%
% Due to the vectorized expressions devec3 executes fairly fast
% in MATLAB's interpreter environment.
%
% This program is free software; you can redistribute it and/or modify
% it under the terms of the GNU General Public License as published by
% the Free Software Foundation; either version 1, or (at your option)
% any later version.
%
% This program is distributed in the hope that it will be useful,
% but WITHOUT ANY WARRANTY; without even the implied warranty of
% MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
% GNU General Public License for more details. A copy of the GNU
% General Public License can be obtained from the
% Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
%-----Check input variables-----
err=[];
if nargin<1, error('de 1st argument must be function name'); else
    if exist(fname)<1; err(1,length(err)+1)=1; end; end;
if nargin<2, VTR = 1.e-6; else
    if length(VTR)~=1; err(1,length(err)+1)=2; end; end;
if nargin<3, D = 2; else
    if length(D)~=1; err(1,length(err)+1)=3; end; end;
if nargin<4, XVmin = [-2 -2];else
    if length(XVmin)~=D; err(1,length(err)+1)=4; end; end;
if nargin<5, XVmax = [2 2]; else
    if length(XVmax)~=D; err(1,length(err)+1)=5; end; end;
if nargin<6, y=[]; end;
if nargin<7, NP = 10*D; else
    if length(NP)~=1; err(1,length(err)+1)=7; end; end;
if nargin<8, itermax = 250; else
    if length(itermax)~=1; err(1,length(err)+1)=8; end; end;
if nargin<9, F = 0.8; else

```



```

    if length(F)~=1; err(1,length(err)+1)=9; end; end;
if nargin<10, CR = 0.5; else
    if length(CR)~=1; err(1,length(err)+1)=10; end; end;
if nargin<11, strategy = 7; else
    if length(strategy)~=1; err(1,length(err)+1)=11; end; end;
if nargin<12, refresh = 10; else
    if length(refresh)~=1; err(1,length(err)+1)=12; end; end;
if length(err)>0
    disp(['error in parameter ' num2str(err)])
    usage('de
(string,scalar,scalar,vector,vector,any,integer,integer,scalar,scalar,int
eger,integer)');
end

if (NP < 5)
    NP=5;
    fprintf(1,' NP increased to minimal value 5\n');
end
if ((CR < 0) | (CR > 1))
    CR=0.5;
    fprintf(1,'CR should be from interval [0,1]; set to default value
0.5\n');
end
if (itermax <= 0)
    itermax = 200;
    fprintf(1,'itermax should be > 0; set to default value 200\n');
end
refresh = floor(refresh);

%-----Initialize population and some arrays-----
-

pop = zeros(NP,D); %initialize pop to gain speed

%---pop is a matrix of size NPxD. It will be initialized-----
%---with random values between the min and max values of the-----
%---parameters-----

for i=1:NP
    pop(i,:) = XVmin + rand(1,D).*(XVmax - XVmin);
end

popold    = zeros(size(pop));    % toggle population
val       = zeros(1,NP);        % create and reset the "cost array"
bestmem   = zeros(1,D);        % best population member ever
bestmemit = zeros(1,D);        % best population member in iteration
nfeval    = 0;                 % number of function evaluations

```

```

%-----Evaluate the best member after initialization-----
-

ibest = 1; % start with first population member
val(1) = feval(fname,pop(ibest,:),y);
bestval = val(1); % best objective function value so far
nfeval = nfeval + 1;
for i=2:NP % check the remaining members
    val(i) = feval(fname,pop(i,:),y);
    nfeval = nfeval + 1;
    if (val(i) < bestval) % if member is better
        ibest = i; % save its location
        bestval = val(i);
    end
end
bestmemit = pop(ibest,:); % best member of current iteration
bestvalit = bestval; % best value of current iteration

bestmem = bestmemit; % best member ever

%-----DE-Minimization-----
%-----popold is the population which has to compete. It is-----
%-----static through one iteration. pop is the newly-----
%-----emerging population.-----

pm1 = zeros(NP,D); % initialize population matrix 1
pm2 = zeros(NP,D); % initialize population matrix 2
pm3 = zeros(NP,D); % initialize population matrix 3
pm4 = zeros(NP,D); % initialize population matrix 4
pm5 = zeros(NP,D); % initialize population matrix 5
bm = zeros(NP,D); % initialize bestmember matrix
ui = zeros(NP,D); % intermediate population of perturbed
vectors
mui = zeros(NP,D); % mask for intermediate population
mpo = zeros(NP,D); % mask for old population
rot = (0:1:NP-1); % rotating index array (size NP)
rotD = (0:1:D-1); % rotating index array (size D)
rt = zeros(NP); % another rotating index array
rtd = zeros(D); % rotating index array for exponential
crossover
a1 = zeros(NP); % index array
a2 = zeros(NP); % index array
a3 = zeros(NP); % index array
a4 = zeros(NP); % index array
a5 = zeros(NP); % index array
ind = zeros(4);

iter = 1;
while (iter <= itermax)% & (bestval > VTR))

```

```

popold = pop; % save the old population

ind = randperm(4); % index pointer array

a1 = randperm(NP); % shuffle locations of vectors
rt = rem(rot+ind(1),NP); % rotate indices by ind(1) positions
a2 = a1(rt+1); % rotate vector locations
rt = rem(rot+ind(2),NP);
a3 = a2(rt+1);
rt = rem(rot+ind(3),NP);
a4 = a3(rt+1);
rt = rem(rot+ind(4),NP);
a5 = a4(rt+1);

pm1 = popold(a1,:); % shuffled population 1
pm2 = popold(a2,:); % shuffled population 2
pm3 = popold(a3,:); % shuffled population 3
pm4 = popold(a4,:); % shuffled population 4
pm5 = popold(a5,:); % shuffled population 5

for i=1:NP % population filled with the best
member
    bm(i,:) = bestmemit; % of the last iteration
end

mui = rand(NP,D) < CR; % all random numbers < CR are 1, 0
otherwise

if (strategy > 5)
    st = strategy-5; % binomial crossover
else
    st = strategy; % exponential crossover
    mui=sort(mui'); % transpose, collect 1's in each column
    for i=1:NP
        n=floor(rand*D);
        if n > 0
            rtd = rem(rotd+n,D);
            mui(:,i) = mui(rtd+1,i); %rotate column i by n
        end
    end
    mui = mui'; % transpose back
end
mpo = mui < 0.5; % inverse mask to mui

if (st == 1) % DE/best/1
    ui = bm + F*(pm1 - pm2); % differential variation
    ui = popold.*mpo + ui.*mui; % crossover
elseif (st == 2) % DE/rand/1
    ui = pm3 + F*(pm1 - pm2); % differential variation

```

```

    ui = popold.*mpo + ui.*mui;      % crossover
elseif (st == 3)                    % DE/rand-to-best/1
    ui = popold + F*(bm-popold) + F*(pm1 - pm2);
    ui = popold.*mpo + ui.*mui;      % crossover
elseif (st == 4)                    % DE/best/2
    ui = bm + F*(pm1 - pm2 + pm3 - pm4); % differential variation
    ui = popold.*mpo + ui.*mui;      % crossover
elseif (st == 5)                    % DE/rand/2
    ui = pm5 + F*(pm1 - pm2 + pm3 - pm4); % differential variation
    ui = popold.*mpo + ui.*mui;      % crossover
end

for ii=1:NP
    for jj=1:D
        if ui(ii,jj)<XVmin(jj);
            ui(ii,jj)=XVmin(jj);
        end
        if ui(ii,jj)>XVmax(jj);
            ui(ii,jj)=XVmax(jj);
        end
    end
end

tempfun = [];
%-----Select which vectors are allowed to enter the new population-----
-----
for i=1:NP
    tempval = feval(fname,ui(i,:),y); % check cost of competitor
    tempfun(i) = tempval;
    nfeval = nfeval + 1;
    if (tempval <= val(i)) % if competitor is better than value in "cost
array"
        pop(i,:) = ui(i,:); % replace old vector with new one (for new
iteration)
        val(i) = tempval; % save value in "cost array"

        %----we update bestval only in case of success to save time-----
        ----
        if (tempval < bestval) % if competitor better than the best
one ever
            bestval = tempval; % new best value
            bestmem = ui(i,:); % new best parameter vector ever
        end
    end
end %---end for imember=1:NP

bestmemit = bestmem; % freeze the best member of this iteration
for the coming

```

```

                                % iteration. This is needed for some of the
strategies.

