

GUILHERME NAMINDOME
MURILO CARVALHO DE OLIVEIRA

Projeto de exoesqueleto de membro superior para estudo de terapia robótica

São Paulo
2013

GUILHERME NAMINDOME
MURILO CARVALHO DE OLIVEIRA

Projeto de exoesqueleto de membro superior para estudo de terapia robótica

Trabalho de Conclusão de Curso apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Bacharel em Engenharia

Orientador(a):
Prof. Dr. Tarcisio Antonio Hess Coelho

São Paulo
2013

GUILHERME NAMINDOME
MURILO CARVALHO DE OLIVEIRA

Projeto de exoesqueleto de membro superior para estudo de terapia robótica

Trabalho de Conclusão de Curso apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Bacharel em Engenharia

Área de Concentração: Reabilitação robótica

Orientador(a):
Prof. Dr. Tarcisio Antonio Hess Coelho

São Paulo
2013

FICHA CATALOGRÁFICA

Namindome, Guilherme

Projeto de exoesqueleto de membro superior para estudo de terapia robótica / G. Namindome, M.C. de Oliveira. -- São Paulo, 2013.

118 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.

1.Robótica 2.Cotovelo 3.Fisioterapia 4.Mecanismo I.Oliveira, Murilo Carvalho de II.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos III.t.

DEDICATÓRIA

Guilherme Namindome dedica este trabalho aos seus pais Roberto e Maria Aparecida, ao seu irmão André e à sua namorada Fernanda, pelo apoio e compreensão durante esta trajetória.

Murilo Carvalho de Oliveira dedica este trabalho à sua mãe Emilça, ao seu pai José, aos seus irmãos e amigos queridos e a todos os pacientes com disfunções motoras que acreditam na tecnologia como luz à sua cura.

AGRADECIMENTOS

Agradecimento especial ao nosso orientador Prof. Dr. Tarcísio Antonio Hess Coelho, pelo suporte que nos foi dado durante o desenvolvimento deste trabalho, pela dedicação percebida ao decorrer da orientação e pelo compromisso em prover a educação e aprendizado.

Ao Marcos, Ricardo e José da Vila Zelina, pelo auxílio que nos foi prestado com toda sua experiência na construção do protótipo.

Aos colegas de graduação e demais amigos e parentes que acreditaram e contribuíram direta ou indiretamente na viabilização deste projeto.

Aos nossos professores do curso de graduação em Engenharia Mecatrônica e à Escola Politécnica da Universidade de São Paulo, por proverem o conhecimento e maturidade.

RESUMO

O número de pessoas que sofrem disfunções motoras tem se elevado devido à diferentes razões, entre elas o envelhecimento da população. Acidentes Vasculares Cerebrais (AVCs) são considerados os principais causadores de disfunções motoras na população de países desenvolvidos. O uso de dispositivos de robótica para reabilitação de membros está se tornando cada vez mais aceito por médicos e terapeutas, onde o objetivo é a redução do tempo de recuperação comparado com métodos convencionais e também o aumento do nível de recuperação dos membros afetados. Neste contexto, este trabalho tem como objetivo o desenvolvimento de uma órtese de membro superior baseada no conceito de mecanismo de cadeia passiva, para auxílio no tratamento de reabilitação. Observou-se que muitas das soluções existentes não abordam o problema da variação do eixo de rotação existente em um cotovelo real pois seus protótipos utilizam juntas rotacionais comuns, o que pode gerar problemas por excesso de esforços na região de fixação do antebraço nos pacientes. Este trabalho proporá uma maior fidelidade entre a junta real e o exoesqueleto projetado em comparação às soluções existentes, partindo da proposta de um mecanismo virtual com centro de rotação variável e da utilização destes resultados no controle do acionamento da órtese. O exoesqueleto será construído como uma cadeia seriada de duas barras que acionará o membro superior humano, modelado como cadeia passiva do mecanismo. A modelagem da cadeia passiva será feita com base no mecanismo plano de quadrilátero articulado.

Palavras-chave: biorrobótica, cotovelo, órtese, junta, reabilitação

ABSTRACT

The number of people suffering from motor dysfunction has raised due to various reasons, including the aging population. Cerebrovascular Accidents (CVA) are the main causes of motor dysfunction in the population of developed countries. The use of robotic devices for limbs rehabilitation are becoming increasingly accepted by doctors and therapists, where the benefits achieved are the reduction of the recovery time compared to conventional methods and also the increase of recovery rates on affected limbs.

In this context, this work aims at the development of an upper limb orthosis based on the concept of passive chain mechanism to aid in rehabilitation treatments. It was observed that many existing solutions do not address the problem of the variation of the rotation axis for a human elbow since the prototypes usually use common rotational joints, which can create problems by excessive forces on the attachment region of the forearm in patients. This paper will propose a higher fidelity between the human joint and the designed exoskeleton compared to existing solutions, based on the proposal of a virtual mechanism with a variable center of rotation. The exoskeleton was built as a serial chain of two aluminium bars which drives the upper limb modeled as a passive mechanism. The modeling of the articulated quadrilateral mechanism provided results necessary in the acting and control of the orthosis.

Keywords: biorobotics, elbow, orthosis, joint, rehabilitation

LISTA DE FIGURAS

Figura 1 - Articulação da junta Úmero-Rádio-Ulnar (Tashjian, Katarincic, 2006)	15
Figura 2 - Experimento para medida de forças de transmissão entre robô e paciente (Amigo et al., 2012).....	18
Figura 3 - Protótipo desenvolvido por Seki et al. (2011) utilizando-se uma junta rotacional e processamento de sinal elétrico	18
Figura 4 - Desenho em CAD do protótipo criado por Vanderniepen et al. (2008) com um pequeno atuador na região inferior	19
Figura 5 - Protótipo criado por Vanderniepen et al. (2008) que utiliza uma mola pré-tensionada	19
Figura 6 - Protótipo com a junta desenvolvida por Pylatiuk et al. (2009) retraída e expandida de 90°	20
Figura 7 - Diagrama da composição dos subsistemas	22
Figura 8 - Mecanismo de cadeia seriada com mobilidade igual a 3.....	23
Figura 9 - Cadeia passiva da junta Úmero-Rádio-Ulnar modelada como quadrilátero articulado	24
Figura 10 - Vetores para equacionamento. (a) Lado esquerdo do mecanismo. (b) Lado direito do mecanismo. Extraído de Coelho, 2009.....	26
Figura 11 - Representação dos vetores v_2 , v_3 e v_4	29
Figura 12 - Rotação dos vetores para o equacionamento.	29
Figura 13 - Experimento realizado para coleta de dados experimentais	32
Figura 14 - Representação macroscópica da órtese em Matlab.....	34
Figura 15 - Detalhe do mecanismo 4R na região das juntas mecânicas.....	35
Figura 16 - Modelo tridimensional do exoesqueleto projeto.....	38
Figura 17 - Detalhe da transmissão ortogonal de movimento por <i>worm gear</i>	39
Figura 18 - (a) Fixação do motor de passo e (b) acoplamento entre eixo do motor e eixo da caixa de redução.....	40
Figura 19 - Fixação ajustável da cadeia seriada na estrutura da órtese	41
Figura 20 - Vista frontal da órtese	42
Figura 21 - Vista lateral da órtese.....	43

Figura 22 - Peças principais do exoesqueleto: (a) caixa de redução e cadeia seriada, (b) placa de fixação dos equipamentos do "braço", (c) placa de fixação dos equipamentos do "antebraço", (d) motor de passo e (e) suporte do motor de passo.....	45
Figura 23 - Detalhe da caixa de redução e da cadeia seriada do exoesqueleto	46
Figura 24 - Vista lateral do protótipo construído e montado.....	47
Figura 25 - Subsistema mecânico.....	47
Figura 26 - (a) Driver para o motor de passo Geckodrive G251 e (b) microcontrolador ATmega328P	48
Figura 27 - Especificações técnicas do driver Geckodrive G251. Extraído do datasheet do componente.....	49
Figura 28 - Diagrama de casos de uso da interface Homem - Máquina.....	51
Figura 29 - Diagrama de casos de uso do controlador ATmega328P.....	51
Figura 30 - Diagrama de componentes da interface Homem - Máquina.....	52
Figura 31 - Diagrama de componentes do controlador ATmega328P.....	52
Figura 32 - Interface Homem Máquina exibindo a posição no modo “Manual”	53
Figura 33 - Interface Homem Máquina trabalhando em modo “Automático”	54
Figura 34- Modelo da órtese em Working Model com o quadrilátero articulado acoplado	57
Figura 35 - Relação entre o deslocamento angular do centro de massa do antebraço e o número de passos do motor	59
Figura 36 - Órtese acoplada ao membro superior.....	60
Figura 37 - Medição de deslocamento angular do dispositivo. A sequência de movimenta mostra o antebraço em 0 grau (a), 40 graus (b), 60 graus (c), 90 graus (d) e 120 graus (e).....	62

LISTA DE TABELAS

Tabela 1 - Matriz de decisão para escolha do motor	37
--	----

LISTA DE SÍMBOLOS

P_i - Pontos prescritos de localização do mecanismo

A_i - Pontos de fixação da barra W com o antebraço

B_i - Pontos de fixação da barra U com o antebraço

O_2 - Ponto fixo no braço para rotação da barra W

O_2 - Ponto fixo no braço para rotação da barra U

\vec{W}_1 - Vetor de posição da barra móvel W do lado esquerdo do mecanismo

\vec{U}_1 - Vetor de posição da barra móvel U do lado direito do mecanismo

\vec{Z}_1 - Vetor de tamanho fixo que mede a distância entre P_i e A_i

\vec{S}_1 - Vetor de tamanho fixo que mede a distância entre P_i e B_i

β_2 - Ângulo entre \vec{W}_1 e \vec{W}_2

β_3 - Ângulo entre \vec{W}_1 e \vec{W}_3

γ_2 - Ângulo entre \vec{U}_1 e \vec{U}_2

γ_3 - Ângulo entre \vec{U}_1 e \vec{U}_3

α_2 - Ângulo entre \vec{Z}_1 e \vec{Z}_2 ou ângulo entre \vec{S}_1 e \vec{S}_2

α_3 - Ângulo entre \vec{Z}_1 e \vec{Z}_3 ou ângulo entre \vec{S}_1 e \vec{S}_3

θ - Ângulo entre \vec{W}_1 e o eixo horizontal

ϕ - Ângulo entre \vec{Z}_1 e o eixo horizontal

σ - Ângulo entre \vec{U}_1 e o eixo horizontal

ψ - Ângulo entre \vec{S}_1 e o eixo horizontal

GLOSSÁRIO

CAD - *Computer Auxiliated Design*

Cadeia ativa - Mecanismo que possui um atuador e movimenta as cadeias adjacentes

Cadeia passiva - Mecanismo que é movimentado indiretamente por outra cadeia

Cadeia seriada - Mecanismo de cadeia aberta, com barras articuladas em série

Matlab - Software numérico para simulações e resoluções de problemas matemáticos

Órtese - Um apoio ou dispositivo externo aplicado ao corpo para modificar os aspectos funcionais ou estruturais do sistema neuromusculoesquelético para obtenção de alguma vantagem mecânica ou ortopédica

Open-Source - Softwares de distribuição livre, cujos códigos-fonte estão disponíveis para modificações e adaptações.

Quadrilátero Articulado - Mecanismo composto de 4 barras rígidas articuladas com juntas rotacionais simples

Rádio - Osso do antebraço que se estende anatomicamente na parte lateral do antebraço, indo do cotovelo até a lateral do punho

SolidWorks - Software de modelagem e simulação 3D

Ulna - Osso longo que compõe, junto com o Rádio, o conjunto ósseo do antebraço humano

Úmero - Osso longo que compõe a porção esquelética do braço humano

Working Model - Software de modelagem dinâmica e cinemática de mecanismos 2D

Worm Drive - Sistema de transmissão entre dois eixos perpendiculares entre si, feitos através de fuso e engrenagem

SUMÁRIO

1.	INTRODUÇÃO.....	14
1.1.	OBJETIVOS.....	16
2.	DESENVOLVIMENTO.....	17
2.1.	REVISÃO DA LITERATURA.....	17
3.	MATERIAIS E MÉTODOS.....	21
3.1.	ORGANIZAÇÃO E CRONOGRAMA.....	21
3.2.	MECANISMO DE CADEIA PASSIVA.....	22
3.3.	MODELAGEM DA JUNTA BIOLÓGICA.....	25
3.4.	COLETA DE DADOS EXPERIMENTAIS.....	31
3.5.	SIMULAÇÃO EM AMBIENTE MATLAB.....	34
3.6.	MODELAGEM EM AMBIENTE CAD.....	36
4.	RESULTADOS.....	44
4.1.	CONSTRUÇÃO DO PROTÓTIPO.....	44
4.2.	PROJETO DO SUBSISTEMA ELETRÔNICO.....	48
4.3.	PROJETO DO SUBSISTEMA DE PROGRAMAÇÃO.....	50
4.3.1.	INTERFACE HOMEM-MÁQUINA.....	53
4.3.1.	CINEMÁTICA DIRETA E CONTROLE DO MECANISMO.....	56
4.4.	TESTES FUNCIONAIS.....	60
5.	DISCUSSÃO.....	62
6.	CONCLUSÕES.....	64
7.	REFERÊNCIAS BIBLIOGRÁFICAS.....	65

APÊNDICES

APÊNDICE A - SIMULAÇÃO EM MATLAB.....	68
APÊNDICE B - DESENHOS DE CONJUNTO DO PROTÓTIPO.....	71
APÊNDICE C - CÓDIGO PARA INTERFACE HOMEM-MÁQUINA.....	79
APÊNDICE D - CÓDIGO EM C++ PARA ATMEGA.....	110

1. INTRODUÇÃO

Acidentes Vasculares Cerebrais (AVCs) são considerados os principais causadores de disfunções motoras na população de países desenvolvidos (Benjamin et al., 2012). No Brasil, estima-se que a taxa de mortalidade provocada por AVCs seja da ordem de 56 pessoas para cada grupo de 100 mil habitantes. Calcula-se, ainda, que o AVC seja responsável por cerca de 8% das internações e de 19% dos custos dos hospitais públicos brasileiros (Filho, 2006). Outras frequentes causas de distúrbios motores são a Paralisia Cerebral e Lesões da Medula Espinhal (Mancini et al., 2004).

Segundo Olaya (2009), as técnicas normalmente utilizadas para o tratamento de disfunções neuromotoras, especialmente no caso de AVCs, são baseadas em exercícios de repetição mecânica. O autor cita que tais exercícios podem ser executados ativo ou passivamente pelo paciente. Nos exercícios passivos, o terapeuta move o membro afetado do paciente, que utiliza pouca atuação motora própria. Nos exercícios ativos, o paciente inicia o movimento e, se necessário, o terapeuta o auxilia a completar o exercício. Uma vez que o progresso da reabilitação está diretamente relacionado à qualidade e intensidade dos exercícios, o uso de exoesqueletos robóticos para estes tipos de terapias trazem bons resultados para casos de recuperação de AVCs (Krebs et al., 1998, Coens, 1999).

Vanderniepen et al. (2008) citam que dentre as vantagens do uso de terapias auxiliadas roboticamente estão os graus de objetividade e repetibilidade alcançados nos tratamentos, muito mais difíceis de se alcançar com terapias manuais. Ainda, os esforços e progressos do paciente podem ser mais precisamente monitorados, provendo aos terapeutas informações detalhadas de cada paciente individualmente, fazendo com que eles possam então adequar os treinamentos para as necessidades de cada paciente em particular.

Amigo et al. (2012) citam que o uso de dispositivos de robótica para reabilitação de membros está se tornando cada vez mais aceito por médicos e terapeutas, onde o objetivo é a redução do tempo de recuperação comparado com métodos convencionais e também o aumento do nível de recuperação dos membros afetados. O autores reali-

zaram um estudo da transmissão das forças decorrentes do desalinhamento do centro de rotação da junta do cotovelo de uma órtese utilizando sensores para medição de força e torque. Eles enfatizam a importância e a dificuldade da aplicação da robótica de reabilitação mecânica para a reprodução do movimento de juntas biológicas reais. A fidelidade entre o movimento real e o simulado pela órtese se mostra um grande desafio, já que o desalinhamento do movimento e da transmissão de forças pode se mostrar maléfico para o tratamento do paciente, causando dor e fadiga muscular. De fato, a Figura 1 ilustra de maneira simplificada a anatomia da junta Úmero-Rádio-Ulnar. Observa-se que a articulação formada pelos três ossos é complexa e não se resume a uma modelagem por junta rotacional. Como será ilustrado experimentalmente em seções posteriores, a articulação possui um centro de rotação variável, devido ao movimento relativo entre os ossos, e portanto consiste em um mecanismo de complexidade maior, que é estudado neste trabalho.



Figura 1 - Articulação da junta Úmero-Rádio-Ulnar (Tashjian, Katarincic, 2006)

1.1. OBJETIVOS

Dada a necessidade da fidelização biológica das soluções robóticas para melhores resultados em tratamentos de recuperação motora, procurou-se buscar pontos em comum nos mecanismos normalmente utilizados e propor melhorias em suas soluções. Em particular, notou-se que as soluções existentes de exoesqueletos da junta de membro superior Úmero-Rádio-Ulnar se utilizam muitas vezes de uma junta rotacional com centro de rotação fixo, o que não representa o mecanismo real biológico devido ao fato de que o centro de rotação da junta real é variável. O trabalho de Amigo et al (2012) ilustra alguns efeitos que estes desalinhamentos podem provocar.

Assim, visando o desenvolvimento de novos mecanismos para tratamento de disfunções motoras que sejam capaz de reduzir esforços indesejáveis e otimizar resultados e tempos de tratamento dos pacientes, este trabalho tem como objetivo o projeto, modelagem e construção de um protótipo para exoesqueleto da junta Úmero-Rádio-Ulnar utilizando o conceito de mecanismo de cadeia passiva. Como descrito nas seções posteriores, a junta Úmero-Rádio-Ulnar pode ser melhor modelada por um mecanismo de quadrilátero articulado. Desta forma, modelando a junta biológica com os resultados a serem obtidos, é possível construir uma órtese de membro superior acionada com duas barras seriadas, tendo a junta biológica como cadeia passiva. O protótipo tem como requisito a reprodução das funcionalidades das órteses para tratamento de membros superiores comumente utilizadas, adicionado o diferencial do controle preciso do movimento através da modelagem da junta biológica por um quadrilátero articulado. Trabalhos futuros relativos à proposta poderão envolver o estudo das forças de reação e de como o sistema impacta na qualidade e eficácia no tratamento dos pacientes.

2. DESENVOLVIMENTO

Visando suportar o desenvolvimento do trabalho realizou-se um estudo sobre o Estado da Arte das soluções de órteses para tratamento de disfunções motoras existentes atualmente. Seguido a isso, foi desenvolvida uma proposta de metodologia no desenvolvimento deste trabalho. Os resultados alcançados se encontram nas próximas subseções.

2.1. REVISÃO DA LITERATURA

No que se refere às soluções de órteses robóticas para tratamento e reabilitação de membros humanos, observou-se, após a pesquisa bibliográfica deste trabalho, que existem diversas soluções de protótipos existentes. Destacou-se que as diversas soluções existentes focam o tema de maneira distinta dependendo de sua aplicação, algumas soluções por exemplo sugerem a utilização de diferentes tipos de atuadores enquanto outras realizam medições e tratamentos de sinais nervosos dos pacientes.

Para o desenvolvimento deste trabalho, procurou-se trabalhar na resolução do problema da dificuldade da aplicação da robótica de reabilitação mecânica para a reprodução do movimento de juntas biológicas reais. A fidelidade entre o movimento real e o simulado pela órtese se mostra um grande desafio, já que o desalinhamento do movimento e a consequente transmissão de forças pode se mostrar maléfico para o tratamento do paciente, causando dor e fadiga muscular.

Amigo et al. (2012), por exemplo, focaram no estudo das forças de transmissão geradas em decorrência do desalinhamento do centro de rotação da órtese da junta do cotovelo, utilizando sensores para medição de força e torque, conforme ilustrado na Figura 2.

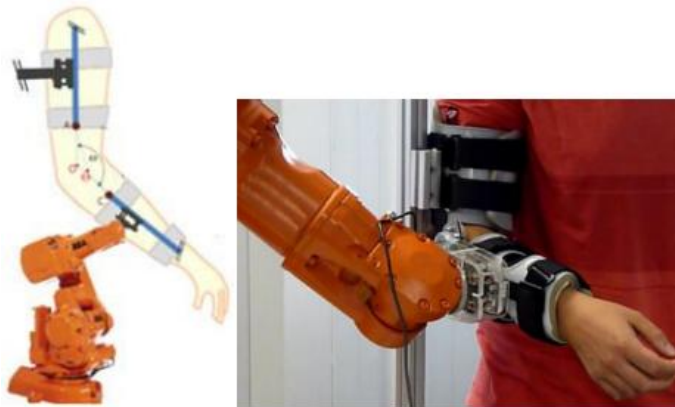


Figura 2 - Experimento para medida de forças de transmissão entre robô e paciente (Amigo et al., 2012)

Como um outro exemplo, Seki et al. (2011), Figura 3, abordaram o projeto de uma órtese para membro superior aplicada a pacientes com uma desordem neurológica denominada Tremor Essencial. O projeto de controle elétrico foi desenvolvido a fim de suprimir os tremores dos pacientes, fossem eles de causa voluntária ou involuntária. O projeto mecânico utilizou uma junta rotacional (centro de rotação fixo), e o controle inclui uma malha de processamento dos sinais neurológicos a fim de se minimizar os tremores.



Figura 3 - Protótipo desenvolvido por Seki et al. (2011) utilizando-se uma junta rotacional e processamento de sinal elétrico

Percebeu-se ainda que há diversos tipos de atuadores que podem ser utilizados, como por exemplo nos trabalhos de Vanderniepen et al. (2008) em que os autores também fornecem a modelagem de um mecanismo articulado por uma junta rotacional sim-

ples, porém adaptada com uma mola pré-tensionada que fornece uma variação de força ao longo do movimento do mecanismo. O mecanismo proposto utiliza dois atuadores, cada qual simulando o movimento do braço e antebraço do paciente conforme ilustrado na Figura 4 e Figura 5.



Figura 4 - Desenho em CAD do protótipo criado por Vanderniepen et al. (2008) com um pequeno atuador na região inferior



Figura 5 - Protótipo criado por Vanderniepen et al. (2008) que utiliza uma mola pré-tensionada

Outro tipo de atuador pode ser encontrado nos trabalhos de Pylatiuk et al. (2009) Figura 6, em que os autores descrevem o projeto de um protótipo de órtese para membro superior utilizando atuadores flexíveis a base de fluidos, com estímulo e controle

embutido no mecanismo, tornando-o portátil. Sensores eletromiográficos são utilizados para obter a entrada da malha de controle do mecanismo, que controla a atuação do motor de fluido da órtese. Nota-se também a utilização de uma junta rotacional no protótipo.

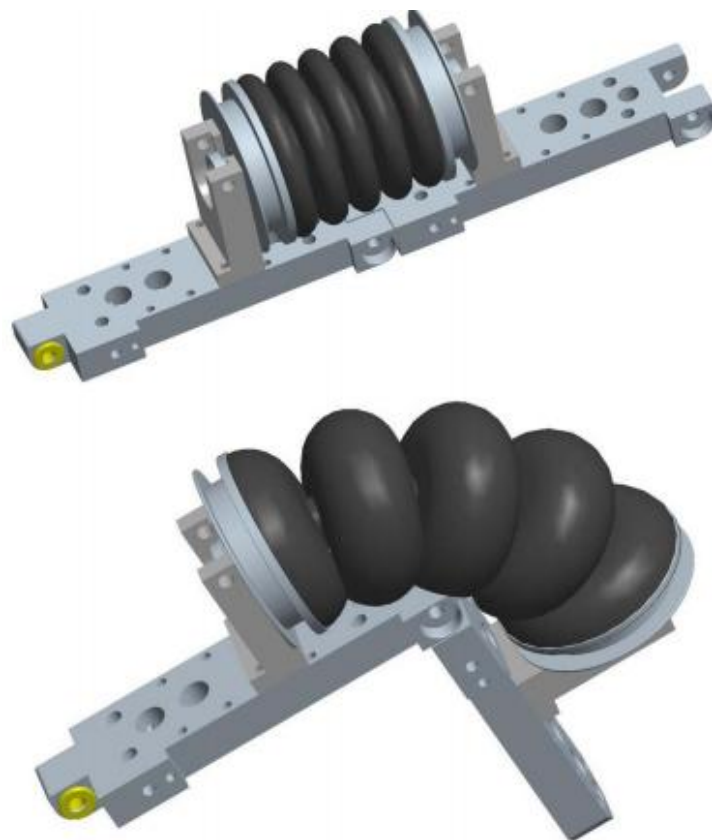


Figura 6 - Protótipo com a junta desenvolvida por Pylatiuk et al. (2009) retraída e expandida de 90°

Diversas soluções semelhantes ao problema da robótica aplicada à reabilitação humana são encontradas atualmente. Especificamente para a órtese da junta do cotovelo, muitas soluções envolvem o projeto do mecanismo com um grau de liberdade, utilizando muitas vezes uma junta rotacional. Os diferentes projetos se diferenciam pelo tipo de acionamento do mecanismo, porém todos possuem em comum a topologia simples de uma junta rotacional bidimensional.

Apesar de cada estudo se diferenciar no que diz respeito ao controle de força e torque da órtese, a solução do problema do centro de rotação variável ainda não está muito bem consolidada. É importante estudar uma solução para este problema, que resultaria numa órtese mais fiel ao movimento humano, que conseqüentemente produziria resultados mais eficazes nas terapias de reabilitação.

É importante, portanto, o estudo da adaptação do mecanismo quando utilizado em um contexto terapêutico em ser humano. Este projeto seguirá as diretrizes já propostas no âmbito da reabilitação robótica, porém se destacará dos demais projetos pela busca de um mecanismo cujo movimento seja fiel ao movimento biológico real de um cotovelo, com a proposta da utilização de um mecanismo de cadeia passiva baseada em um quadrilátero articulado.

A próxima seção descreverá a proposta de metodologia de desenvolvimento deste trabalho. Os resultados alcançados se encontram nas próximas seções.

3. MATERIAIS E MÉTODOS

Nesta seção serão descritos o equacionamento teórico, os procedimentos experimentais, as simulações e a organização do projeto ao longo do curso do trabalho.

3.1. ORGANIZAÇÃO E CRONOGRAMA

Para o desenvolvimento deste trabalho, foi escolhida uma abordagem de estudo, projeto e desenvolvimento do protótipo a partir de uma divisão de subsistemas. A criação destes subsistemas (mecânico, elétrico e programação) visa uma melhor estrutura para a aplicação das metodologias de projeto, uma vez que a divisão em subsistemas permite uma abordagem em espiral de projeto independente para cada frente do projeto. Ao refinar cada subsistema separadamente, observa-se uma melhor organização do projeto como um todo. A

Figura 7 ilustra a composição dos subsistemas.

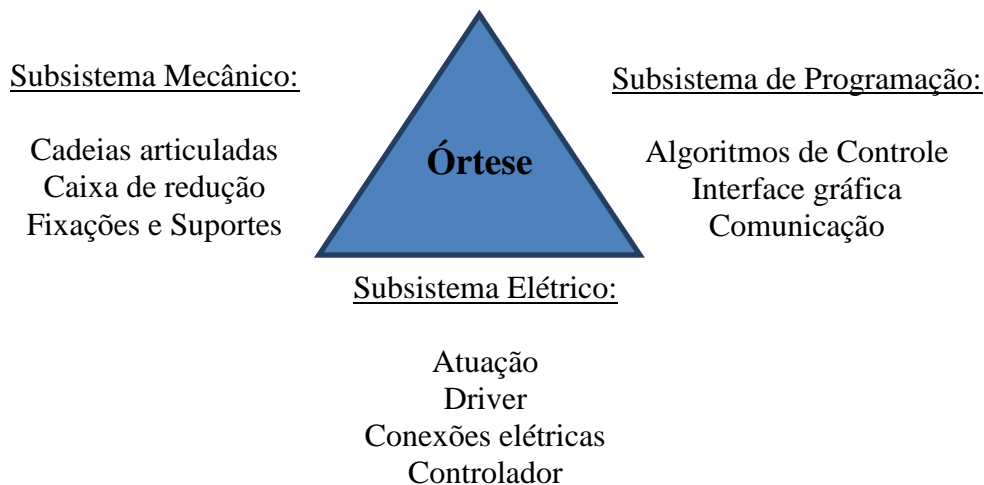


Figura 7 - Diagrama da composição dos subsistemas

Após a definição dos subsistemas, definiu-se um cronograma de trabalho enfatizando todas as atividades macro que serão realizadas ao decorrer do desenvolvimento do projeto. A primeira etapa do trabalho concentrou as atividades de projeto e construção do subsistema mecânico do protótipo e a segunda etapa agregou as atividades de projeto dos subsistemas elétricos e de programação bem como a realização da montagem dos conjuntos e realização de testes operacionais.

3.2. MECANISMO DE CADEIA PASSIVA

O exoesqueleto construído neste projeto é baseado no conceito de mecanismo de cadeia passiva. Este mecanismo possui duas cadeias em paralelo, uma ativa e outra passiva. A cadeia passiva é efetivamente a cadeia do mecanismo que define a mobilidade do conjunto. Para movimentar esta cadeia passiva faz-se portanto o uso de uma cadeia ativa em paralelo.

Com o intuito de ilustrar o conceito de mecanismo de cadeia passiva, supõe-se a seguinte cadeia de duas barras seriadas, articuladas entre si e em pontos do "Braço" e "Antebraço", como ilustra a Figura 8. Inicialmente supõe-se inexistente a junta que une o "Braço" ao "Antebraço". De acordo com a representação da Figura 8, a mobilidade do mecanismo é igual a 3, ou seja, acionando a rotação da barra conectada ao "Braço", o corpo rígido denominado "Antebraço" pode executar duas translações no plano e uma rotação no eixo perpendicular ao plano. Trata-se portanto, de um mecanismo com maior mobilidade do que a junta biológica do membro superior.

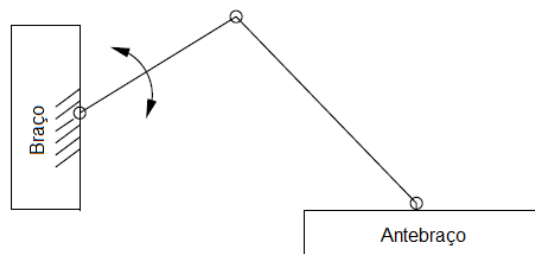


Figura 8 - Mecanismo de cadeia seriada com mobilidade igual a 3

No entanto, o membro superior humano é um mecanismo com mobilidade mais limitada, isto é, movimenta os corpos rígidos dentro de limites de distância definidos pela anatomia da junta biológica. Portanto, para restringir o movimento do mecanismo da Figura 8, deve-se adicionar uma cadeia passiva entre os corpos rígidos "Braço" e "Antebraço", que em essência é a junta Úmero-Rádio-Ulnar do paciente que utilizará a órtese.

Segundo Coelho (2008), se existe uma cadeia passiva em paralelo com a cadeia ativa do mecanismo, então o conjunto terá mobilidade exatamente igual à mobilidade da cadeia passiva. Assim sendo, para que o mecanismo da Figura 8 tenha mobilidade igual à do membro superior humano, a cadeia passiva a ser adicionada deve ser a da junta biológica do Úmero, Rádio e Ulna. Portanto, ao acoplar o mecanismo da Figura 8 no membro superior de um ser humano, a cadeia seriada passará a acompanhar fielmente os movimentos do membro superior humano.

Além disso, será necessário controlar o acionamento da cadeia ativa, para que o membro superior humano se movimente de acordo com o esperado. O controle deste acionamento deve levar em consideração a modelagem cinemática da junta biológica. Esta modelagem será vista na próxima seção, onde a junta Úmero-Rádio-Ulnar será modelada como um mecanismo de 4 barras articuladas, ou quadrilátero articulado. O mecanismo completo, ou seja, o exoesqueleto acoplado ao corpo humano, pode ser esquematizado como o modelo da Figura 9. É possível observar, portanto, que a adição da cadeia passiva restringe o movimento da cadeia seriada, e o conjunto possui agora a mesma mobilidade que a cadeia passiva. O protótipo a ser construído será somente a cadeia ativa, que movimentará o membro superior humano, modelado como um quadrilátero articulado.

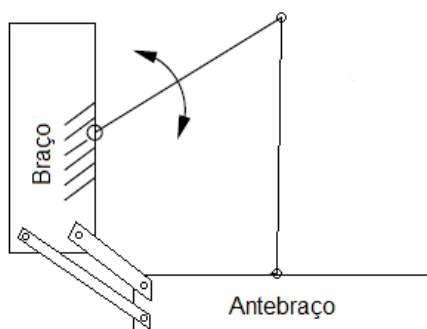


Figura 9 - Cadeia passiva da junta Úmero-Rádio-Ulnar modelada como quadrilátero articulado

É importante notar que o membro superior pode acionar e movimentar a cadeia seriada assim como a cadeia seriada pode acionar a junta biológica e movimentar o membro superior. As duas cadeias em paralelo formam um mecanismo de mobilidade exatamente igual à mobilidade do braço e antebraço humano, já que o membro superior humano passa a ter o papel de cadeia passiva e portanto define a mobilidade do mecanismo como um todo.

A órtese a ser projetada neste trabalho é um mecanismo de duas barras seriadas, como ilustrado na Figura 8. O acionamento de uma das barras movimentará o membro superior humano que é modelado como a cadeia passiva do mecanismo. Porém,

para garantir a repetibilidade do movimento da cadeia passiva, deve-se estudar a modelagem da junta do membro superior, que é apresentada nas seções posteriores.