%----Output section-----
----
%tempfun
%fprintf(1,'\n');
%population=ui(:,1:D)
fprintf(1,'\n');
average=1/NP*(sum(tempfun));
worst=max(tempfun);
grafico(iter,:)= [iter bestval average worst];
fprintf(1,'
=====
=====\\n');
    fprintf(1,'
                                Objective
Function \\n');
    fprintf(1,'
=====
=====\\n');
    if (refresh > 0)
        fprintf(1,' Iteration: %d, Best Value: %.11f, Average:
%.11f, Worst: %.11f\\n',iter,bestval,average,worst);
        for n=1:D
            fprintf(1,' x(%d) = %f',n,bestmem(n));
        end
        fprintf(1,'\n');
    end

    iter = iter + 1;
end %---end while ((iter < itermax) ...
fprintf(1,'
=====
=====\\n');

```

Modelagem dinâmica dos atuadores do Módulo de Suporte Ativo de Peso Corporal

```

%Programa de Pós-Graduação em Engenharia Mecânica
%Doutorando: Lucas Antônio O. Rodrigues, MSc.
%especificação do suporte corporal BWS

close all
format shortg
clc

%% Dados iniciais do problema

%Módulo de Elasticidade - Alumínio
E= 69e9;

%Perfil: tubo retangular

esp=5; %espessura da barra
A= (1e-3*25.4)*(1e-3*25.4)-(1e-3*(25.4-esp))*(1e-3*(25.4-esp));
I= (1e-3*25.4)*(1e-3*25.4)^3/12-((1e-3*(25.4-esp))*(1e-3*(25.4-
esp))^3)/12);
g=0;

%1. Parâmetros do problema:
t_c=0.4; %1.1
Tempo de ciclo [s];
Mp=150; %1.2
Massa crítica do sistema [kg];
d_1= 0.2;
%d = [d_1*sin(deg2rad(25)) 0.15 0.1 0.15 d_1*sin(deg2rad(25))]; %1.3
Trajetória das articulações [m];
d = [0.2 0.15 0.1 0.15 0.2]; % Apenas
para o Capítulo 2019
%2. Acelerações [m/s^2]
%a= [10.57*d_1 3.75 1 3.75 10.57*d_1];
a= [d(1)/t_c^2 d(2)/t_c^2 d(3)/t_c^2 d(4)/t_c^2 d(5)/t_c^2]; %Apenas
para o Capítulo 2019
%3. Forças ativas [N]
f= [Mp/2*a(1) Mp/2*(g+a(2)) Mp*a(3) Mp/2*(g+a(4)) Mp/2*a(5)];

%% Cálculo do torque necessário
%barras r_2 e r_3 do mecanismo de acionamento:
offset=0.4; %distância entre o atuador e
o trilho do atuador;
r_2=.25;
r_3=.3;

```

```

ptos=
100; %resoluçao
adotada;
t= linspace(0,t_c/2,ptos);
x=zeros(5,ptos);
for i=1:5
    x(i,:)=a(i).*t.^2/2;
end

q=zeros(5,ptos);
T=zeros(5,ptos);
Tmax=zeros(5,1);

for i=1:5
    q(i,:)=real(acos((-
r_3^2+r_2^2+(x(i,:)+offset).^2)./(2*r_2.*(x(i,:)+offset))));
    T(i,:)=f(i)*r_2*sin(q(i,:));
    Tmax(i)=max(abs(T(i,:)));
end

%% Cálculo da Velocidade angular de r_2

%velocidade máxima na trajetória de cada atuador:
v=zeros(5,1);
omega_2=zeros(5,1);
for i=1:5
    v(i)=sqrt(2*a(i)*(x(i,end)));
    omega_2(i)=(v(i)/(r_2*sin(q(i,end))))*...
        ((sin(pi-asin((r_2/r_3)*cos(q(i,end)))))/sqrt(1-
((r_2^2*cos(q(i,end))^2)/r_3^2)))^-1;
end

RPM=omega_2.*60/(2*pi);

%% Potência do motor elétrico

P=Tmax.*RPM/9550;

%% Análise das tensões - tração/compressão

%comprimentos das barras

r = [ .3 .3 .3 .3 .3 .3 .3 .3 .3 .3 ];
%barras 11 12 21 22 31 32 41 42 51 52
%Cálculo das tensões
f_barras=[];
sigma=zeros(1,length(r));

for i=1:5

```

```

    f_barras= [f_barras f(i) f(i)];
end
for i=1:length(r)
    sigma(i)=1e-6*f_barras(i)*(r(i)/2)*0.0254/(2*I);
end

%% singularidades

for i=1:size(q,1)
    for j=1:size(q,2)
        if (abs(r_2*cos(q(i,j))-r_3*sqrt(1-
((r_2^2*cos(q(i,j))^2)/(r_3^2))))<=1e-3)
            h=1e12;
            disp('POSICAO SINGULAR DETECTADA!')
            disp(['q_ ' num2str(i) ', indice ' num2str(j) '. Valor: '
num2str(rad2deg(q(i,j)))])
            %break;
        end
    end
end
end

%resolucao do atuador
delta_q=max(rad2deg(q),[],2)-min(rad2deg(q),[],2);
delta_s=max(x,[],2);

reso=(delta_s*1e3)./delta_q;

%% Resultados

disp('Potencia necessaria calculada por atuador [W]: ')
disp(P*1000)
disp('Rotação máxima observada por atuador [RPM]: ')
disp(RPM)
disp('Resolução dos atuadores [mm/°]: ')
disp(reso)

```


Otimização dos parâmetros de controle dos atuadores

Função principal

```

clc
clear all
close all
format long

VTR = 1.e-20;
D =3; % número de variáveis de projeto (problema)

XVmin = [0.1 0 0]; % vetor que define o limite inferior para o vetor de
variáveis de projeto
XVmax = [2 1 1]; % vetor que define o limite superior para o vetor de
variáveis de projeto

%configurando comunicação:
[s,flag] = setupSerial('COM12');

%-----
motor = 0; %Motor a otimizar
%-----

y= s; % vetor de parâmetros de entrada (no modelo a ser otimizado)
NP = 30; % tamanho da população
itermax = 10; % número máximo de gerações
F = 0.5; % taxa de perturbação
CR = 0.8; % probabilidade de cruzamento
strategy=7; % estratégia a ser utilizada (default)
refresh=10; % nível de impressão na tela - não precisa alterar esse
valor ...
K_final = [];

%MOTOR 0
if motor ==0
    [X,FO,NF]=differential_evolution('eval_PID0',VTR,D,XVmin,XVmax,y,NP,itermax,F,CR,strategy,refresh)
    % X o projeto final obtido
    % FO o valor da função objetivo
    % NF o número de avaliações da função objetivo
    K_final = [K_final; motor X(1) X(2) X(3)];
end

%MOTOR 1
if motor ==1

```

```

[X,FO,NF]=differential_evolution('eval_PID1',VTR,D,XVmin,XVmax,y,NP,it
ermax,F,CR,strategy,refresh)
% X o projeto final obtido
% FO o valor da funo objetivo
% NF o nmero de avaliaes da funo objetivo
K_final = [K_final; motor X(1) X(2) X(3)];
end

%MOTOR 2
if motor ==2
[X,FO,NF]=differential_evolution('eval_PID2',VTR,D,XVmin,XVmax,y,NP,it
ermax,F,CR,strategy,refresh)
% X o projeto final obtido
% FO o valor da funo objetivo
% NF o nmero de avaliaes da funo objetivo
K_final = [K_final; motor X(1) X(2) X(3)];
end

%MOTOR 3
if motor ==3
[X,FO,NF]=differential_evolution('eval_PID3',VTR,D,XVmin,XVmax,y,NP,it
ermax,F,CR,strategy,refresh)
% X o projeto final obtido
% FO o valor da funo objetivo
% NF o nmero de avaliaes da funo objetivo
K_final = [K_final; motor X(1) X(2) X(3)];
end

%MOTOR 4
if motor ==4
[X,FO,NF]=differential_evolution('eval_PID4',VTR,D,XVmin,XVmax,y,NP,it
ermax,F,CR,strategy,refresh)
% X o projeto final obtido
% FO o valor da funo objetivo
% NF o nmero de avaliaes da funo objetivo
K_final = [K_final; motor X(1) X(2) X(3)];
end

fclose(s);
format shortg

props = java.lang.System.getProperties;
props.setProperty('mail.smtp.auth','true');
setpref('Internet','SMTP_Username','lucasaor@ufu.br');
setpref('Internet','SMTP_Password','*****');

```

```

sendmail('lucas.ep@outlook.com','MATLAB - Otimizaç?o conclu?da', ...
['A otimizaç?o do controle dos motores da HOPE-G foi conclu?da com
sucesso. ' 10 ...
'Segue abaixo os resultados obtidos' 10 10 ...
'Atuador' 'K_p' 'K_i' 'K_d' 10 ...
num2str(K_final) 10 ...
'Os resultados tamb?m foram salvos no OneDrive no arquivo
"results.txt. Espero n?o ter destru?do o laborat?rio."' 10 10 ...
'Att;' 10 10 ...
'MATLAB bot']);

save('results.txt','-ascii','K_final');

```

função pseudo-objetivo

```

function result = eval_PID(x,y)

% Enviando Par?metros:
PIDsend(y(1),x(1),x(2),x(3),y(2));

% Lendo resposta
[T_s,osh,erro] = PIDget(y(1));

% minimizar o tempo de acomodaç?o, eliminar overshoot e erro
result= T_s/2 + osh^2 + abs(erro^3);

```

Função de requisição à API de otimização

```

function [flag] = PIDsend(s,Kp, Ki, Kd,motor)
%sending PID data to arduino:
    a='a';

    fprintf(s,'%c','M');
    while (a ~= 'M')
        a=fread(s,1,'uchar');
    end
    %disp(['PIDsend_return_1:' a])
    if (a=='M')
        fprintf(s,'%d',motor);
    end

    fprintf(s,'%c','P');

```

```

while (a ~= 'P')
    a=fread(s,1,'uchar');
end
%disp(['PIDsend_return_1:' a])
if (a=='P')
    fprintf(s,'%f',Kp);
end

fprintf(s,'%c','I');
while (a ~= 'I')
    a=fread(s,1,'uchar');
end;
%disp(['PIDsend_return_2:' a])
if (a=='I')
    fprintf(s,'%f',Ki);
end

fprintf(s,'%c','D');
while (a ~= 'D')
    a=fread(s,1,'uchar');
end
%disp(['PIDsend_return_3:' a])
if (a=='D')
    fprintf(s,'%f',Kd);
end
Kp
Ki
Kd
end

```

Função de leitura da resposta da API de otimização

```

function [p1,p2, p3] = PIDget(s)
a='a';
fprintf(s,'%c','T');

while (a ~= 'T')
    a=fread(s,1,'uchar');
end
%disp(['PIDget_return_1:' a])
if (a=='T')
    p1 = fscanf(s,'%e');
end

fprintf(s,'%c','O');
while (a ~= 'O')
    a=fread(s,1,'uchar');
end;

```

```
%disp(['PIDget_return_2:' a])
if (a=='0')
    p2 = fscanf(s,'%e');
end

fprintf(s,'%c','E');
while (a ~= 'E')
    a=fread(s,1,'uchar');
end
%disp(['PIDget_return_3:' a])
if (a=='E')
    p3 = fscanf(s,'%e');
end

end
```

Código-fonte da central de controle (microcontrolador)

```

/* HOPE-G
   PROJECT ARDUINO-UNITY BWS HEALTH CHECK-UP v0.5.0
   Last Updated:04/01/2022
   Code Development: Rodrigues, L.A.O., MSc.

*/
// Libraries used

#include <Encoder.h>
#include <PID_v1.h>

//Pending Changes:
// Set a timer for every motor movement;
// Create an "Idle" position for every motor;

//----- Variables and Pins-----
// Values are assigned to motors 0, 1, 2, 3 and 4, respectively;

int encodPinA1[] = {2, 20, 18, 19, 3}; //Encoder A Pins;
int encodPinB1[] = {23, 21, 16, 17, 15}; // Encoder B pins;
int M1[] = {31, 32, 24, 36, 39}; // Motor Direction pin 1; RED
WIRE of the motor ALWAYS ON A!
int M2[] = {30, 33, 25, 37, 38}; // Motor Direction pin 2; GREEN
WIRE of the motor ALWAYS ON B!
int pwm1[] = {8, 9, 4, 5, 6}; // Motor PWM pin;
int startPoint[] = {42, 44, 46, 48, 50}; // Start course sensor pin;
int endPoint[] = {43, 45, 47, 49, 51}; // End course sensor pin.
int motor = 0;
int prevInput = 0;
int motorKiller = 3;

//double X = 1 / 4.5; //for Y = 0, X:
//double Y = -475 / 4.5;
//PID Parameters
double kp[] = {1, 1.7893, 4, 1.782, 4}; //Proportional Gain
double ki[] = {0.01, 0, 0, 0, 0}; //Integrative Gain
double kd[] = {0.00007, 0.0, 0.12, 0.0852, 0.0965}; //Derivative Gain
double input[] = {0, 0, 0, 0, 0}; //motor
input from encoders (control error signal)
double output[] = {0, 0, 0, 0, 0}; //Motor
PWM output (Control output to plant)
double setpoint[] = {0, 0, 0, 0, 0}; //Reference signal (This parameter
represents the desired position of the actuators)

```

```

//Time Control;
int interval = 500;
int interval2 = 2000; //Update intervals of the player position in ms

volatile long timer; //Timers to control the update rate
volatile long timer0;
volatile long timer1; //Timers to control the pose transitions
volatile long timer2; //timer to control output

char Pos = 0; //Configuration of the platform: f-> Neutral; g-> Roll
left; h-> Roll right; i-> Yaw left; j-> Yaw right;
char prevPos = 'f';
int currentPosition = 1;
int firstDiff = 1;
int diff = 0;
int EncThres = 25;

//----- Object constructors-----

//PID controllers
PID myPID0(&input[0], &output[0], &setpoint[0], kp[0], ki[0], kd[0],
DIRECT);
PID myPID1(&input[1], &output[1], &setpoint[1], kp[1], ki[1], kd[1],
DIRECT);
PID myPID2(&input[2], &output[2], &setpoint[2], kp[2], ki[2], kd[2],
DIRECT);
PID myPID3(&input[3], &output[3], &setpoint[3], kp[3], ki[3], kd[3],
DIRECT);
PID myPID4(&input[4], &output[4], &setpoint[4], kp[4], ki[4], kd[4],
DIRECT);

//Encoder controllers
Encoder myEnc0(encodPinA1[0], encodPinB1[0]);
Encoder myEnc1(encodPinA1[1], encodPinB1[1]);
Encoder myEnc2(encodPinA1[2], encodPinB1[2]);
Encoder myEnc3(encodPinA1[3], encodPinB1[3]);
Encoder myEnc4(encodPinA1[4], encodPinB1[4]);

void pwmOut(int out)
{ // to H-Bridge board
  if (out > 0)
  {
    digitalWrite(M1[motor], HIGH); // drive motor CW
    digitalWrite(M2[motor], LOW);
    analogWrite(pwm1[motor], out);
  }
  else
  {

```