3.3. MODELAGEM DA JUNTA BIOLÓGICA

Para o dimensionamento da junta do tipo quadrilátero articulado, inicialmente foi utilizado um método analítico para a síntese do mecanismo e análise de viabilidade. O método abordado a seguir utiliza três posições prescritas que o mecanismo deve alcançar. O método se inicia escolhendo um sistema de coordenadas como origem e separando o mecanismo plano 4R em duas metades: a metade esquerda e a metade direita. A resolução das duas metades se dá de maneira análoga e é realizada em função da escolha de alguns parâmetros de entrada.

O equacionamento do lado esquerdo é feito acompanhando a notação da Figura 10. Considerou-se como prescritos os pontos P_1 , P_2 , e P_3 bem como as orientações correspondentes das linhas $\overline{A_1P_1}$, $\overline{A_2P_2}$, e $\overline{A_3P_3}$. O ângulo entre $\overline{A_2P_2}$ e $\overline{A_1P_1}$ é α_2 e o ângulo entre $\overline{A_3P_3}$ e $\overline{A_1P_1}$ é α_3 .

É importante notar quais são as entradas e saídas deste equacionamento. Como objetivo, pretende-se determinar os ângulos β_2 , β_3 , \vec{W} e \vec{Z} , totalizando 6 incógnitas que devem ser dimensionadas. As entradas para o problema são os pontos P_1 , P_2 , e P_3 , as orientações correspondentes das linhas $\overline{A_1P_1}$, $\overline{A_2P_2}$, e $\overline{A_3P_3}$ e as coordenadas do centro fixo O_2 . Vale lembrar que este dimensionamento refere-se ao lado esquerdo do mecanismo, e o cálculo do lado direito é análogo, determinando-se as dimensões das barras do lado direito do mecanismo.

É possível observar que os vetores Z e S estarão no mesmo corpo rígido, no caso deste trabalho, estarão na região do antebraço do exoesqueleto. Além disso, os pontos O_2 e O_4 também se encontram fixos, na região do braço do exoesqueleto. Desta forma, os dois principais elementos do mecanismo de quadrilátero articulado são as barras móveis W e U . A barra W se refere ao elemento a ser dimensionado do lado esquerdo do mecanismo, enquanto a barra U se refere ao elemento a ser dimensionado

do lado direito do mecanismo. É importante destacar a nomenclatura das barras W e U, pois esta nomenclatura será utilizada em seções posteriores.

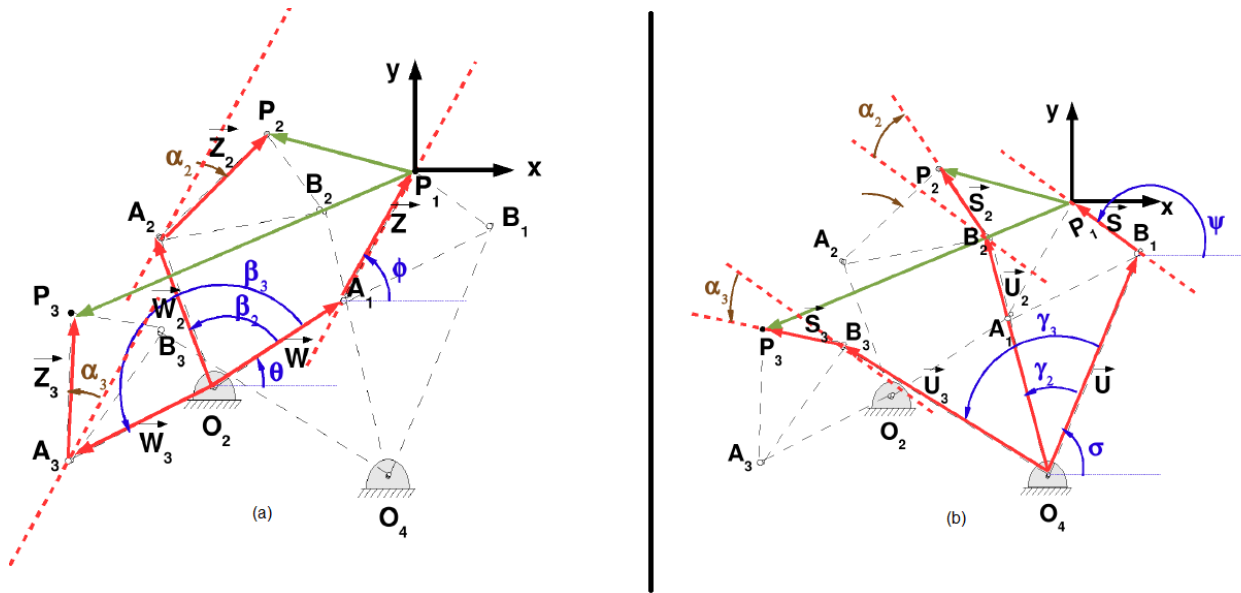


Figura 10 - Vetores para equacionamento. (a) Lado esquerdo do mecanismo. (b) Lado direito do mecanismo. Extraído de Coelho, 2009.

Utilizou-se a seguinte notação para os pontos prescritos:

$$(P_{1x}; P_{1y}) (P_{2x}; P_{2y}) (P_{3x}; P_{3y})$$

$$\overrightarrow{P_2P_1} = (P_{2x} - P_{1x})\hat{i} + (P_{2y} - P_{1y})\hat{j} = (P_{2x} - P_{1x}) + i(P_{2y} - P_{1y}) \quad (1)$$

$$\overrightarrow{P_3P_1} = (P_{3x} - P_{1x})\hat{i} + (P_{3y} - P_{1y})\hat{j} = (P_{3x} - P_{1x}) + i(P_{3y} - P_{1y}) \quad (2)$$

Vetores do Lado Esquerdo

Para a posição inicial do mecanismo, tem-se:

$$\vec{R}_{A_1O_2} = \vec{W} = W_x\hat{i} + W_y\hat{j} = |\vec{W}|e^{i\theta} = |\vec{W}|\cos\theta + i|\vec{W}|\sin\theta \quad (3)$$

$$\vec{R}_{P_1A_1} = \vec{Z} = Z_x\hat{i} + Z_y\hat{j} = |\vec{Z}|e^{i\phi} = |\vec{Z}|\cos\phi + i|\vec{Z}|\sin\phi \quad (4)$$

Para as posições subsequentes do mecanismo, basta rotacionar os vetores de acordo com os ângulos β e α :

$$\vec{R}_{A_2O_2} = \vec{W}_2 = \vec{W}e^{i\beta_2} \quad (5)$$

$$\vec{R}_{P_2A_2} = \vec{Z}_2 = \vec{Z}e^{i\alpha_2} \quad (6)$$

$$\vec{R}_{A_3O_2} = \vec{W}_3 = \vec{W}e^{i\beta_3} \quad (7)$$

$$\vec{R}_{P_3A_3} = \vec{Z}_3 = \vec{Z}e^{i\alpha_3} \quad (8)$$

Equacionamento do Lado Esquerdo

Caminho $O_2 - A_1 - P_1 - O_2$:

$$\vec{W} + \vec{Z} = \overrightarrow{P_1O_2} \quad (9)$$

Caminho $O_2 - A_2 - P_2 - P_1 - A_1 - O_2$:

$$\vec{W}_2 + \vec{Z}_2 - \overrightarrow{P_2P_1} - \vec{Z} - \vec{W} = \vec{0} \quad (10)$$

Rearranjando os termos:

$$\vec{W}(e^{i\beta_2} - 1) + \vec{Z}(e^{i\alpha_2} - 1) = \overrightarrow{P_2P_1} \quad (11)$$

Caminho $O_2 - A_3 - P_3 - P_1 - A_1 - O_2$:

$$\vec{W}_3 + \vec{Z}_3 - \overrightarrow{P_3P_1} - \vec{Z} - \vec{W} = \vec{0} \quad (12)$$

Rearranjando os termos:

$$\vec{W}(e^{i\beta_3} - 1) + \vec{Z}(e^{i\alpha_3} - 1) = \overrightarrow{P_3P_1} \quad (13)$$

Tem-se então o seguinte sistema, utilizando as equações (9), (11) e (13):

$$\begin{cases} \vec{W} + \vec{Z} = \overrightarrow{P_1O_2} \\ \vec{W}(e^{i\beta_2} - 1) + \vec{Z}(e^{i\alpha_2} - 1) = \overrightarrow{P_2P_1} \\ \vec{W}(e^{i\beta_3} - 1) + \vec{Z}(e^{i\alpha_3} - 1) = \overrightarrow{P_3P_1} \end{cases}$$

Na forma matricial:

$$\begin{bmatrix} 1 & 1 \\ (e^{i\beta_2} - 1) & (e^{i\alpha_2} - 1) \\ (e^{i\beta_3} - 1) & (e^{i\alpha_3} - 1) \end{bmatrix} \begin{bmatrix} \vec{W} \\ \vec{Z} \end{bmatrix} = \begin{bmatrix} \overrightarrow{P_1O_2} \\ \overrightarrow{P_2P_1} \\ \overrightarrow{P_3P_1} \end{bmatrix}$$

Tem-se então um sistema com 6 equações (cada equação vetorial possui parte real e imaginária) e 6 incógnitas ($\beta_2, \beta_3, \text{real}(\overline{W}), \text{imag}(\overline{W}), \text{real}(\overline{Z})$ e $\text{imag}(\overline{Z})$). É importante notar que, dentre todas as variáveis conhecidas, as coordenadas de O_2 são os únicos parâmetros que não são conhecidos *a priori*, e são utilizados como a variável de projeto. Dependendo da escolha de O_2 , obter-se-á um mecanismo diferente, e após algumas iterações será possível determinar um ponto O_2 ideal para a aplicação deste mecanismo. Para que o sistema tenha solução indeterminada (ou seja, em função da escolha de O_2), seu determinante deve ser igual a zero, ou seja:

$$\begin{vmatrix} 1 & 1 & P_1 - O_2 \\ (e^{i\beta_2} - 1) & (e^{i\alpha_2} - 1) & P_2 - P_1 \\ (e^{i\beta_3} - 1) & (e^{i\alpha_3} - 1) & P_3 - P_1 \end{vmatrix} = 0 \quad (14)$$

Desenvolvendo o determinante, obtém-se a seguinte expressão:

$$\begin{aligned} & -e^{i\beta_2}(P_3 - O_2 - e^{i\alpha_3}(P_1 - O_2)) + e^{i\beta_3}(P_2 - O_2 - e^{i\alpha_2}(P_1 - O_2)) \\ & = -e^{i\alpha_2}(P_3 - O_2) + e^{i\alpha_3}(P_2 - O_2) \end{aligned} \quad (15)$$

A equação obtida é da forma:

$$\overline{v}_2 e^{i\beta_2} + \overline{v}_3 e^{i\beta_3} = \overline{v}_4 \quad (16)$$

onde:

$$\overline{v}_2 = -(P_3 - O_2 - e^{i\alpha_3}(P_1 - O_2)) \quad (17)$$

$$\overline{v}_3 = (P_2 - O_2 - e^{i\alpha_2}(P_1 - O_2)) \quad (18)$$

$$\overline{v}_4 = -e^{i\alpha_2}(P_3 - O_2) + e^{i\alpha_3}(P_2 - O_2) \quad (19)$$

Desta forma, tem-se os vetores conhecidos $\overline{v}_2, \overline{v}_3$ e \overline{v}_4 e as incógnitas β_2 e β_3 , representados geometricamente na Figura 11.

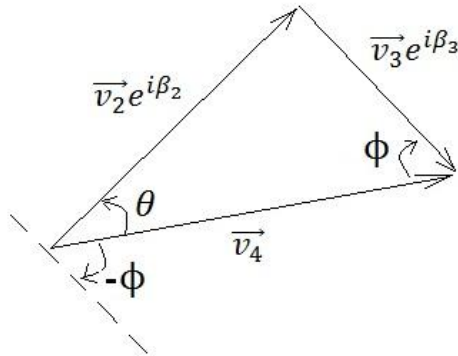


Figura 11 - Representação dos vetores \vec{v}_2 , \vec{v}_3 e \vec{v}_4 .

Utilizando a Lei dos Cossenos, obtém-se:

$$\|\vec{v}_3\|^2 = \|\vec{v}_2\|^2 + \|\vec{v}_4\|^2 - 2 \cdot \|\vec{v}_2\| \cdot \|\vec{v}_4\| \cdot \cos\theta \quad (20)$$

Analogamente,

$$\|\vec{v}_2\|^2 = \|\vec{v}_3\|^2 + \|\vec{v}_4\|^2 - 2 \cdot \|\vec{v}_3\| \cdot \|\vec{v}_4\| \cdot \cos\phi \quad (21)$$

É possível então, determinar o valor dos ângulos θ e ϕ .

Além disso, rotacionando-se o vetor \vec{v}_4 em um ângulo $+\theta$, obtém-se \vec{v}_2 , assim como a rotação de \vec{v}_4 em um ângulo $-\phi$, obtém-se \vec{v}_3 . Neste ponto é importante notar a dualidade de soluções. A Figura 11 mostra o equacionamento para determinado β_2 e β_3 . Se o triângulo inverso for considerado, de acordo com a Figura 12, ao rotacionar o vetor \vec{v}_4 em $-\theta$ e $+\phi$, obtém-se respectivamente os vetores \vec{v}_2 e \vec{v}_3 rotacionados de ângulos β_2' e β_3' , resultando numa solução também viável.

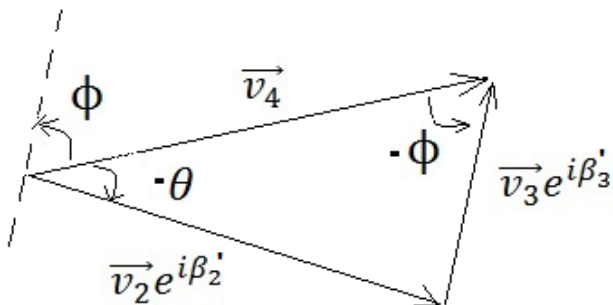


Figura 12 - Rotação dos vetores para o equacionamento.

Assim sendo, tem-se:

$$\frac{\vec{v}_2}{\|\vec{v}_2\|} e^{i\beta_2} = \frac{\vec{v}_4}{\|\vec{v}_4\|} e^{\pm i\theta} \quad (22)$$

$$\frac{\|\vec{v}_2\|}{\|\vec{v}_2\|} e^{i(\arg v_2 + \beta_2)} = \frac{\|\vec{v}_4\|}{\|\vec{v}_4\|} e^{i(\arg v_4 \pm \theta)} \quad (23)$$

Finalmente,

$$\beta_2 = \arg v_4 - \arg v_2 \pm \theta \quad (24)$$

Analogamente, para o vetor \vec{v}_3 , tem-se:

$$\frac{\vec{v}_3}{\|\vec{v}_3\|} e^{i\beta_3} = \frac{\vec{v}_4}{\|\vec{v}_4\|} e^{\mp i\phi} \quad (25)$$

$$\frac{\|\vec{v}_3\|}{\|\vec{v}_3\|} e^{i(\arg v_3 + \beta_3)} = \frac{\|\vec{v}_4\|}{\|\vec{v}_4\|} e^{i(\arg v_4 \mp \phi)} \quad (26)$$

Então,

$$\beta_3 = \arg v_4 - \arg v_3 \mp \phi \quad (27)$$

Portanto, dados os pontos P_1 , P_2 , e P_3 , as orientações correspondentes das linhas $\overline{A_1P_1}$, $\overline{A_2P_2}$, e $\overline{A_3P_3}$ e as coordenadas do centro fixo O_2 , determinou-se analiticamente os ângulos β_2 , β_3 . Ainda resta dimensionar o tamanho das barras do lado esquerdo do quadrilátero, ou seja, os vetores \vec{W} e \vec{Z} .

Assim, voltando ao sistema original, formado pelas as equações (9), (11) e (13), tem-se:

$$\begin{bmatrix} 1 & 1 \\ (e^{i\beta_2} - 1) & (e^{i\alpha_2} - 1) \\ (e^{i\beta_3} - 1) & (e^{i\alpha_3} - 1) \end{bmatrix} \begin{bmatrix} \vec{W} \\ \vec{Z} \end{bmatrix} = \begin{bmatrix} P_1 - O_2 \\ P_2 - P_1 \\ P_3 - P_1 \end{bmatrix}$$

Eliminando a primeira equação, tem-se que:

$$\begin{bmatrix} (e^{i\beta_2} - 1) & (e^{i\alpha_2} - 1) \\ (e^{i\beta_3} - 1) & (e^{i\alpha_3} - 1) \end{bmatrix} \begin{bmatrix} \vec{W} \\ \vec{Z} \end{bmatrix} = \begin{bmatrix} P_2 - P_1 \\ P_3 - P_1 \end{bmatrix}$$

Anteriormente, foi demonstrado a existência de uma dualidade de solução, pela presença de β_2 e β_3 e β_2' e β_3' . Desta forma, o sistema anterior possuirá quatro soluções, uma para cada par β_2 e β_3 . Porém, é possível demonstrar que somente uma das quatro soluções é viável para o objetivo do problema. Desta forma, o sistema de equações se torna definido e possui uma única solução viável, que será encontrada em simulações em Matlab. É importante lembrar que todo o equacionamento visto nesta seção é análogo para o lado direito do mecanismo. A resolução em Matlab forneceu os resultados que serão descritos na seção 3.5.