```
digitalWrite(M1[motor], LOW);
digitalWrite(M2[motor], HIGH); // drive motor CCW
analogWrite(pwm1[motor], abs(out));

}
}

void motorOff()
{
  ////Serial.print("Motor Off: ");
  ////Serial.println(motor);
  digitalWrite(M1[motor], LOW);
  digitalWrite(M2[motor], LOW);
  analogWrite(pwm1[motor], 0);
}

void goToStart()
{
  int aux = 0;
  while (!digitalRead(startPoint[motor]) && (aux < 18000))
  {
    aux++;
    pwmOut(-120);
  }
  motorOff();
  output[motor] = 1;
  ////Serial.print("Reset encoder ");
  ////Serial.print(motor);
  ////Serial.print(" Input status: ");
  switch (motor)
  {
    case 0:
      myEnc0.write(0);
      ////Serial.println(myEnc0.read());
      break;
    case 1:
      myEnc1.write(0);
      ////Serial.println(myEnc1.read());
      break;
    case 2:
      myEnc2.write(0);
      ////Serial.println(myEnc2.read());
      break;
    case 3:
      myEnc3.write(0);
      ////Serial.println(myEnc3.read());
      break;
    case 4:
      myEnc4.write(0);
```



```

        // //Serial.println(myEnc4.read());
        break;
    }
}

void setup()
{
    //start and end sensors setup:
    for (motor = 0; motor <= 4; motor++)
    {
        pinMode(startPoint[motor], INPUT_PULLUP);
        pinMode(endPoint[motor], INPUT_PULLUP);
    }

    //PID controllers setup:
    myPID0.SetMode(AUTOMATIC);
    myPID0.SetSampleTime(1);
    myPID0.SetOutputLimits(-255, 255);

    myPID1.SetMode(AUTOMATIC);
    myPID1.SetSampleTime(1);
    myPID1.SetOutputLimits(-255, 255);

    myPID2.SetMode(AUTOMATIC);
    myPID2.SetSampleTime(1);
    myPID2.SetOutputLimits(-255, 255);

    myPID3.SetMode(AUTOMATIC);
    myPID3.SetSampleTime(1);
    myPID3.SetOutputLimits(-255, 255);

    myPID4.SetMode(AUTOMATIC);
    myPID4.SetSampleTime(1);
    myPID4.SetOutputLimits(-255, 255);

    motor = 0;

    TCCR1B = TCCR1B & 0b11111000 | 1; // set 31KHz PWM to reduce motor
noise

    // Serial Communication Setup:
    Serial.begin(9600);
    ////Serial.println();
    ////Serial.println("-----
");
    ////Serial.println("Start!");

```

```

//CHECKING START AND END
/*
//Serial.println("-----
");
//Serial.println("Checking Startpoints and Endpoints...");

bool _startPoint[] = {false, false, false, false, false};
bool _endPoint[] = {false, false, false, false, false};
int testPass = 0;

while (testPass < 10)
{
for (int i = 0; i < 5; i++)
{
if ((digitalRead(startPoint[i]) == 1) && (_startPoint[i] == false))
{
_startPoint[i] = true;
//Serial.print("Start Point of motor ");
//Serial.print(i);
//Serial.println(" is OK.");
testPass++;
}
if ((digitalRead(endPoint[i]) == 1) && (_endPoint[i] == false))
{
_endPoint[i] = true;
//Serial.print("End Point of motor ");
//Serial.print(i);
//Serial.println(" is OK.");
testPass++;
}
}
}
delay(200);
}

//Serial.println("All starts and end sensors are OK.");

//CHECKING ENCODERS
//Serial.println("-----
");
//Serial.println("Checking encoders");

bool _input[] = {false, false, false, false, false};
testPass = 0;

while (testPass < 5)
{
for (motor = 0; motor <= 4; motor++)
{
switch (motor)

```

```
{
case 0:
  input[motor] = myEnc0.read();
  if ((_input[motor] == false) && (input[motor] >= 50))
  {
    _input[motor] = true;
    //Serial.print("Encoder ");
    //Serial.print(motor);
    //Serial.println(" is OK.");
    testPass++;
  }
  break;

case 1:
  input[motor] = myEnc1.read();
  if ((_input[motor] == false) && (input[motor] >= 50))
  {
    _input[motor] = true;
    //Serial.print("Encoder ");
    //Serial.print(motor);
    //Serial.println(" is OK.");
    testPass++;
  }
  break;

case 2:
  input[motor] = myEnc2.read();
  if ((_input[motor] == false) && (abs(input[motor]) >= 50))
  {
    _input[motor] = true;
    //Serial.print("Encoder ");
    //Serial.print(motor);
    //Serial.println(" is OK.");
    testPass++;
  }
  break;

case 3:
  input[motor] = myEnc3.read();
  if ((_input[motor] == false) && (abs(input[motor]) >= 50))
  {
    _input[motor] = true;
    //Serial.print("Encoder ");
    //Serial.print(motor);
    //Serial.println(" is OK.");
    testPass++;
  }
  break;
```

```

case 4:
  input[motor] = myEnc4.read();
  if ((_input[motor] == false) && (abs(input[motor]) >= 50))
  {
    _input[motor] = true;
    //Serial.print("Encoder ");
    //Serial.print(motor);
    //Serial.println(" is OK.");
    testPass++;
  }
  break;
}
delay(200);
}
}

//CHECKING MOTORS

////Serial.println("Checking Motors...");
*/
for (motor = 0; motor <= 4; motor++)
{
  ////Serial.print("Testing motor ");
  ////Serial.println(motor);
  if (motor == 2 && motor == 4) {
    goToStart();
  }
  input[motor] = 0;
  /*
  switch (motor)
  {
  case 0:
    while (input[motor] < 50)
    {
      if (digitalRead(endPoint[motor])) {
        break;
      }
      pwmOut(180);
      input[motor] = myEnc0.read();
      delay(10);
    }
    motorOff();
    //Serial.print("Motor ");
    //Serial.print(motor);
    //Serial.println(" is OK.");
    break;

```

case 1:

```
while (input[motor] < 30)
{
  if (digitalRead(endPoint[motor])) {
    break;
  }
  pwmOut(180);
  input[motor] = myEnc1.read();
  delay(10);
}
motorOff();
//Serial.print("Motor ");
//Serial.print(motor);
//Serial.println(" is OK.");
break;

case 2:
while (input[motor] < 50)
{
  if (digitalRead(endPoint[motor])) {
    break;
  }
  pwmOut(120);
  input[motor] = myEnc2.read();
  delay(10);
}
motorOff();
////Serial.print("Motor ");
////Serial.print(motor);
////Serial.println(" is OK.");
break;

case 3:
while (input[motor] < 50)
{
  if (digitalRead(endPoint[motor])) {
    break;
  }
  pwmOut(120);
  input[motor] = myEnc3.read();
  delay(10);
}
motorOff();
////Serial.print("Motor ");
////Serial.print(motor);
////Serial.println(" is OK.");
break;

case 4:
while ((input[motor] < 50))
```

```

    {
        if (digitalRead(endPoint[motor])) {
            break;
        }
        pwmOut(120);
        input[motor] = myEnc4.read();
        delay(10);
    }
    motorOff();
    ///Serial.print("Motor ");
    ///Serial.print(motor);
    ///Serial.println(" is OK.");
    break;
}
delay(200);
*/
}

///Serial.print("Check-up Complete!");
///Serial.println("Entering Main operation mode.");
///Serial.println("Updating motor 3 position...");

prevInput = input[3];