Uma vez equacionado o problema, é possível partir para a descrição do procedimento experimental que fornece os valores numéricos de entrada para a simulação em MATLAB, a ser descrita nas próximas seções.

3.4. COLETA DE DADOS EXPERIMENTAIS

Uma vez escolhida a topologia do mecanismo, partiu-se para o estudo da síntese do mesmo através de métodos disponíveis na literatura. Segundo Hess (2009), é possível equacionar um quadrilátero articulado e projetar suas dimensões tendo como entrada três posições prescritas do mecanismo e a variação angular destas posições.

A Figura 13 mostra o resultado da medição do experimento realizado para aquisição das posições do antebraço em relação ao braço. O experimento foi realizado em um adulto de 25 anos do sexo masculino, de estatura na média brasileira de 1,70 metro.

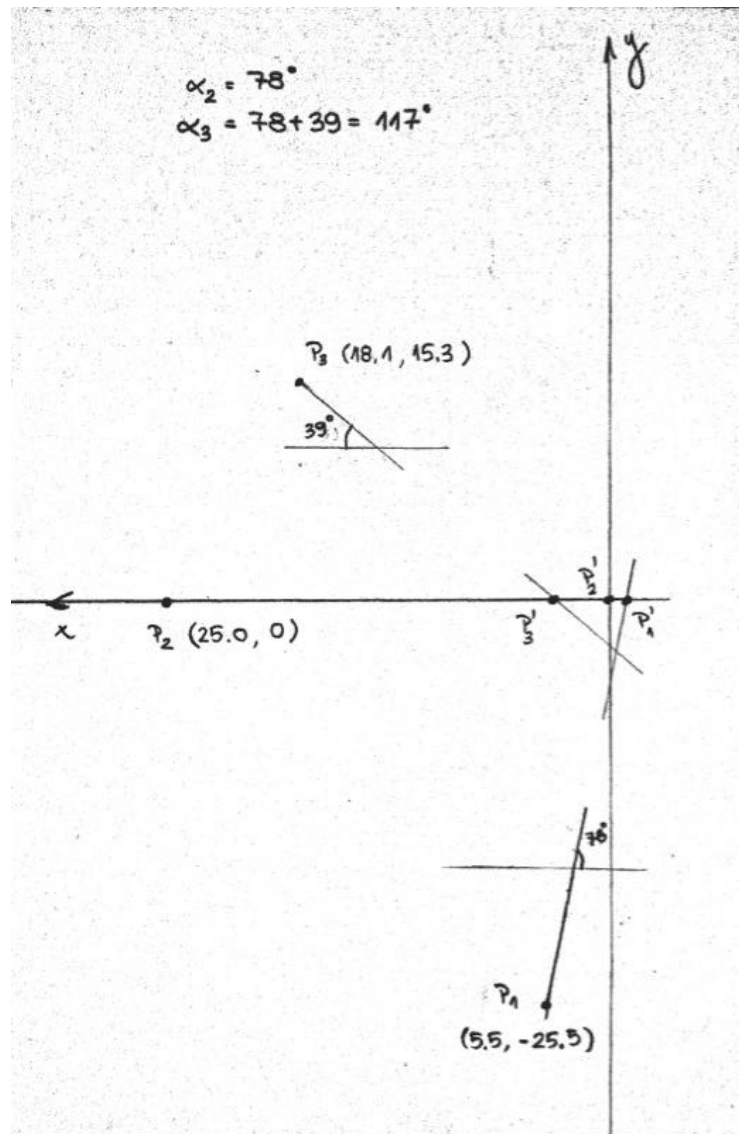


Figura 13 - Experimento realizado para coleta de dados experimentais

O procedimento experimental pode ser descrito nas seguintes etapas:

- 1) Foi estabelecido um sistema de coordenadas globais em uma folha A3, onde o eixo y representa o eixo do braço (osso do úmero) enquanto o eixo x representa o eixo do antebraço (ossos do rádio e ulna).
- 2) A folha A3 com o sistema de coordenadas foi fixada em um anteparo vertical, para que a medição de pontos pudesse ser feita de modo natural para o ser humano, ou seja, com o braço posicionado paralelamente ao corpo.
- 3) Fixou-se o braço do sujeito de teste de modo a não permitir seu deslocamento e rotacionou-se o antebraço, marcando as coordenadas dos pontos P_i , que pertencem ao corpo rígido do antebraço, junto com um pequeno segmento de reta que representa a direção do antebraço em uma dada posição. É possível notar, pela extensão do segmento de reta até a origem, que a rotação do antebraço de fato não possui um centro de rotação fixo, como esperado. Os pontos P_i' , ponto de intersecção do segmento com o eixo x, não convergem para um mesmo local, o que justifica a escolha de uma junta de centro de rotação variável para o projeto.
- 4) As coordenadas dos pontos e a inclinação dos segmentos foram medidos devidamente, para serem utilizados como dados de entrada nas simulações posteriores.

Como o método de síntese do mecanismo considera somente três pontos prescritos, optou-se por medi-los em um curso de 117° , partindo da posição de extensão total do antebraço em relação ao braço. Caso os três pontos fossem medidos no curso total do movimento (extensão total até flexão total do membro), haveria uma perda considerável de precisão do movimento do mecanismo, já que três pontos seriam insuficientes para discretizar o movimento do mecanismo.

Finalmente, como ilustra a Figura 13, optou-se por garantir a plena extensão do antebraço em oposição à plena flexão do mesmo e através da sequência dos pontos e da diferença angular entre eles, foi possível dimensionar um mecanismo 4R, como será abordado em seções posteriores.

3.5. SIMULAÇÃO EM AMBIENTE MATLAB

Após o equacionamento matemático, implementou-se a solução em Matlab e foram simulados diferentes casos, onde as variáveis de projeto eram as coordenadas dos pontos O_2 e O_4 .

Como o eixo vertical utilizado para o experimento simulava a posição do braço, foi sugerido a posição $O_2 = -2.00 + 3.00i$ e $O_4 = 1 + 5i$ para os centros fixos da órtese. O código para a simulação encontra-se no Apêndice A, e o resultado obtido pode ser visualizado na Figura 14 e Figura 15.

A Figura 14 mostra a visão macroscópica do mecanismo, e a Figura 15 detalha a região da junta 4R. Os pontos P1, P2 e P3 foram medidos na região do punho durante o experimento, o que justifica a representação dos pontos em regiões distantes da origem do gráfico. Além disso, é interessante notar a proporção entre a região da junta mecânica propriamente dita e o resto do mecanismo. Os segmentos maiores mostrados na Figura 14 são equivalentes ao antebraço do membro superior.

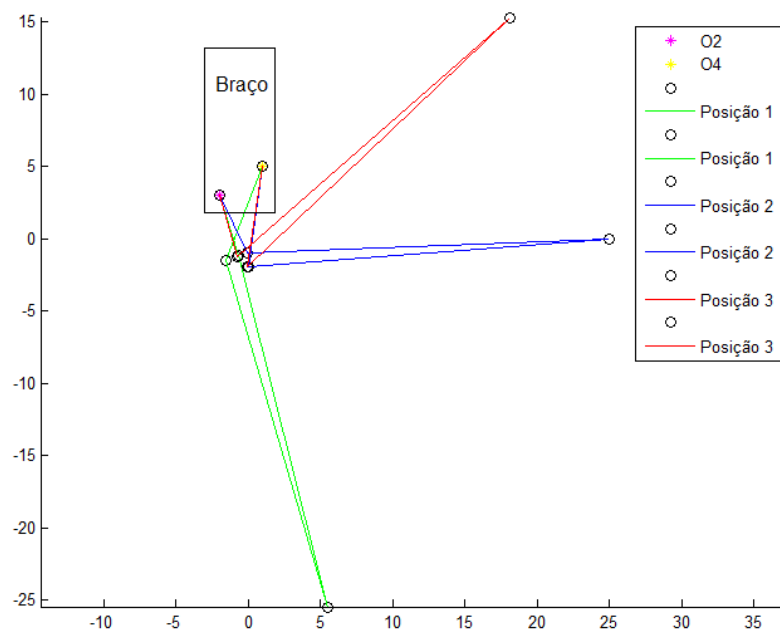


Figura 14 - Representação macroscópica da órtese em Matlab.

É possível validar a simulação de acordo com o equacionamento previamente feito, uma vez que os pontos P_i medidos experimentalmente estão de acordo com os pontos extremos de localização do mecanismo obtido em Matlab. No entanto, a Figura 14 somente ilustra de maneira macroscópica que o mecanismo de fato passa pelos pontos prescritos. O resultado de interesse para a simulação é a região da junta, que está detalhada na Figura 15.

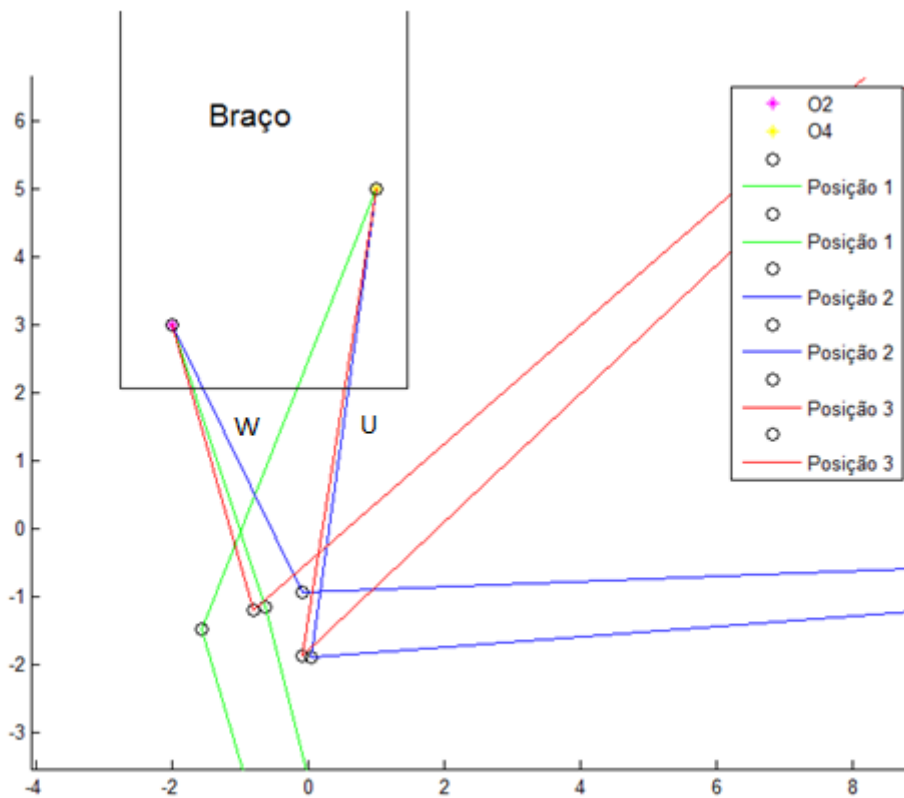


Figura 15 - Detalhe do mecanismo 4R na região das juntas mecânicas.

O resultado obtido pelo Matlab atendeu às especificações de projeto, e aqui cabem alguns comentários a respeito da solução da simulação. Os pontos pivôs do braço do exoesqueleto foram denominados O_2 e O_4 , representados respectivamente pelos círculos preto e rosa. Além disso, de acordo com a nomenclatura definida na seção 3.3, os pontos O_2 e O_4 estão conectados por uma junta rotacional simples às barras W e U, respectivamente, que estão indicadas na Figura 15.

Como dados de saída do Matlab, as dimensões das barras W e U são, em milímetros:

Barra W = 43,8 mm

Barra U = 69,7 mm

Como conclusão da simulação em Matlab, observa-se que o mecanismo proposto atende às especificações de projeto, portanto é possível modelar a junta Úmero-Rádio-Ulnar por meio de um mecanismo de 4 barras articuladas. É determinado então um mecanismo virtual equivalente ao mecanismo biológico do membro superior. Não é necessária a construção física do mecanismo virtual. No entanto, o resultado numérico obtido é utilizado no equacionamento do controle de acionamento da órtese.

3.6. MODELAGEM EM AMBIENTE CAD

Após a simulação e dimensionamento do quadrilátero articulado em Matlab, foi possível partir para a modelagem da solução obtida em CAD, utilizando o software SolidWorks. Durante todas as modelagens em CAD, o material considerado para a estrutura do exoesqueleto foi o alumínio, que garante um menor peso para a órtese e assegura a transmissão de forças e torques suficientes para o uso projetado.

A modelagem do exoesqueleto segue o conceito descrito na seção 3.2, isto é, a órtese projetada consiste em um mecanismo de duas barras seriadas e articuladas, que serão acopladas no membro superior humano e acionada por um motor de passo localizado no braço da estrutura.

A escolha do motor dentre as opções de motores DC, servomotores e motores de passo considerou os critérios de peso, custo, facilidade de controle e necessidade de hardware adicional. A Tabela 1 ilustra a matriz de decisão para a escolha do motor de acionamento da órtese. De acordo com os critérios estabelecidos e a respectiva ponderação, definiu-se o uso do motor de passo para acionamento do exoesqueleto, justificado principalmente pelo custo e por prover facilidade no controle em malha aberta.

Tabela 1 - Matriz de decisão para escolha do motor

	Ponderação	Stepper	DC	Servo
Custo	2	10	7	5
Controle	3	9	6	10
Peso	2	5	8	8
Necessidade de hardware adicional	1	9	4	7
	Relação final	8,25	6,5	7,875

A fim de estudar o comportamento da órtese em testes futuros, determinou-se que a cadeia seriada seria composta de duas barras de mesmo tamanho, articulada em um ponto fixo no braço e um ponto variável e ajustável no antebraço. Dessa forma, todos os parâmetros dimensionais da estrutura estão determinados e são constantes, sendo que o único parâmetro variável é a fixação da articulação no antebraço da órtese. Tendo somente uma variável, será possível estudar a influência da fixação da estrutura no membro do paciente, a fim de minimizar os esforços do motor e produzir uma terapia mais confortável ao paciente.

O modelo em CAD do exoesqueleto projetado se encontra na Figura 16. É importante observar que este modelo final foi resultado de diversas iterações, que fizeram parte do processo de refinamento da espiral de projeto. Partiu-se de um modelo conceitual bidimensional modelado em Working Model e uma vez validado partiu-se para a modelagem tridimensional em SolidWorks. Esta seção descreve detalhadamente o modelo projetado.

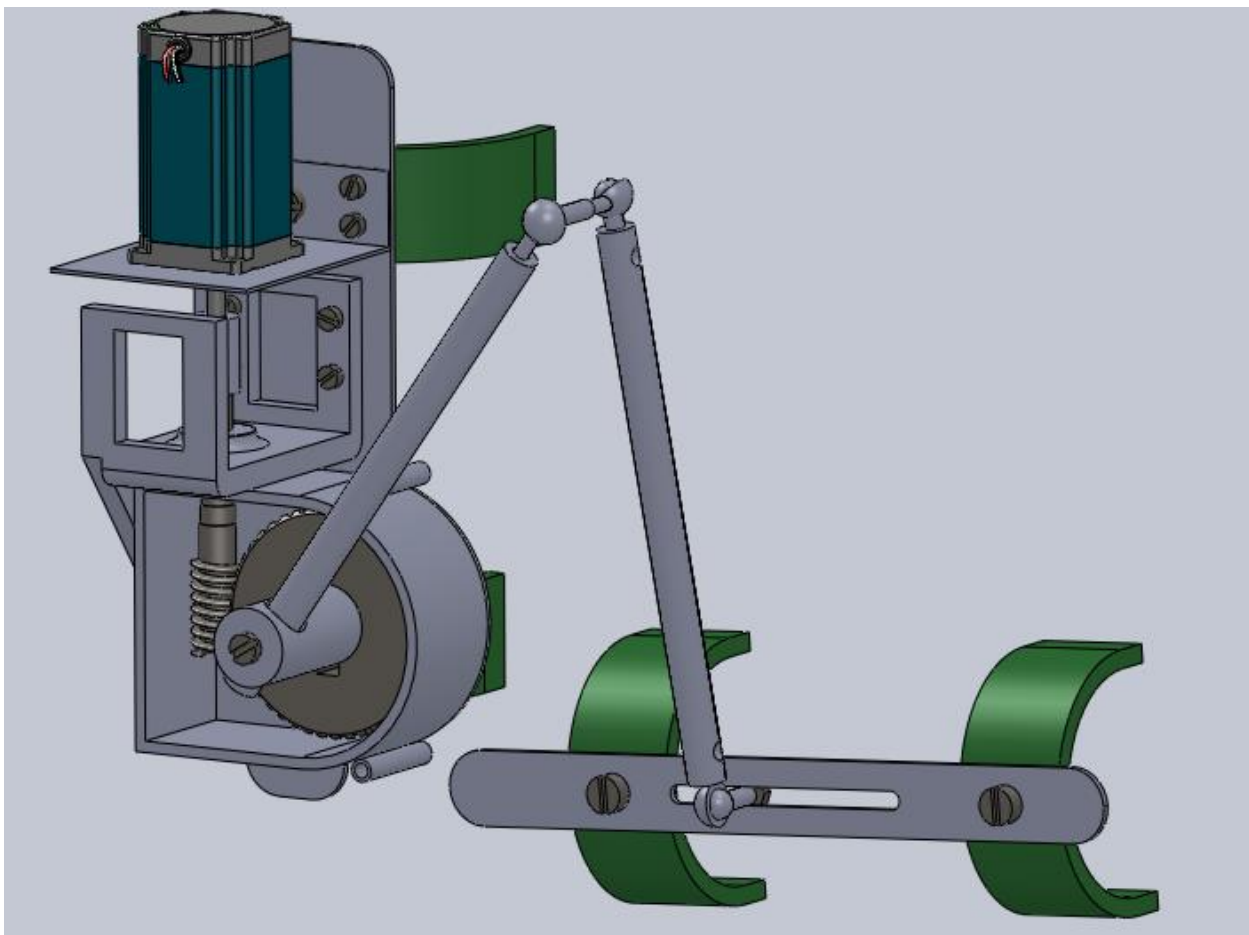


Figura 16 - Modelo tridimensional do exoesqueleto projeto

A Figura 16 ilustra a modelagem do exoesqueleto a ser construído durante o projeto. Nota-se duas peças estruturais principais, o "braço" e "antebraço" da órtese, que serão acoplados no membro superior do paciente. A ligação entre as duas partes se dá por meio da cadeia seriada de duas barras, também ilustrado na Figura 16. Uma das barras é acionada por um motor de passo que movimenta a estrutura e o membro superior do paciente, fazendo o papel de cadeia passiva. A escolha do motor de passo é justificada em seções posteriores.