}

void loop()
{
    input[3] = myEnc3.read();
    timer = millis();

    if (timer - timer0 > interval) {

        if (digitalRead(endPoint[motorKiller])) {
            //Serial.println("Motor Killed!");
            motor = motorKiller;
            motorOff();
        }
        if (digitalRead(startPoint[3])) {
            ///Serial.println("START HERE!!");
            currentPosition = 0;
        }
        if (digitalRead(endPoint[3])) {
            ///Serial.println("END HERE!!");
            currentPosition = 2;
        }
        timer0 = millis();
    }
}

```

```
if (firstDiff == 1) {
    diff = 0;
    prevInput = input[3];
    firstDiff = 0;
}
else {
    diff = input[3] - prevInput;
}

if ( diff < -EncThres) {
    currentPosition--;
    if (currentPosition < 0) {
        currentPosition = 0;
    }
}

if (diff > EncThres) {
    currentPosition++;
    if (currentPosition > 2) {
        currentPosition = 2;
    }
}

if (currentPosition == 0) {
    Serial.print("d");
    Serial.flush();
}

if (currentPosition == 1) {
    Serial.print("c");
    Serial.flush();
}

if (currentPosition == 2) {
    Serial.print("b");
    Serial.flush();
}

}
}
if (Serial.available() > 0) {
    Pos = Serial.read();
    Serial.flush();
    switch (Pos) {
        case 'f':
            //Serialln("Comand Recieved: Neutral Position.");
            motorKiller = 3;
            break;
    }
}
```

```

    case 'g':
        //Serial.println("Comand Recieved: Roll Left.");
        motorKiller = 0;
        break;
    case 'h':
        //Serial.println("Comand Recieved: Roll right.");
        motorKiller = 1;
        break;
    case 'i':
        ///Serial.println("Comand Recieved: Yaw Left.");
        motorKiller = 2;
        break;
    case 'j':
        ///Serial.println("Comand Recieved: Yaw right.");
        motorKiller = 4;
        break;
    }
}

switch (Pos)
{
    case 'f': //Neutral Position

        for (int i = 0; i <= 4; i++)
        {
            motor = i;
            setpoint[i] = 0;
            if (i != 3)
            {
                goToStart();
                motorOff();
            }
            else {
                motorOff();
            }
        }

        Pos = NULL;
        break;
    case 'g': //Roll Left
        motor = 1;
        setpoint[1] = 120;
        //Update Motor 1
        input[motor] = myEnc1.read();
        myPID1.Compute();
        //          Serial.print("motor: ");
        //          Serial.print(motor);
        //          Serial.print("; encoder: ");

```



```

//          Serial.print(input[motor]);
//          Serial.print("; setpoint: ");
//          Serial.print(setpoint[motor]);
//          Serial.print("; output: ");
//          Serial.print(output[motor]);
if ((output[motor] <= 0 && !digitalRead(startPoint[motor])) ||
(output[motor] >= 0 && !digitalRead(endPoint[motor])))
{
//          Serial.println("; Status: On");
pwmOut(output[motor]);
}
else
{
motorOff();
//          Serial.println("; Status: Off");
}
break;
case 'h': //Roll Right
setpoint[0] = 150;

//Update Motor 0
motor = 0;
input[motor] = myEnc0.read();
myPID0.Compute();
Serial.print("motor: ");
Serial.print(motor);
Serial.print("; encoder: ");
Serial.print(input[motor]);
Serial.print("; setpoint: ");
Serial.print(setpoint[motor]);
Serial.print("; output: ");
Serial.print(output[motor]);
if ((output[motor] <= 0 && !digitalRead(startPoint[motor])) ||
(output[motor] >= 0 && !digitalRead(endPoint[motor])))
{
Serial.println("; Status: On");
pwmOut(output[motor]);
}
else
{
motorOff();
Serial.println("; Status: Off");
}
break;
case 'i': //Yaw Left

setpoint[2] = 150;
setpoint[4] = 0;
//Update Motor 2

```

```

motor = 2;
input[motor] = myEnc2.read();
myPID2.Compute();
////Serial.print("motor: ");
////Serial.print(motor);
////Serial.print("; encoder: ");
////Serial.print(input[motor]);
////Serial.print("; setpoint: ");
////Serial.print(setpoint[motor]);
////Serial.print("; output: ");
////Serial.print(output[motor]);
if ((output[motor] < 0 && !digitalRead(startPoint[motor])) ||
(output[motor] > 0 && !digitalRead(endPoint[motor])))
{
  ////Serial.println("; Status: On");
  pwmOut(output[motor]);
}
else
{
  motorOff();
  ////Serial.println("; Status: Off");
}
//Update Motor 4
motor = 4;
input[motor] = myEnc4.read();
myPID4.Compute();
////Serial.print("motor: ");
////Serial.print(motor);
////Serial.print("; encoder: ");
////Serial.print(input[motor]);
////Serial.print("; setpoint: ");
////Serial.print(setpoint[motor]);
////Serial.print("; output: ");
////Serial.print(output[motor]);
if ((output[motor] < 0 && !digitalRead(startPoint[motor])) ||
(output[motor] > 0 && !digitalRead(endPoint[motor])))
{
  ////Serial.println("; Status: On");
  pwmOut(output[motor]);
}
else
{
  motorOff();
  ////Serial.println("; Status: Off");
}
break;
case 'j': //Yaw Right

```

```

setpoint[4] = 150;
setpoint[2] = 0;

//Update Motor 2
motor = 2;
input[motor] = myEnc2.read();
myPID2.Compute();
////Serial.print("motor: ");
////Serial.print(motor);
////Serial.print("; encoder: ");
////Serial.print(input[motor]);
////Serial.print("; setpoint: ");
////Serial.print(setpoint[motor]);
////Serial.print("; output: ");
////Serial.print(output[motor]);
if ((output[motor] < 0 && !digitalRead(startPoint[motor])) ||
(output[motor] > 0 && !digitalRead(endPoint[motor])))
{
    ////Serial.println("; Status: On");
    pwmOut(output[motor]);
}
else
{
    motorOff();
    ////Serial.println("; Status: Off");
}
//Update Motor 4
motor = 4;
input[motor] = myEnc4.read();
myPID4.Compute();
////Serial.print("motor: ");
////Serial.print(motor);
////Serial.print("; encoder: ");
////Serial.print(input[motor]);
////Serial.print("; setpoint: ");
////Serial.print(setpoint[motor]);
////Serial.print("; output: ");
////Serial.print(output[motor]);
if ((output[motor] < 0 && !digitalRead(startPoint[motor])) ||
(output[motor] > 0 && !digitalRead(endPoint[motor])))
{
    ////Serial.println("; Status: On");
    pwmOut(output[motor]);
}
else
{
    motorOff();
    ////Serial.println("; Status: Off");
}

```

```
        break;
    }

    if (timer - timer2 > interval2) {
        timer2 = millis();
        prevInput = input[3];
    }
}
```

Scripts utilizados para desenvolvimento do jogo sério

Câmera seguidora do personagem

```
using UnityEngine;

public class CameraFollow : MonoBehaviour
{
    public Transform player;
    public Vector3 offset;
    public Vector3 finalPosition;
    public float finalRotation = -60.0f;
    public float tiltLimit = 15.0f;
    private float delta = 0.01f;
    private float smooth = 1.0f;
    public GameObject GameController;

    public int tilt = 0;
    public float waitTime = 40.0f;
    private float timer = 0.0f;
    private float yaw_displace = 2.0f;
    private Quaternion target;

    private void Start()
    {
        target = Quaternion.Euler(transform.eulerAngles.x, 0, 0);
    }

    void Update()
    {
        if (GameController.GetComponent<RoomManager>().GameStatus)
        {
            transform.rotation = Quaternion.Slerp(transform.rotation,
target, Time.deltaTime * smooth);
            transform.position = player.position + offset;
            timer += Time.deltaTime;

            if (timer > waitTime)
            {
                timer -= waitTime;
                waitTime = Random.Range(waitTime * 0.75f, waitTime *
1.25f);

                if (tilt == 0)
                {
                    tilt = Random.Range(1, 5);
                    GetComponent<AudioSource>().Play();
                }
            }
        }
    }
}
```