A Figura 17 ilustra o detalhe da região de transmissão do movimento do motor para a barra acionada.

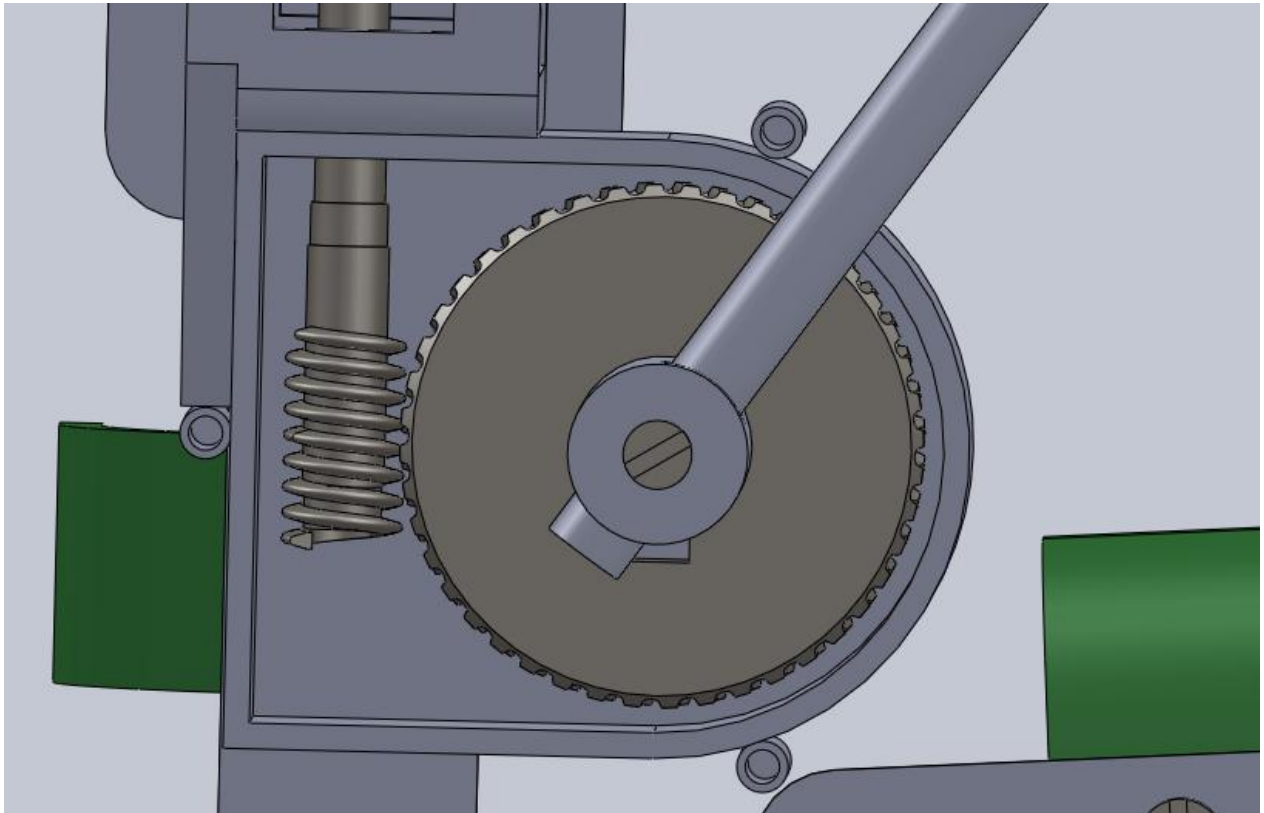


Figura 17 - Detalhe da transmissão ortogonal de movimento por *worm gear*

Por questões ergonômicas e de balanceamento estrutural, optou-se por posicionar o motor de passo na vertical, com o eixo direcionado para baixo. Desta forma, fez-se necessário o projeto de uma transmissão por engrenagens que gerasse um movimento ortogonal à rotação do motor. Além da ortogonalidade, a transmissão também deveria diminuir a rotação de saída e aumentar o torque fornecido à barra acionada. Para isso, foi utilizado um par de engrenagens de dentes helicoidais (*worm drive*), de relação de transmissão de 20:1. Utilizando um motor de passo de 5kgf.cm., o torque de saída da caixa de redução será de 100kgf.cm, suficiente para o movimento do antebraço do membro superior humano quando acoplado ao exoesqueleto.

A fixação do motor de passo na estrutura pode ser vista na Figura 18 (a), parafusada em um suporte que também é fixo na estrutura da órtese. Essa montagem garante a imobilização do motor na estrutura e evita folgas e movimentos indevidos que podem levar à falha do acionamento da órtese. De maneira similar, a caixa de redução também

é fixada na estrutura por meio de parafusos, e o eixo do motor é acoplado ao eixo da caixa de redução por meio de um acoplamento usinado em alumínio, como ilustra a Figura 18 (b).

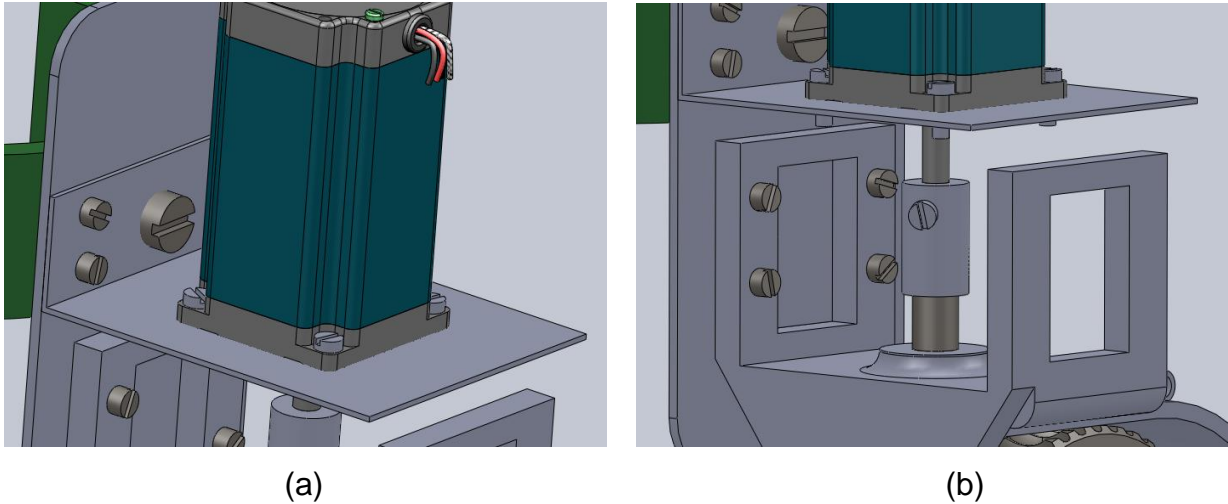


Figura 18 - (a) Fixação do motor de passo e (b) acoplamento entre eixo do motor e eixo da caixa de redução.

Como discutido anteriormente, para melhor estudar os efeitos da órtese na terapia de reabilitação humana, optou-se por fixar algumas montagens, tornando constantes alguns parâmetros dimensionais e restringindo as variáveis de controle. Desta forma, o ponto de articulação da barra acionada no "braço" da órtese foi determinado e fixado, não podendo ser ajustado. Similarmente, o tamanho das barras também foi previamente determinado e não será ajustável. O único parâmetro que pode ser ajustável é o ponto de articulação da cadeia seriada no "antebraço" da órtese, como ilustra a Figura 19. Assim sendo, tem-se somente uma variável de controle a ser estudada, e portanto pode-se analisar a influência da variação deste parâmetro na eficiência da órtese na terapia de reabilitação.

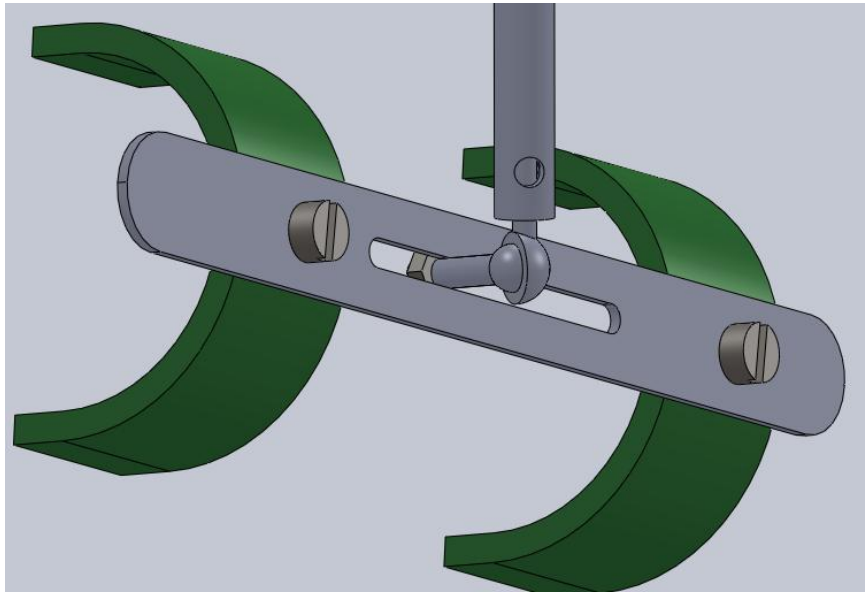


Figura 19 - Fixação ajustável da cadeia seriada na estrutura da órtese

Nota-se, a partir da Figura 20, que a cadeia seriada não se movimenta no mesmo plano, o que justifica a escolha de juntas esféricas em substituição às juntas rotacionais simples. A junta esférica permite uma folga natural do membro superior, que deve ser acomodada para evitar incômodos no paciente. Além disso, a junta esférica permite o movimento da cadeia seriada no espaço tridimensional, como ilustrado na Figura 20.

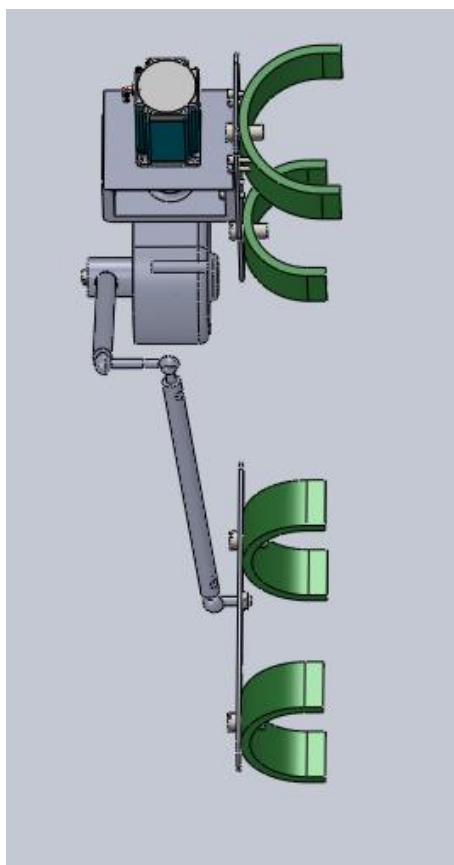


Figura 20 - Vista frontal da órtese

A vista lateral da órtese, ilustrada na Figura 21, mostra o mecanismo completo, pronto para ser acoplado ao membro superior humano. Esta órtese possui mobilidade igual a 3, isto é, a peça denominada "antebraço" pode realizar duas translações e uma rotação em torno do eixo perpendicular ao plano do mecanismo. No entanto, a mobilidade da órtese sem o membro humano é muito maior do que a que se almeja alcançar. A adição da cadeia passiva com o membro superior humano limita a mobilidade da órtese e transforma o conjunto completo em um mecanismo de mobilidade exatamente igual à mobilidade do membro superior humano, acionado por um motor de passo.

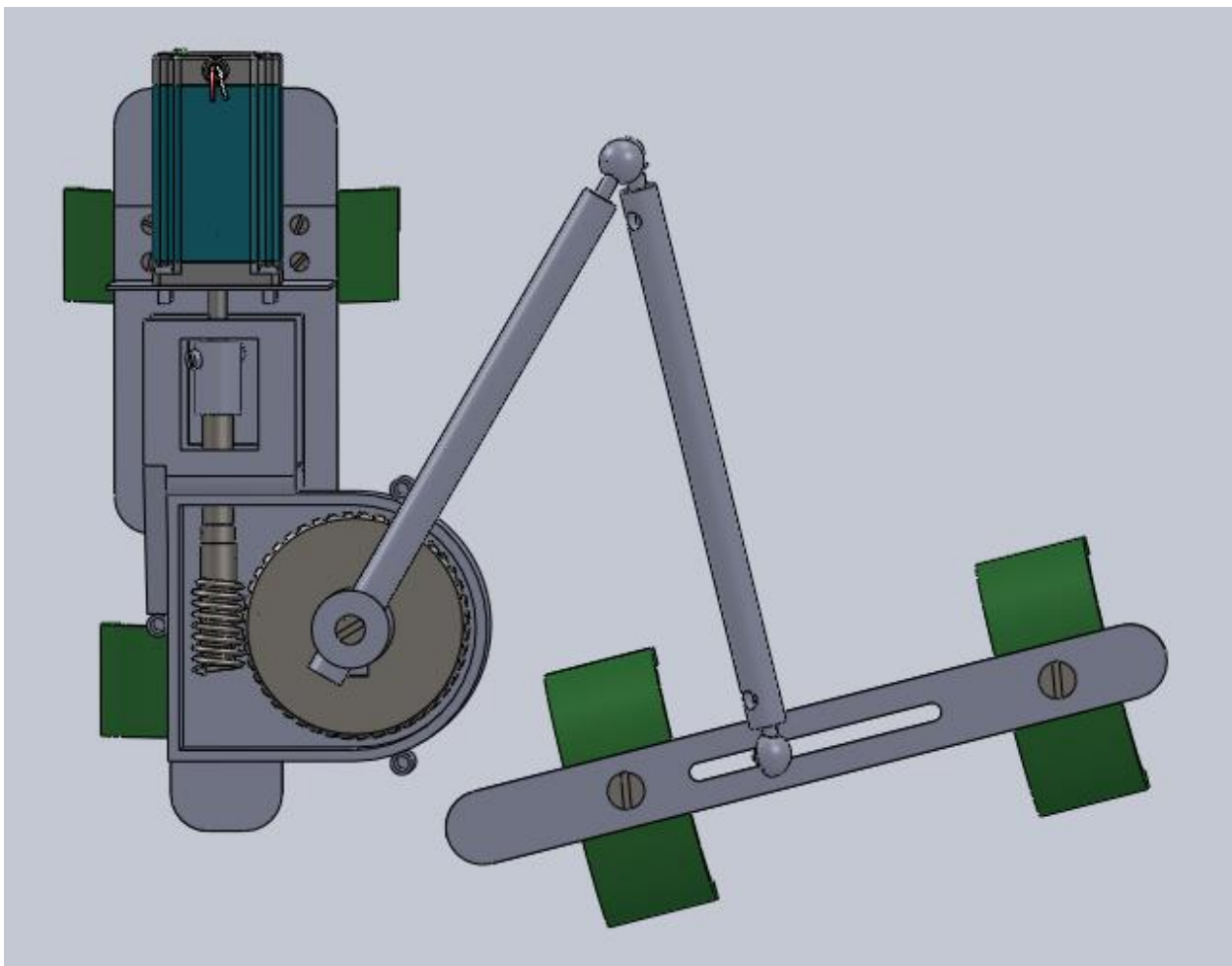


Figura 21 - Vista lateral da órtese

Dessa forma, a modelagem tridimensional em SolidWorks se torna validada e atende aos requisitos de projeto, autorizando a construção do protótipo. Os desenhos de montagem e desenhos de conjunto se encontram no Apêndice B.

4. RESULTADOS

4.1. CONSTRUÇÃO DO PROTÓTIPO

Após o estudo de viabilidade e simulação em Matlab, observou-se que o mecanismo proposto atende funcionalmente aos requisitos de projeto. Além disso, o exoesqueleto deve ter uma construção simples e prover a ergonomia do dispositivo, que será utilizado por pacientes e não deve, de maneira alguma, ser inseguro ou produzir efeitos negativo na terapia. Desta forma, o estudo final do design culminou no exoesqueleto ilustrado na Figura 21. O exoesqueleto proposto atende funcionalmente o requisito do projeto, e também apresenta construção simples e ergonomia ao paciente que utilizará o dispositivo.

Como etapa seguinte ao projeto, partiu-se para a construção e montagem do protótipo. A fim de se construir um exoesqueleto que seja leve e de uso prático, o projeto considera que a estrutura básica da órtese (apoio para braço, antebraço e barras móveis) serão de alumínio, que suporta as cargas mecânicas do motor e garante um menor peso para a estrutura. O sistema de transmissão e os elementos de fixação serão de aço, para garantir maior confiabilidade e segurança ao projeto. Além disso, as bráçadeiras ilustradas na modelagem em CAD são versões preliminares, e sofrerão modificações na versão do protótipo. Também será proposta uma camada de espuma no contato do membro superior com o exoesqueleto, para garantir o conforto do produto.

As peças principais do exoesqueleto foram então construídas e o resultado está ilustrado na Figura 22. As peças (b) e (c) serão as partes do exoesqueleto em contato com o membro superior humano, ou seja, serão o "braço" e "antebraço" da estrutura. O conjunto (a), formado pela caixa de redução e as duas barras articuladas, serão fixadas na placa (b), junto com o motor de passo (d) devidamente fixo no seu suporte (e). Por fim, a extremidade livre da cadeia seriada será fixada na placa do antebraço (c), onde será feito um furo oblongo para ajuste da fixação da cadeia seriada.

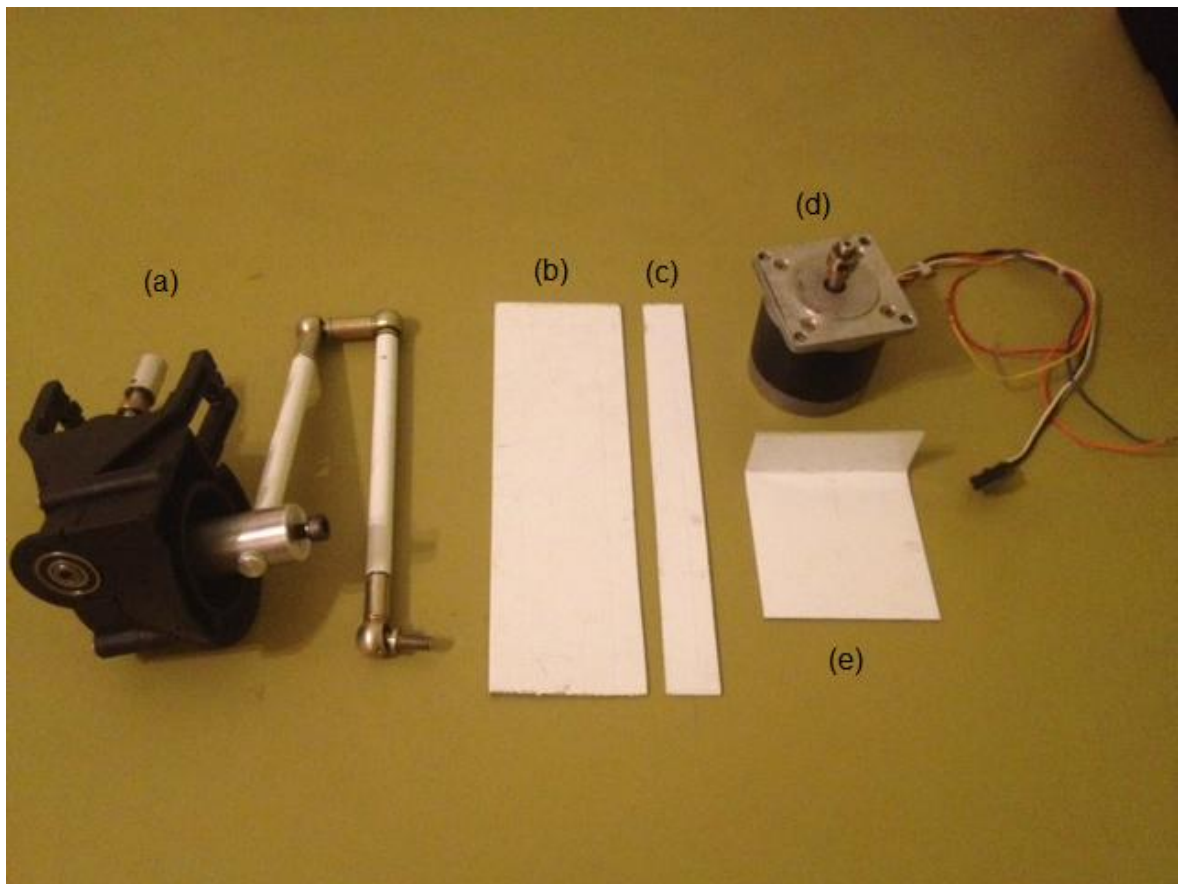


Figura 22 - Peças principais do exoesqueleto: (a) caixa de redução e cadeia seriada, (b) placa de fixação dos equipamentos do "braço", (c) placa de fixação dos equipamentos do "antebraço", (d) motor de passo e (e) suporte do motor de passo

A Figura 23 mostra com maior detalhe a caixa de redução e a cadeia seriada articulada do exoesqueleto. É possível notar a ortogonalidade da transmissão do movimento de entrada e saída da caixa de redução. Além disso, é possível observar as articulações por juntas esféricas entre as barras da cadeia seriada.



Figura 23 - Detalhe da caixa de redução e da cadeia seriada do exoesqueleto

A montagem das peças seguiu o projeto modelado em CAD, e resultou no protótipo ilustrado na Figura 24 e Figura 25. É evidente que alguns ajustes serão feitos uma vez que todos os subsistemas forem integrados, mas a versão inicial do subsistema mecânico mostrou-se adequado às necessidades, e atende aos requisitos de projeto propostos.

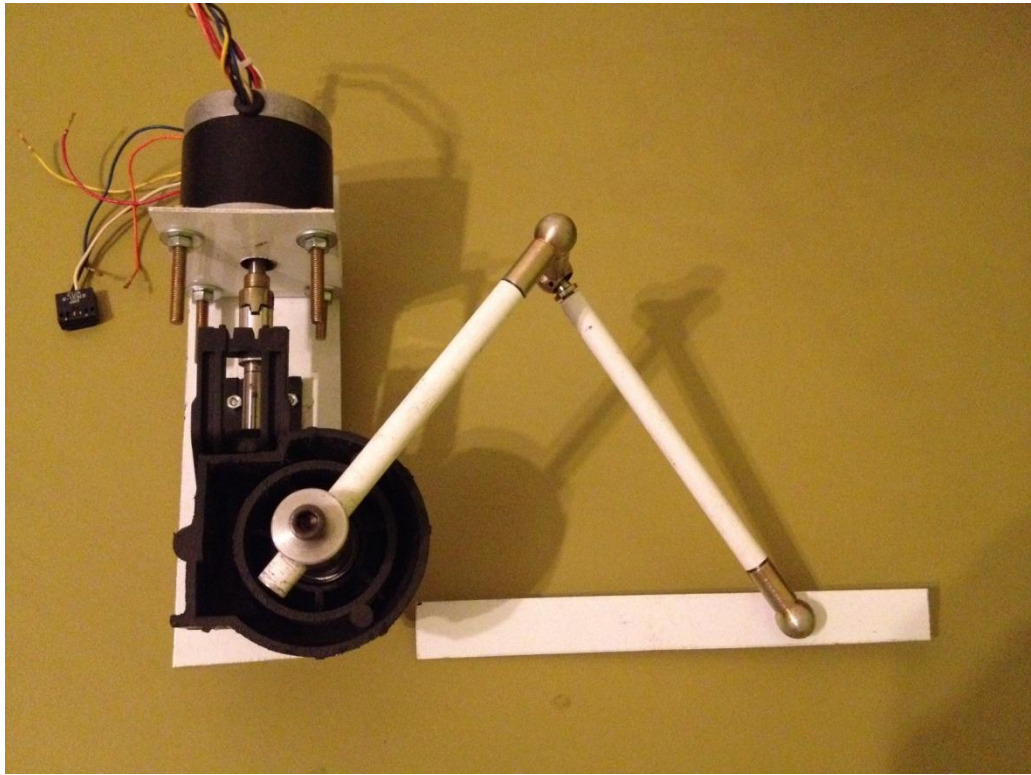


Figura 24 - Vista lateral do protótipo construído e montado

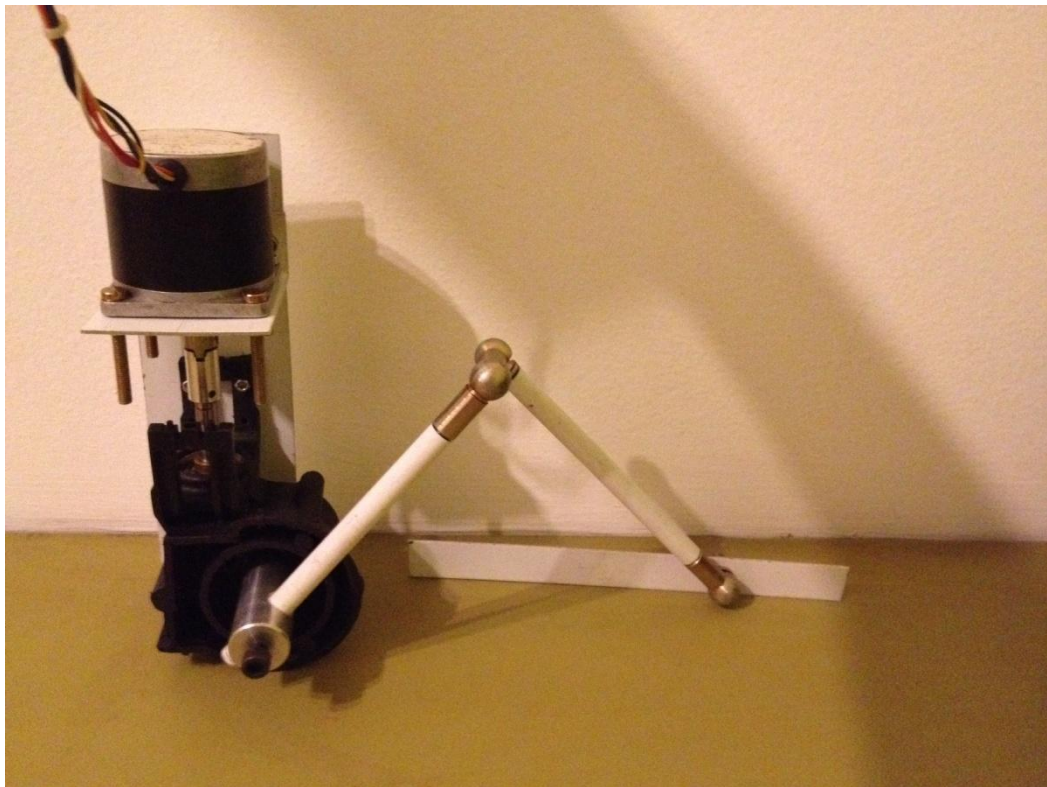


Figura 25 - Subsistema mecânico

Todas as peças do exoesqueleto foram construídas de acordo com a modelagem 3D em CAD. Desta forma, uma vez montado o protótipo, foi possível movimentar o mecanismo e acoplá-lo ao membro superior humano para a realização de testes funcionais. Paralelamente à finalização do subsistema mecânico, foi possível dar início ao projeto dos subsistemas eletrônicos e de programação, que foram integrados ao protótipo para o acionamento e automação do exoesqueleto. As próximas seções descrevem os projetos dos demais subsistemas deste trabalho.

4.2. PROJETO DO SUBSISTEMA ELETRÔNICO

De forma a garantir o controle do motor durante as sessões de tratamento e viabilizar a efetivação dos protocolos de comunicação, o subsistema eletrônico possui como componentes um driver de motor de passo Geckodrive G251, responsável pelo controle de pulsos ao motor, um microcontrolador ATmega238P e uma placa de inputs e outputs para facilitar as conexões do microcontrolador. A Figura 26 mostra os componentes principais do circuito eletrônico de acionamento do exoesqueleto

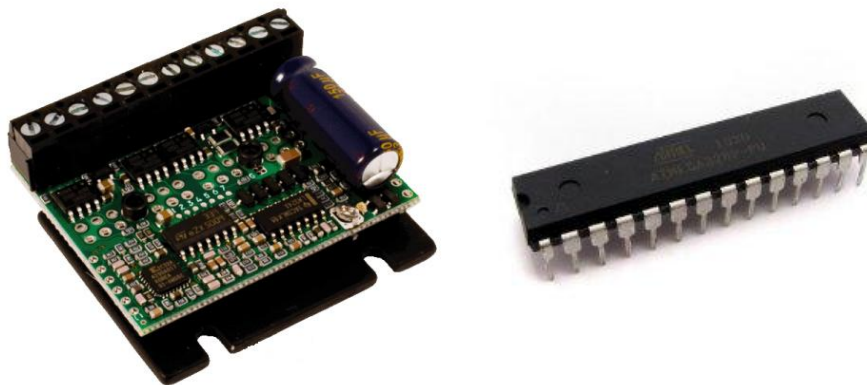


Figura 26 - (a) Driver para o motor de passo Geckodrive G251 e (b) microcontrolador ATmega328P

As especificações técnicas do driver podem ser encontradas no datasheet do Geckodrive G251, de onde é extraída a Figura 27.

GECKODRIVE G251

MAIN CONNECTOR:

- 1 GND
- 2 +VDC
- 3 I-SET
- 4 I-SET
- 5 PHASE A
- 6 PHASE /A
- 7 PHASE B
- 8 PHASE /B
- 9 DIRECTION
- 10 STEP
- 11 DISABLE
- 12 SIGNAL GND

CURRENT SET RESISTOR:

CONNECT 1/4W RESISTOR FROM 'I-SET' (TERM. 3) TO 'I-SET' (TERM. 4).
MOTOR PHASE CURRENT = 1 AMP PER 1,000 OHMS.

SPECIFICATIONS:

SUPPLY VOLTAGE	+15VDC MIN, +50VDC MAX
PHASE CURRENT	0A TO 3.5A
POWER DISSIPATION	3.3W AT 3.5A AND 50VDC
RESOLUTION	10 MICROSTEPS PER FULL STEP
INPUT LEVEL	0V TO 3.3V MIN, 0V TO 5V MAX
INPUT CURRENT	1mA
WEIGHT	0.45 OZ (12.5 GRAMS)
TEMPERATURE	0C TO 75C
STEP PULSE RATE	0Hz TO 300kHz
STEP PULSE WIDTH	1 MICROSECOND MIN
AUTO STANDBY	70% OF SET CURRENT

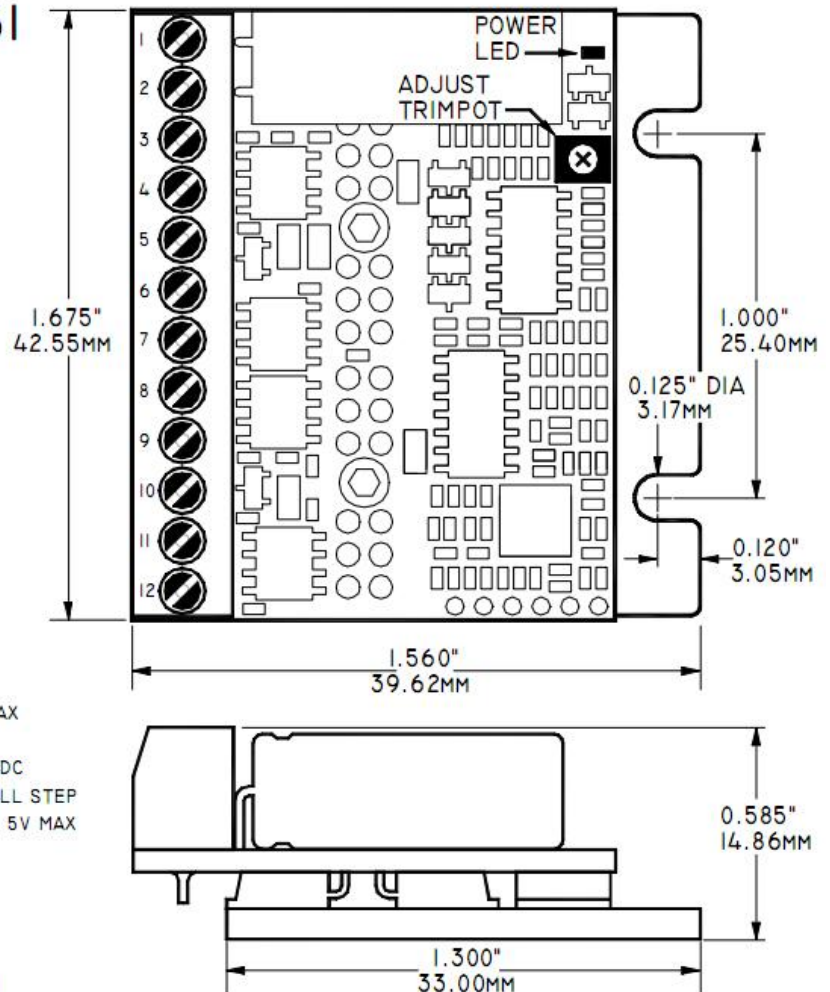


Figura 27 - Especificações técnicas do driver Geckodrive G251. Extraído do datasheet do componente.