```

        else
        {
            tilt = 0;
        }
        //Debug.Log("Tilt atual: ");
        // Debug.Log(tilt);
    }

    switch (tilt)
    {
        case 0:

            if (offset.x < 0)
            {
                offset.x += delta / 10;
            }

            if (offset.x > 0)
            {
                offset.x -= delta / 10;
            }
            target = Quaternion.Euler(transform.eulerAngles.x, 0,
0);
            break;

        case 1:

            target = Quaternion.Euler(transform.eulerAngles.x, 0,
tiltLimit);
            break;
        case 2:
            target = Quaternion.Euler(transform.eulerAngles.x, 0,
-tiltLimit);

            break;

        case 3:
            if (offset.x < yaw_displace)
            {
                offset.x += delta / 10;
            }

            target = Quaternion.Euler(transform.eulerAngles.x, -
tiltLimit, 0);
            break;

        case 4:
            if (offset.x > -yaw_displace)
            {

```

```

        offset.x -= delta / 10;
    }
    target = Quaternion.Euler(transform.eulerAngles.x,
tiltLimit, 0);
    break;
}

} else
{
    transform.rotation = Quaternion.Slerp(transform.rotation,
Quaternion.Euler(transform.rotation.x, finalRotation,
transform.rotation.z), Time.deltaTime * smooth);
    finalPosition.z = player.position.z;
    transform.position = finalPosition;
    tilt = 0;
}
}
}
}

```

Gerenciador de animações do personagem principal

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PlayerAnim : MonoBehaviour
{
    public Animator anim;
    public GameObject GameController;

    private float animationParameter = 0;

    // Update is called once per frame
    void Update()
    {
        if (!GameController.GetComponent<RoomManager>().GameStatus)
        {
            anim.SetBool("EndGame", true);
        }

        if(gameObject.GetComponent<PlayerMovement>().horizontalAnim ==-1)
        {
            if (animationParameter > -1)
            {
                animationParameter -= 0.1f;
            }
        }
    }
}

```

```

1)    if (gameObject.GetComponent<PlayerMovement>().horizontalAnim ==
    {
        if (animationParameter < 1)
        {
            animationParameter += 0.1f;
        }
    }

    anim.SetFloat("Horizontal", animationParameter);
}
}

```

Gerenciador de colisões do personagem principal

```

using UnityEngine;

public class PlayerCollision : MonoBehaviour
{
    public AudioSource[] Audios;
    public Transform sparkle;
    public GameObject BonusParticles;
    public GameObject ScoreText;
    public int BonusValue = 20;
    public int CollisionHelper = 5;
    public int HelperScoreLimit= 20;

    private int collisionCount;
    private int helperStatus;
    private int helperScore;

    private void OnCollisionEnter(Collision collisionInfo)
    {
        if(collisionInfo.collider.tag == "Obstacle")
        {
            sparkle.transform.position = new
            Vector3(collisionInfo.collider.transform.position.x,
            sparkle.transform.position.y, sparkle.transform.position.z);
            Audios[0].Play();
            sparkle.GetComponent<ParticleSystem>().Play();
            ScoreText.GetComponent<Score>().multiplier = 1;
            helperScore = ScoreText.GetComponent<Score>().scoreValue;

            if (Mathf.Abs(helperScore -
            ScoreText.GetComponent<Score>().scoreValue) <= HelperScoreLimit)
            {

```



```
        if (collisionCount >= CollisionHelper)
        {
            helperStatus = 1;
            collisionCount = 0;
        }
        else
        {
            collisionCount++;
        }
    }
    else
    {
        collisionCount = 1;
    }
}

if (collisionInfo.collider.tag == "Bonus")
{
    BonusParticles.GetComponent<ParticleSystem>().Play();
    Destroy(collisionInfo.gameObject);
    ScoreText.GetComponent<Score>().scoreValue += BonusValue *
ScoreText.GetComponent<Score>().multiplier;
    Audios[1].Play();
}

}

private void OnTriggerEnter(Collider other)
{
    if (helperStatus == 1)
    {
        SetHelper(other);
        helperStatus = 0;
    }
}

void SetHelper(Collider other)
{
    if (other.tag == "tileLeft")
    {
        Debug.Log("Indo pra esquerda!");
    }

    if (other.tag == "tileMiddle")
    {
        Debug.Log("Indo pro meio!");
    }
}
```

```

    }

    if (other.tag == "tileRight")
    {
        Debug.Log("Indo pra direita!");
    }
}
}

```

Gerenciador da movimentação do personagem principal

```

using UnityEngine;
using System.IO.Ports;

public class PlayerMovement : MonoBehaviour
{
    // This is a reference to the Rigidbody component called "rb"
    public Rigidbody rb;
    public GameObject GameController;
    public GameObject Camera;

    float playerX = 0.0f;
    float playerY = 0.558f;
    float playerZ = 0.0f;
    public float deltaZ = 0.05f;
    public float deltaX = 0.02f;
    public bool Status;
    public float horizontalAnim = 0;

    private int tilt = 0;
    private int lastTilt = 0;
    private int sensor = 99;
    private int arduino = 0;

    private SerialPort sp = new SerialPort("COM3", 9600);

    public SerialPort Sp
    {
        get
        {
            return sp;
        }
        set
        {

```

```

        sp = value;
    }
}

private void Start()
{
    Sp.Open();
    Sp.ReadTimeout = 1;

    if (Sp.IsOpen)
    {
        Debug.Log("serial aberta!");
        //if (sp.BytesToRead > 0)
        //{
        //    handShake = sp.ReadChar();
        //    Debug.Log("Char Lido!");
        //    if (handShake == 97) {
        //        Debug.Log("Char Escrito!");
        //        sp.WriteLine("a");
        //    }
        //}
    }
}

// We marked this as "Fixed"Update because we
// are using it to mess with physics.
void FixedUpdate()
{
    Status = GameController.GetComponent<RoomManager>().GameStatus;
    tilt = Camera.GetComponent<CameraFollow>().tilt;
    playerX = rb.position.x;
    playerY = rb.position.y;
    playerZ = rb.position.z;

    if (Sp.IsOpen)
    {
        if (Sp.BytesToRead > 0)
        {
            sensor = Sp.ReadChar();
        }
        arduino = ArduinoMove(sensor);

        if (tilt != lastTilt)
        {
            lastTilt = tilt;
            switch(tilt)
            {