O motor de passo utilizado possui tensão nominal de 15V, com 1,6A por fase e torque de 5 kfg.cm. O acionamento do exoesqueleto é utilizado com uma fonte de tensão e corrente variáveis, ajustada para alimentar corretamente o circuito.

O microcontrolador é alimentado com uma tensão de 5V fornecida pela interface Java através do cabo de conexão serial. Esta fonte de alimentação ao ATmega238P é independente da fonte de alimentação do motor de passo, que necessita de uma tensão

e corrente maior de trabalho. Os pulsos para o motor são gerados pelo driver, alimentado a partir de uma fonte de alimentação de 15V DC. O motor de passo utilizado é alimentado com 15V DC, possui 1,6A como amperagem máxima com 5kgf.cm de torque máximo.

4.3. PROJETO DO SUBSISTEMA DE PROGRAMAÇÃO

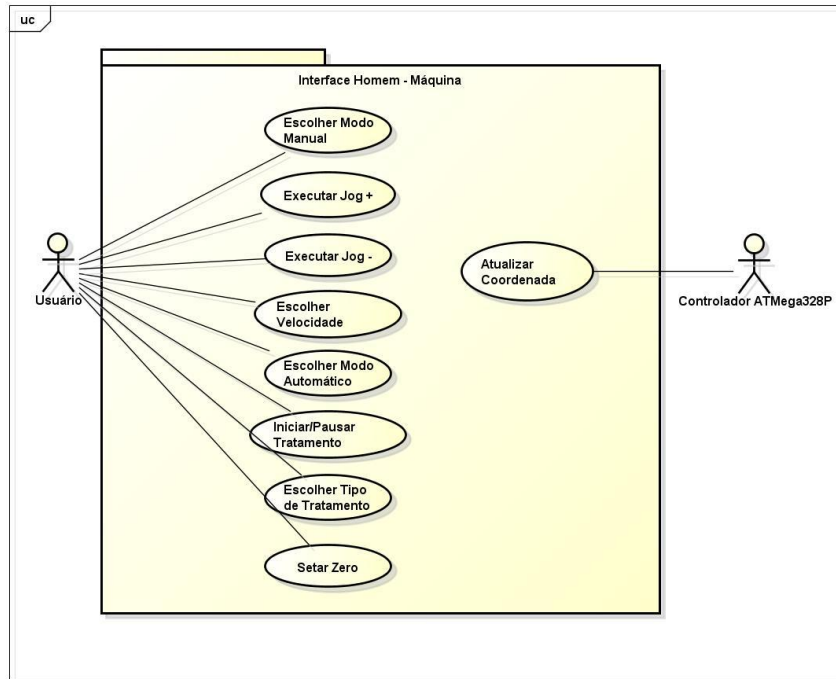
O subsistema de programação tem por objetivo integrar o subsistema eletrônico ao mecânico, realizando o acionamento do exoesqueleto remotamente, por meio de uma interface gráfica que comanda os dispositivos eletrônicos responsáveis pela movimentação do motor de passo.

Antes da implementação do software de projeto, foi necessário estudar as funcionalidades encontradas em órteses comumente utilizadas, e compreender como um programa de computador poderia auxiliar nos procedimentos de tratamento fisioterápicos dos pacientes. Estes resultados foram unidos aos conhecimentos de projetos mecatrônicos embarcados, provendo como solução final um software que fosse capaz de assegurar a robustez, repetibilidade, segurança e eficácia dos tratamentos. Uma documentação de projeto de software foi criada para auxiliar na sua implementação, e será descrita nesta seção.

O conjunto do software consiste em uma interface gráfica elaborada utilizando a linguagem Java e um microcontrolador ATmega328P utilizado no controle do motor de passo. Este microcontrolador foi escolhido devido à disponibilidade de códigos *open-source* na literatura, que auxiliam o desenvolvimento da programação. A comunicação entre a interface gráfica em Java e o microcontrolador se dá por intermédio de uma porta de comunicação USB-Serial utilizando o padrão RS-232. A implementação do protocolo de comunicação foi efetuada no ambiente Java e também no ambiente do ATmega328P, utilizando um protocolo de troca de mensagens que se encontra em fase de desenvolvimento.

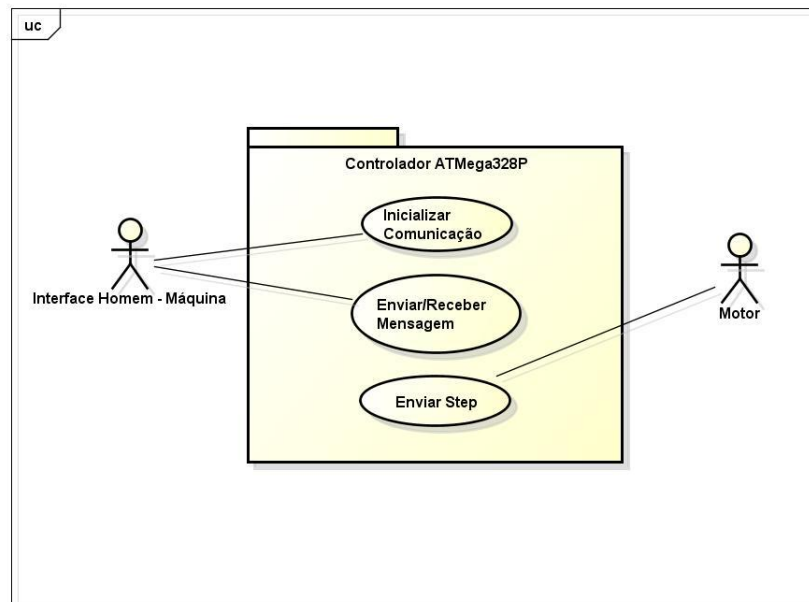
No intuito de melhor descrever a estrutura da programação, a Figura 28 e

Figura 29 ilustram os diagramas de casos da Interface Homem-Máquina e do Controlador ATmega328P.



powered by Astah

Figura 28 - Diagrama de casos de uso da interface Homem - Máquina



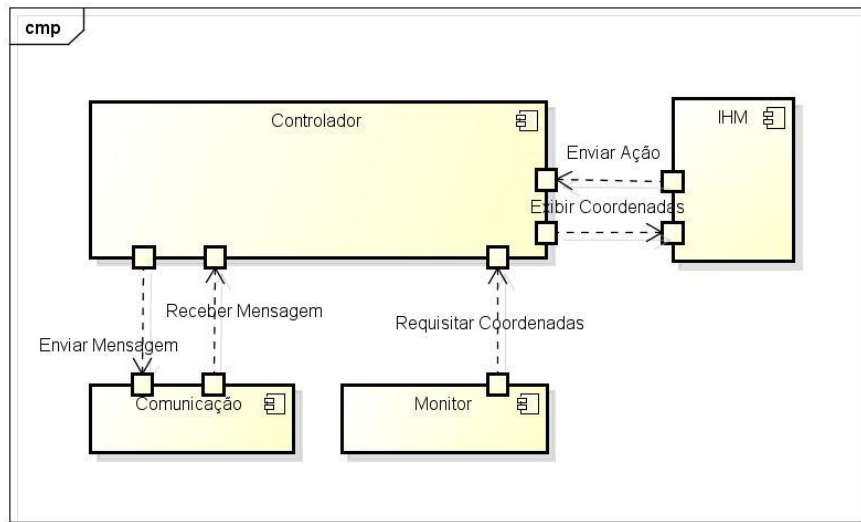
powered by Astah

Figura 29 - Diagrama de casos de uso do controlador ATmega328P

Da mesma forma, a estrutura de classes e funções dos códigos pode ser vista nos diagramas de componentes. A

Figura 30 mostra o diagrama de componentes da interface Homem - Máquina, com as classes e funções que relacionam o usuário do computador com a interface gráfica. Por outro lado, a

Figura 31 mostra o diagrama de componentes do controlador ATmega328P, que relaciona as classes e funções pertinentes à relação entre o controlador e o acionamento



do motor de passo.

Figura 30 - Diagrama de componentes da interface Homem - Máquina

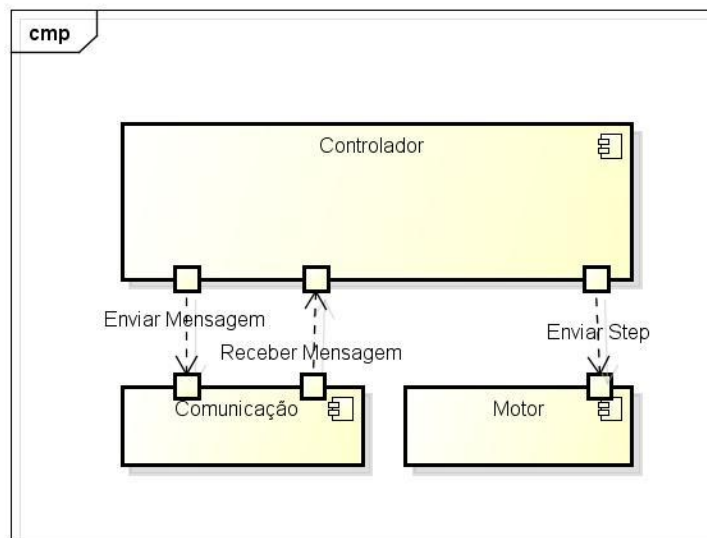


Figura 31 - Diagrama de componentes do controlador ATmega328P

4.3.1. INTERFACE HOMEM-MÁQUINA

Após realizado o projeto e documentação UML do subsistema de programação, o qual contemplou a análise dos requisitos e funcionalidades do programa, deu-se início à sua implementação. A Interface Homem Máquina (IHM) do protótipo consiste em um ambiente implementado utilizando a linguagem JAVA onde o usuário efetua as ordens de atuação do sistema. A Figura 32 ilustra os resultados atingidos, apresentando a proposta de IHM.

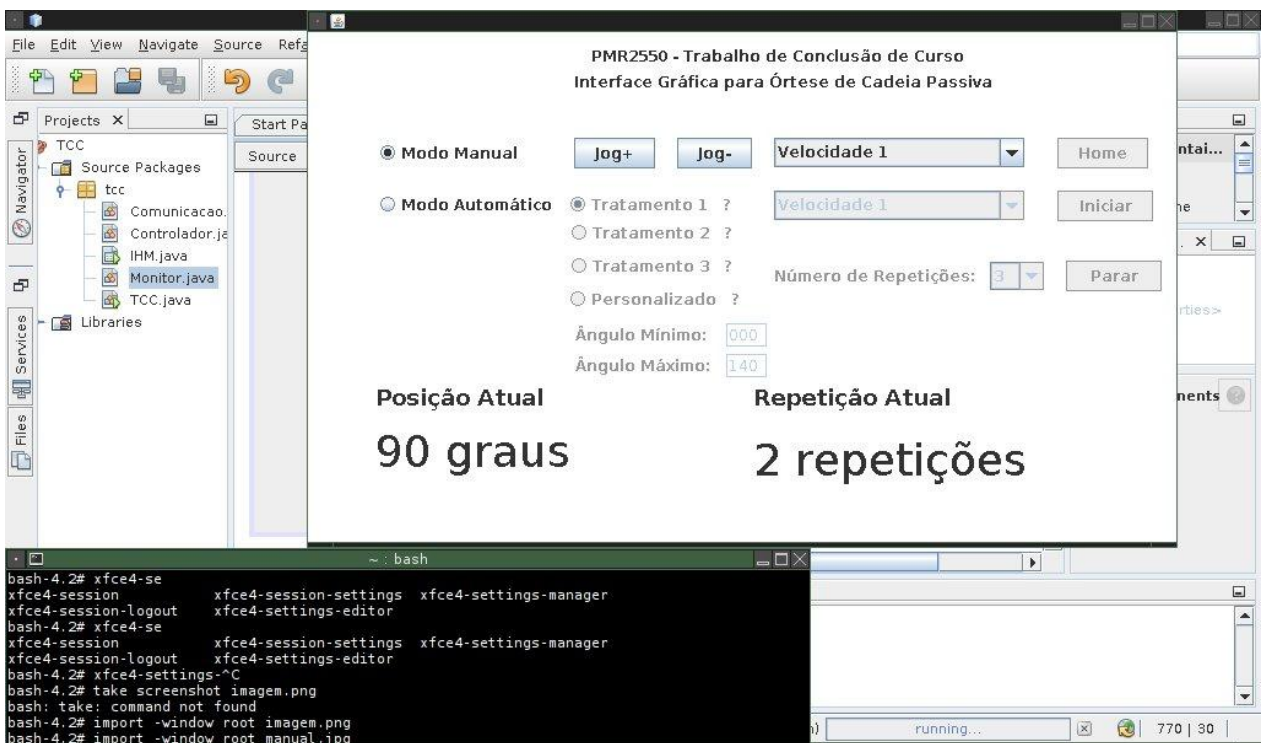


Figura 32 - Interface Homem Máquina exibindo a posição no modo “Manual”

Observa-se que a IHM possibilita a escolha de dois modos de operação do sistema, um modo chamado “Manual” e outro chamado “Automático”. No primeiro, o movimento da órtese é controlado a partir dos dois botões chamados “Jog+” e “Jog-”, cuja ativação envia sinais ao motor para efetuarem passos positivos ou negativos conforme o usuário assim o desejar. No segundo, o modo “Automático”, o usuário

escolhe um tipo de Tratamento (ângulos máximos e mínimos) e o número de repetições desejadas para a realização de um tratamento de fisioterapia. A Figura 33 ilustra a IHM trabalhando em modo “Automático”.

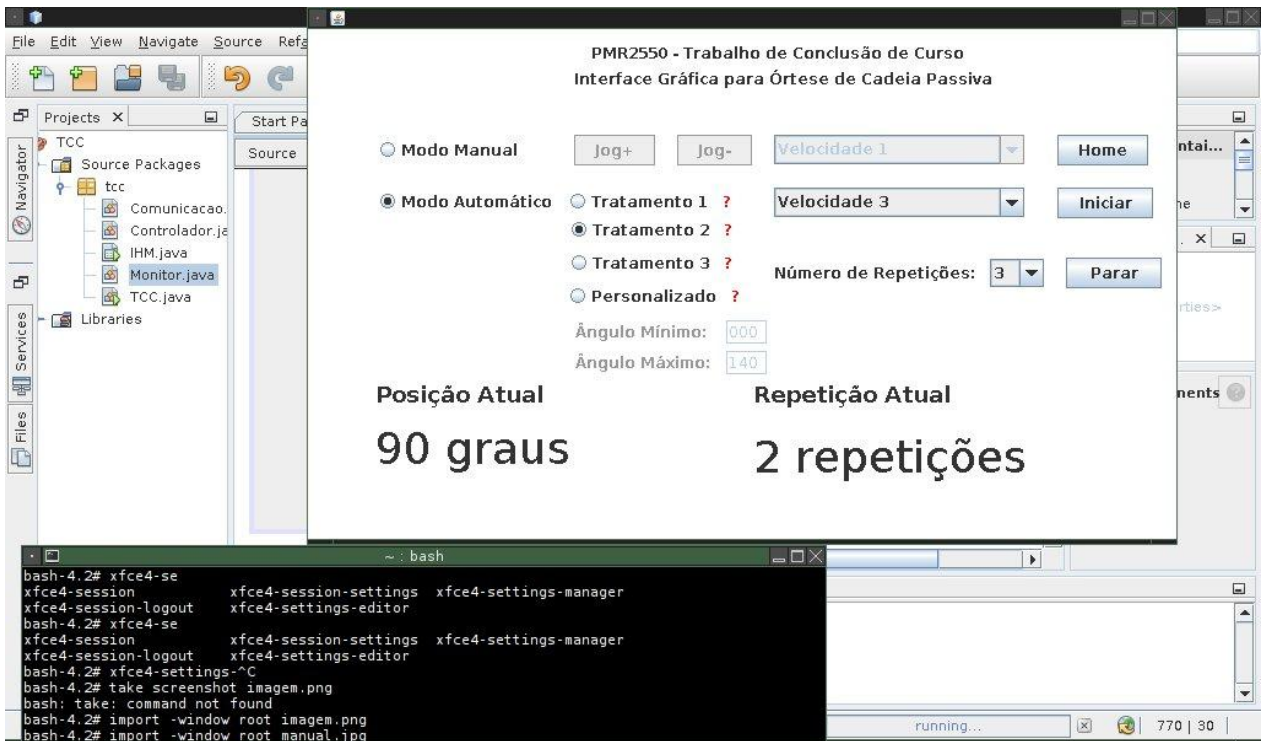


Figura 33 - Interface Homem Máquina trabalhando em modo “Automático”

A Interface JAVA possui cinco componentes cujas relações garantem o seu funcionamento, sendo eles as classes *TCC.java*, *Controlador.java*, *Comunicacao.java*, *IHM.java* e *Monitor.java*. Os códigos fonte que implementam estas classes são encontrados no Apêndice C.