```

```
        case 0:
            Sp.WriteLine("f");
            break;

        case 1:
            Sp.WriteLine("g");
            break;

        case 2:
            Sp.WriteLine("h");
            break;

        case 3:
            Sp.WriteLine("i");
            break;

        case 4:
            Sp.WriteLine("j");
            break;
    }
}

if (Status == true)
{

    // Add a forward force
    rb.transform.position = new Vector3(playerX, playerY, playerZ
+ deltaZ);

    if (Input.GetKey(KeyCode.D) || arduino==2) // If the player
is pressing the "d" key
    {
        // Add a force to the right
        rb.transform.position = new Vector3(playerX + deltaX,
playerY, playerZ + deltaZ);
    }

    if (Input.GetKey(KeyCode.A)|| arduino ==1) // If the player
is pressing the "a" key
    {
        // Add a force to the left
        rb.transform.position = new Vector3(playerX - deltaX,
playerY, playerZ + deltaZ);
    }
}
```

```
        if (rb.position.y < 0.556f)
        {
            rb.transform.position = new Vector3(playerX, 0.558f,
playerZ);
        }

        if (rb.rotation.y != 0)
        {
            rb.transform.eulerAngles = new Vector3(0, 0, 0); ;
        }
    } else
    {
        rb.transform.position = new Vector3(playerX, playerY,
playerZ);
    }
}

private int ArduinoMove(int position)
{
    switch (position)
    {
        case 97:
            if (rb.transform.position.x < -5.05f)
            {
                horizontalAnim = 1;
                return 2;
            }
            if (rb.transform.position.x >= -5.0f)
            {
                horizontalAnim = -1;
                return 1;
            }
            break;

        case 98:
            if (rb.transform.position.x <-3.05f)
            {
                horizontalAnim = 1;
                return 2;
            }
            if (rb.transform.position.x > -2.95f)
            {
                horizontalAnim = -1;
                return 1;
            }
            break;
        case 99:
```

```

        if (rb.transform.position.x < -0.05f)
        {
            horizontalAnim = 1;
            return 2;
        }
        if (rb.transform.position.x > 0.05f)
        {
            horizontalAnim = -1;
            return 1;
        }
        break;

    case 100:
        if (rb.transform.position.x < 2.95f)
        {
            horizontalAnim = 1;
            return 2;
        }
        if (rb.transform.position.x > 3.05f)
        {
            horizontalAnim = -1;
            return 1;
        }
        break;

    case 101:
        if (rb.transform.position.x < 5.0f)
        {
            horizontalAnim = 1;
            return 2;
        }
        if (rb.transform.position.x > 5.05f)
        {
            horizontalAnim = -1;
            return 1;
        }
        break;
    }
    horizontalAnim = 0;
    return 0;
}
}
}

```

Gerenciador de posição do portal

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

```

```

public class PortalOffset : MonoBehaviour
{
    public Transform player;
    public Vector3 offset;

    void Update()
    {
        transform.position = new Vector3 (offset.x, offset.y,
player.position.z + offset.z);
    }
}

```

Controle de objetos do ambiente de jogo

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class RoomManager : MonoBehaviour
{
    public bool GameStatus;
    static public int SessionTime = 10;
    public GameObject SessionEndText;
    public GameObject restartButton;
    public GameObject exitButton;
    public GameObject stopButton;

    private void Start()
    {
        GameStatus = true;
        SessionEndText.SetActive(false);
        restartButton.SetActive(false);
        exitButton.SetActive(false);
    }
    public void StartGame()
    {
        GameStatus = true;
    }

    public void StopGame()
    {
        GameStatus = false;
        gameObject.GetComponent<AudioSource>().Play();
    }
}

```

```

        SessionEndText.SetActive(true);
        restartButton.SetActive(true);
        exitButton.SetActive(true);
        stopButton.SetActive(false);
    }
}

```

Controle de pontuação do jogo

```

using UnityEngine;
using UnityEngine.UI;

public class Score : MonoBehaviour
{
    public GameObject Camera;
    public GameObject GameController;
    public Text scoreText;
    public GameObject MultiplierText;
    public int multiplier = 1;
    public int scoreValue = 0;

    private float theTime = 0.0f;
    private int tilt = 0;
    private int lastTilt = 0;

    void Start()
    {
        tilt = Camera.GetComponent<CameraFollow>().tilt;
        lastTilt = tilt;
        MultiplierText.SetActive(false);
    }

    void Update()
    {
        if (GameController.GetComponent<RoomManager>().GameStatus)
        {
            theTime += Time.deltaTime;
        }

        if (theTime >= 1)
        {
            scoreValue += (int)theTime * multiplier;
            theTime -= (int)theTime;
            scoreText.text = scoreValue.ToString("0");
            tilt = Camera.GetComponent<CameraFollow>().tilt;
        }
    }
}

```



```

    }

    if (tilt != 0 && lastTilt != tilt)
    {
        if (multiplier < 4)
        {
            multiplier += 1;
        }
        lastTilt = tilt;
    }

    if (multiplier > 1)
    {
        MultiplierText.SetActive(true);
    } else
    {
        MultiplierText.SetActive(false);
    }
}
}

```

Gerenciador de *Tiles*

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class TileManager : MonoBehaviour
{
    public GameObject[] tilePrefabs;
    public GameObject bonusCube;
    public int tilesOnScreen = 8;
    public int bonuslimit = 2;

    private Transform playerTransform;
    private float spawnZ = 0f;
    private float tileLength = 50f;
    private float bonusCubeY = 2.558f;
    private List<GameObject> activeTiles;
    private float safeZone = 85f;
    private int lastPrefabIndex = 0;
    private int bonusCounter = 0;
}

```

```

void Start()
{
    activeTiles = new List<GameObject>();
    playerTransform =
GameObject.FindGameObjectWithTag("Player").transform;
    for(int i = 0; i < tilesOnScreen-1; i++)
    {
        SpawnTile();
    }

}

void Update()
{
    if(playerTransform.position.z - safeZone > (spawnZ -
tilesOnScreen * tileLength))
    {
        SpawnTile();
        DeleteTile();
        bonusCounter += 1;
        if (bonusCounter >= bonuslimit)
        {
            bonusCounter = 0;
            SpawnBonus();
        }
    }

}

void SpawnTile(int prefabIndex = -1)
{
    GameObject go;
    go = Instantiate(tilePrefabs[RandomPrefabIndex()]) as GameObject;
    go.transform.SetParent(transform);
    go.transform.position = Vector3.forward * spawnZ;
    spawnZ += tileLength;
    activeTiles.Add(go);
}

void SpawnBonus(int prefabIndex = -1)
{
    GameObject cube;
    int cubePos = Random.Range(0, 3);
    cube = Instantiate(bonusCube) as GameObject;

    switch (cubePos)
    {
        case 0:

```

```

        cube.transform.position = new Vector3(-3.0f, bonusCubeY,
spawnZ);
        break;
    case 1:
        cube.transform.position = new Vector3(0.0f, bonusCubeY,
spawnZ);
        break;
    case 2:
        cube.transform.position = new Vector3(3.0f, bonusCubeY,
spawnZ);
        break;
    }
}

void DeleteTile()
{
    Destroy(activeTiles[0]);
    activeTiles.RemoveAt(0);
}

int RandomPrefabIndex()
{
    if (tilePrefabs.Length <= 1)
        return 0;

    int randomIndex = lastPrefabIndex;
    while (randomIndex == lastPrefabIndex)
    {
        randomIndex = Random.Range(0, tilePrefabs.Length);
    }

    lastPrefabIndex = randomIndex;
    return randomIndex;
}
}

```

Gerenciador de rotações da câmera

```

using UnityEngine;
using UnityEngine.UI;

public class TimeCounter : MonoBehaviour
{
    public GameObject GameController;
    public Text text;
    float theTime = 0f;
}

```

```

public int speed = 1;
public static int StartTime = 10;
int secs = 60;

private void Start()
{
    if (SetSessionTime.SessionTime != 0)
    {
        StartTime = SetSessionTime.SessionTime;
    }
}

void Update()
{
    //string hours = Mathf.Floor((theTime % 216000) /
3600).ToString("00");
    string minutes = StartTime.ToString("00");
    string seconds = (60 - secs).ToString("00");

    if (GameController.GetComponent<RoomManager>().GameStatus)
    {
        theTime += Time.deltaTime * speed;
        if (Mathf.Floor(theTime % 60) >= 1)
        {
            secs++;
            theTime -= theTime;
        }
    }

    if (secs>60)
    {
        if (StartTime !=0)
        {
            StartTime--;
            secs = 1;
        } else
        {
            secs = 60;
            GameController.GetComponent<RoomManager>().GameStatus =
false;
        }
    }

    text.text = minutes + ":" + seconds;
}
}

```

Projeto de Pesquisa para testes em seres humanos aprovado no CEP

Escopo Aprovado para testes piloto

Participarão do estudo pessoas saudáveis e com sintomas clínicos de AVE isquêmico divididas em dois grupos: Grupo de Pessoas Saudáveis (GPS) e Grupo de Pessoas com AVE isquêmico (GPAI). O GPS será composto por indivíduos com idade superior a 18 anos, sem histórico de doenças neurológicas, de ambos os sexos, e o GPAI por indivíduos com idade superior a 18 anos, que tenham sofrido AVE isquêmico de forma a afetar a sua mobilidade, de ambos os sexos. Todos os participantes assinarão o termo de consentimento livre e esclarecido, que será aplicado no primeiro encontro, pelo pesquisador principal, após serem explicados todos os procedimentos.

O número de participantes da pesquisa foi selecionado com base em estatística de proporção para uma incidência de 5%, assumindo que se tenha 99% de sucessos (n_s) nos testes da pesquisa, resultando 1% de falhas (n_f) (TRIOLA, 1999). Portanto, para a probabilidade de 99% e um erro percentual de 5% a amostra necessária deverá ser de 25 pessoas, sendo 15 para o GPS e 10 para o GPAI.

Os participantes da pesquisa do GPS serão recrutados entre os colaboradores e profissionais da saúde da Clínica do Curso de Fisioterapia da UFU. Os participantes da pesquisa do GPAI serão recrutados na Clínica do Curso de Fisioterapia da UFU. O convite para participação da pesquisa será feito no local, logo após a realização das atividades de forma a não interferir no tratamento ou trabalho de nenhum dos participantes. Dessa forma, não será necessário nenhum deslocamento dos participantes para realização da pesquisa.

Deve se ressaltar que o presente estudo é de caráter inicial, onde deseja-se confirmar a aplicabilidade do dispositivo descrito nas terapias de reabilitação das articulações do corpo humano, em especial em pacientes pós-AVE. Futuramente, novas pesquisas deverão ser submetidas a fim de avaliar um público maior, o que faz essa pesquisa inicial essencial.

Procedimentos para a coleta de dados

Os procedimentos de coleta de dados serão realizados na Clínica do Curso de Fisioterapia/UFU. Os indivíduos serão instruídos e os termos de consentimento livre e estabelecido assinados.

A avaliação do funcionamento do mecanismo será realizada coletando os dados gerados pelos sensores na estrutura. Esta coleta será feita em tempo real com o auxílio de um microcontrolador, que enviará os dados a um computador PC por meio de uma interface USB. O microcontrolador também enviará comandos para um jogo no computador, desenvolvido para ser utilizado em conjunto com a estrutura e é controlado em função da movimentação desta.

Os participantes da pesquisa serão instruídos a posicionar-se na plataforma do suporte corporal, vestindo o cinto de segurança e apoiando-se sobre o selim acoplado. Todos os ajustes necessários para o conforto dos participantes serão executados. Os participantes que necessitarem de auxílio para serem posicionados no suporte receberão ajuda da equipe de pesquisa. Após a estrutura ser ajustada, os atuadores da estrutura serão calibrados e a esteira acionada. O participante, então, irá iniciar o jogo, caminhando sobre a estrutura e seguindo as instruções de movimentação apresentadas durante o jogo. As sessões terão duração de aproximadamente 10 minutos, onde os dados serão coletados ininterruptamente. As sessões de movimentação serão acompanhadas pela profissional da saúde Profa Dra Jadiane Dionisio, fisioterapeuta.

As sessões de testes serão realizadas semanalmente ao longo de três meses. No início das sessões, os participantes do GPAI terão as características de suas lesões registradas, tais como a Escala de Ashworth Modificada, lado parético, hemisfério dominante etc. Ao término das sessões, os participantes da pesquisa deverão responder um questionário contendo sete perguntas sobre a experiência. Este questionário é baseado no Game Experience Questionnaire (GEQ), utilizado para a avaliação de jogos eletrônicos com critérios objetivos e bem estabelecidos (IJSELSTEIJN et al., 2008; NACKE et al., 2009).

O tempo total para cada sessão, incluindo as explicações necessárias, montagem, familiarização com a estrutura de reabilitação, sessão e desmontagem será de aproximadamente 30 minutos.

Critérios de Inclusão

O GPS será composto por pessoas de ambos os sexos maiores de 18 anos que já estejam na Clínica de Fisioterapia da UFU durante o período de realização da pesquisa e que não possuam nenhum tipo de lesão que afete a execução da marcha humana normal.

O GPAI será composto por Pessoas de ambos os sexos maiores de 18 anos, que tenham sofrido AVE e estejam em processo de tratamento oferecido pela Clínica de Fisioterapia da UFU durante o período de realização da pesquisa na UFU, com espasticidade em um dos membros inferiores diagnosticada pela Escala de Ashworth Modificada entre 1 e 2.

Critérios de Exclusão

Serão excluídos dos testes aquelas pessoas que possuam qualquer tipo de lesões ou doenças respiratórias, cardiovasculares, articulares ou outras condições que possam ser agravadas ou causar dores durante a execução dos exercícios em esteira durante a pesquisa.

Também serão excluídas aquelas que apresentarem acometimentos visuais, auditivos ou cognitivos graves.

Nenhuma das etapas da avaliação oferece risco grave à sua saúde, tampouco o expõe a situações constrangedoras. Poderá ocorrer possível dor muscular decorrente de atividade física no

equipamento para reabilitação, porém ambos prevalentes em curto período. Caso necessário, pode-se diminuir o número de repetições ou a amplitude do movimento e até mesmo suspender a pesquisa.

A identidade dos participantes será preservada em todas as etapas do estudo e o risco de identificação é mínimo. Somente os pesquisadores que participarão das coletas de dados terão acesso as informações referentes a identificação e estas não serão divulgadas em nenhum momento. Todos os formulários serão identificados apenas por códigos numéricos aleatórios. Os dados obtidos dos sensores e formulários dos participantes serão apresentados em artigos e resumos de congressos por meio de valores numéricos (médias e desvios-padrão), o que garantirá o sigilo.

Para evitar a identificação dos participantes, durante os testes, os dados coletados serão nomeados de maneira padrão e sistêmica, como “Dados do Participante X”, sendo X um número inteiro qualquer, além do mais, as gravações em vídeo serão feitas do pescoço para baixo, dificultando o reconhecimento dos participantes. Todos os dados da pesquisa incluindo as gravações em vídeo, ficarão registrados sob guarda e responsabilidade dos pesquisadores em arquivo digital por um período mínimo de 5 (cinco) anos após o término da pesquisa.

Como as coletas serão realizadas com o paciente devidamente acomodado na estrutura, não há risco significativo de quedas. Além disso, todos os testes experimentais serão acompanhados por profissionais da área de saúde que trabalharão no momento visto que tais testes serão realizados nas dependências da Clínica de Fisioterapia da UFU.

Não haverá nenhum gasto ou ganho financeiro referente a participação no presente estudo.

A participação no presente estudo irá contribuir para a melhoria na qualidade de vida para pacientes e profissionais da saúde ligados nas atividades de reabilitação, bem como contribuições científicas relacionadas à modelagem de estruturas robóticas e fisioterapêuticas.

Análise dos dados

Os dados de ambas as origens serão analisados utilizando técnicas de Estatística e *Machine Learning* aplicadas à Ciência de Dados utilizando a biblioteca de análise de Dados Pandas para Python, ou o software R. As rotinas irão agrupar os dados das variações de posição em séries temporais, onde serão extraídas as medidas estatísticas básicas (média, desvio padrão, quartis, mínimo, máximo etc.). As mesmas medidas serão extraídas dos dados de participação e pontuação atingida pelos pacientes.

Posteriormente, será elaborada uma rotina para traçar as correlações entre as variáveis analisadas. O objetivo será verificar se há evolução nas medidas estatísticas acerca da posição em relação aos participantes, se há correlação entre o nível de assistência aplicado pela estrutura e o

desempenho no jogo, e se há correlação entre as demais variáveis monitoradas. Todos os dados levantados também serão comparados entre o GPS e o GPAI.

A partir dos dados levantados, será realizada a análise dos resultados, os quais serão sintetizados em artigos e submetidos para publicação em periódicos científicos.