A classe *TCC.java* é a classe principal que possui objetos de todas as demais e serve de base à execução ao programa.

A classe *Controlador.java* é a classe que possibilita o controle de todo o software, atuando e respondendo a requisições das outras classes. Do ponto de vista de projeto, escolheu-se por isolar cada componente de maneira que suas funções fossem exclusivas e organizadas.

A classe *Comunicacao.java* é a classe que abriga todo o código referente à comunicação entre o controlador e a IHM. Nela, os métodos de envio e recebimento de mensagens são implementados, proporcionando o uso do protocolo de comunicação desenvolvido entre as partes.

A classe *IHM.java* é a classe que exhibe na tela as ferramentas de uso ao usuário para o controle da órtese. Nela o usuário recebe informações das coordenadas da órtese e envia informações de configurações de velocidade ou atuação dos motores.

A classe *Monitor.java* desempenha um papel de comunicar com o controlador a cada instante de tempo perguntando pelas coordenadas atuais do sistema. Por padrão, esta comunicação é efetuada a cada 500ms, de modo a não sobrecarregar o protocolo de comunicação com as demais necessidades do sistema.

Em nível inferior, o controlador ATMEGA é responsável pela atuação direta ao driver do motor de passo, implementado em linguagem C++. Os códigos fonte desta implementação também são encontrados no Apêndice D. Seus componentes são uma função *loop*, responsável pelo gerenciamento dos passos do motor a partir de flags enviadas pela interface JAVA, uma função *serialEvent*, responsável pelo gerenciamento da comunicação com a criação dos comandos do protocolo de comunicação, uma função *tratarComando*, responsável pelo tratamento destes comandos e duas funções de transformações de coordenadas, que implementam a curva de movimentação do ângulo do antebraço da órtese em relação à posição angular do motor.

O protocolo de comunicação desenvolvido possibilita que a comunicação seja feita de ambos os lados com segurança. A função de cada um dos componentes da IHM e as strings que devem ser enviadas ao controlador são explicados a seguir:

Modo Manual: Seta o modo em Manual a partir do envio da string “:SETAR -MODO MANUAL!”;

JOG+: Efetua um jog positivo a partir do envio da string “:JOG+!”;

JOG-: Efetua um jog negativo a partir do envio da string “:JOG-!”;

Modo Automático: Seta o modo em Manual a partir do envio da string “:SETAR -MODO AUTOMATICO!”;

Velocidade: Seta a velocidade a partir do envio da string “:SETAR -VELOCIDADE V!”, em que V é o número de microsegundos entre dois passos consecutivos;

Tratamento: Seta o tipo de tratamento a ser executado a partir do envio da string “:SETAR -TRATAMENTO ANGMIN ANGMAX!”, em que ANGMIN é o ângulo inicial do antebraço e ANGMAX é o ângulo final do antebraço desejado no tratamento;

Ângulo Mínimo: Valor do ângulo mínimo desejado pelo usuário;

Ângulo Máximo: Valor do ângulo máximo desejado pelo usuário;

Número de Repetições: Seta o número de repetições desejado pelo usuário a partir do envio da string “:SETAR -REPETICOES N!”, em que N é o número de repetições desejado pelo usuário;

Home: Zera as coordenadas na posição atual a partir do envio da string “:HOME!”;

Iniciar: Inicia o tratamento automático a partir do envio da string “:INICIAR!”;

Pausar: Inicia o tratamento automático a partir do envio da string “:PAUSAR!”;

Parar: Inicia o tratamento automático a partir do envio da string “:PARAR!”;

Posição Atual: Exibe a posição angular do antebraço a partir do envio pelo monitor de uma string de requisição de coordenadas do tipo “:POS!”;

Repetição Atual: Exibe o número de repetições efetuadas a partir do envio pelo monitor de uma string de requisição de coordenadas do tipo “:POS!”;

4.3.1.CINEMÁTICA DIRETA E CONTROLE DO MECANISMO

Uma vez criada a interface Homem-Máquina que controlará remotamente o exoesqueleto, é necessário integrar no software uma transformação de coordenadas que relaciona a rotação do motor com o deslocamento angular do antebraço do paciente. Esta transformação de cinemática direta é essencial para efetuar o controle do mecanismo pelo software.

Utilizando o software Working Model, foi criado um modelo do exoesqueleto com as dimensões reais do protótipo. Além do modelo do protótipo, foi também incluído o modelo do quadrilátero articulado dimensionado na seção 3.5. O quadrilátero, no modelo em Working Model, representa a junta biológica humana, e foi acoplada com o exoesqueleto para que o deslocamento angular do antebraço possa ser estimado com maior precisão. A posição inicial e final da órtese pode ser vista na Figura 34. Na figura, as partes azuis representam as placas do exoesqueleto acopladas no braço e antebraço. As partes em laranja representam a cadeia ativa do exoesqueleto e as partes em cinza representam a região do membro superior humano próximo à junta biológica.

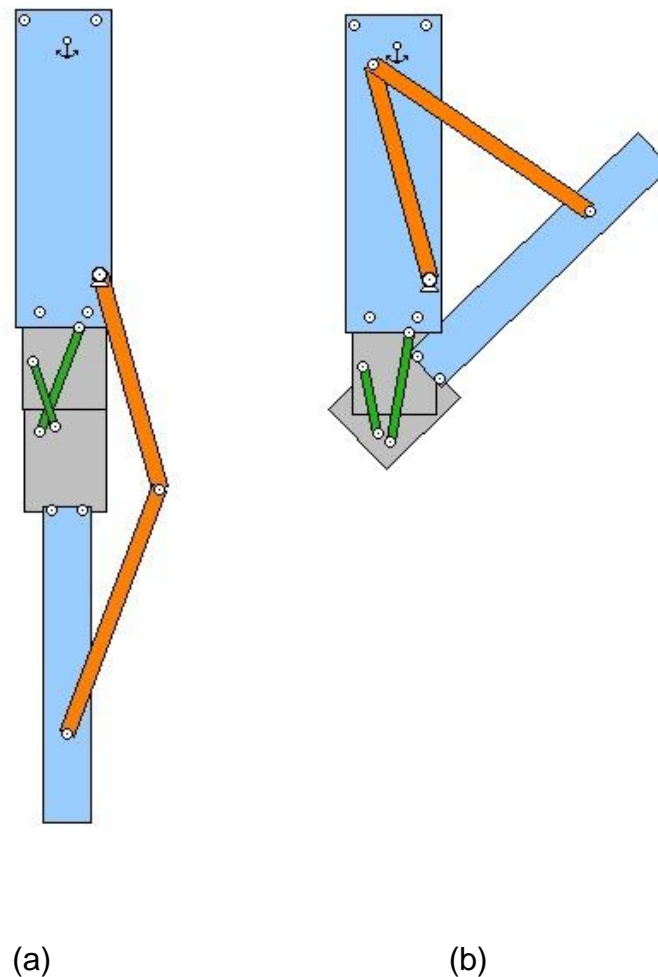


Figura 34- Modelo da órtese em Working Model com o quadrilátero articulado acoplado

Para gerar a equação de transformação de coordenadas, o modelo em Working Model foi submetido a um movimento na barra acionada pelo motor. O deslocamento angular do centro de massa da barra acionada e do centro de massa do antebraço foram então medidos, para que uma relação entre as variáveis pudesse ser criada.

Uma vez encontrada a relação entre os deslocamentos angulares, a rotação da barra acionada pode também ser relacionada com a rotação do motor, através do fator de redução da caixa de redução. Além disso, para o motor de passo utilizado, cada passo é responsável por um deslocamento angular de 1,8 graus. Assim, é possível equacionar o deslocamento angular do antebraço do exoesqueleto, tendo com entrada a quantidade de passos enviada ao motor. Finalmente, o driver Geckdrive G251 aciona o motor de passo através de micropassos, com uma resolução 10 vezes maior que a resolução do motor. Portanto, cada passo do motor de passo é equivalente a 10 pulsos do driver.

Os dados do movimento da simulação em Working Model foram exportados para uma planilha de dados, e foi possível gerar a relação de cinemática direta para o mecanismo. A relação entre as variáveis é ilustrada pelo gráfico da Figura 35. A equação de transformação de coordenadas é utilizada no software de controle do exoesqueleto. Desta forma, a cinemática direta permite o posicionamento do antebraço do paciente dada a quantidade de passos enviada ao driver G251.

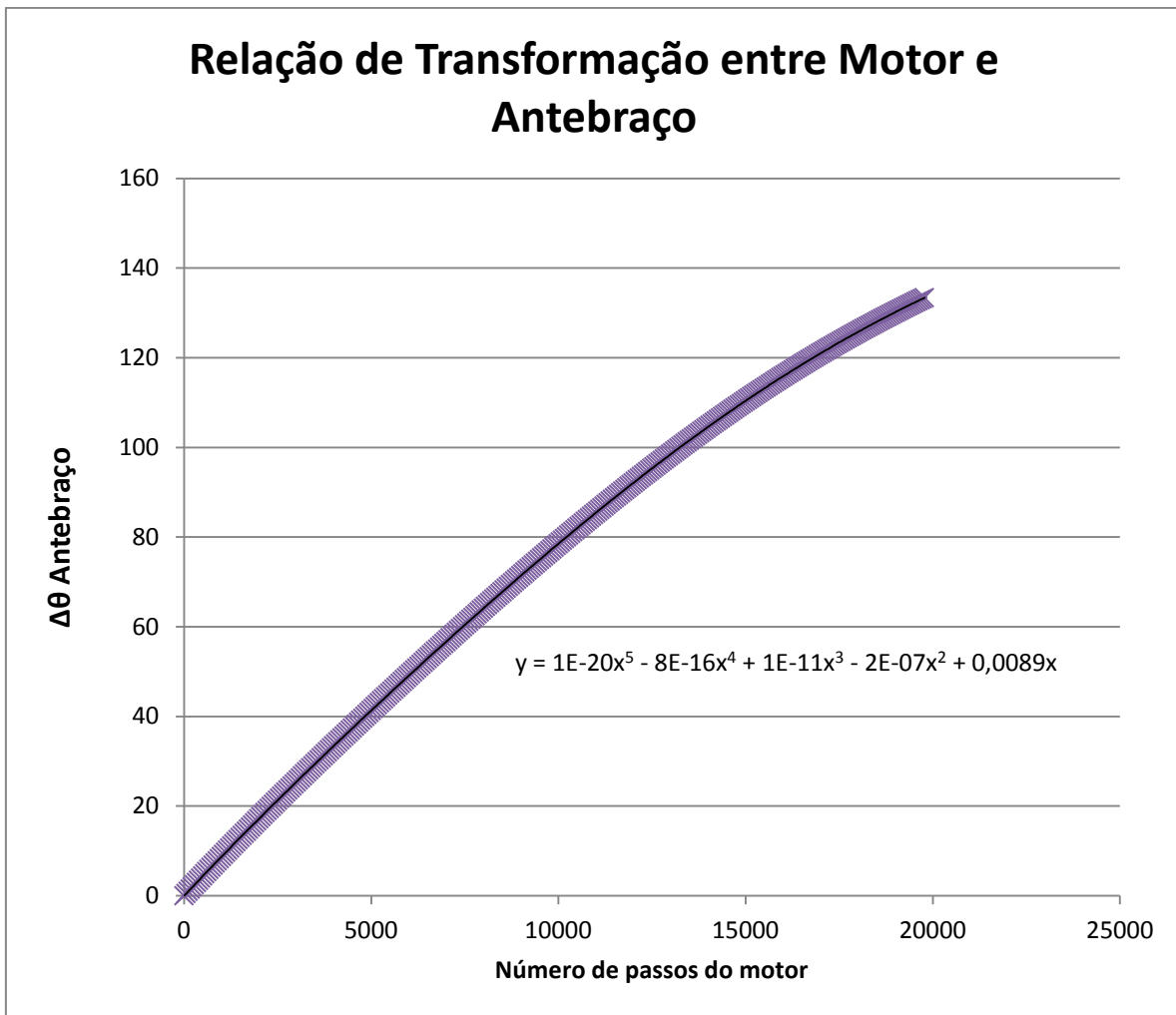


Figura 35 - Relação entre o deslocamento angular do centro de massa do antebraço e o número de passos do motor

Esta relação entre a quantidade de pulsos do driver e a rotação do antebraço é utilizada no software de controle do mecanismo, que mostra na IHM as coordenadas atuais da posição do antebraço em tempo real. Ao realizar os testes do exoesqueleto no corpo humano, foi possível confrontar os dados de predição da equação com os dados reais do experimento, como será visto na seção 4.4.

4.4. TESTES FUNCIONAIS

Após a elaboração dos três subsistemas do exoesqueleto, foi possível integrar o conjunto e realizar testes de validação do funcionamento. Para fixar a órtese no membro superior humano, foram utilizadas braçadeiras fixadas na estrutura do braço e antebraço do exoesqueleto. Dessa forma, foi possível testar o mecanismo ativo acoplado com o membro superior como a cadeia passiva. A Figura 36 ilustra o resultado final da integração dos subsistemas e a órtese acoplada em um membro superior humano.

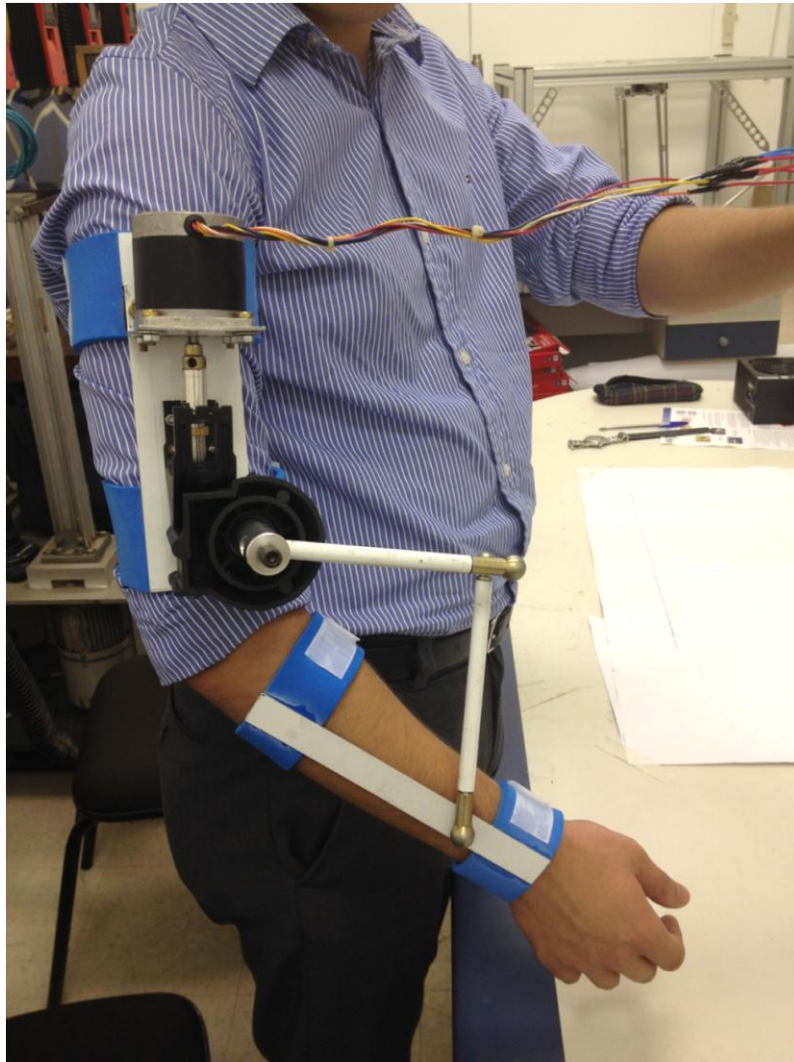
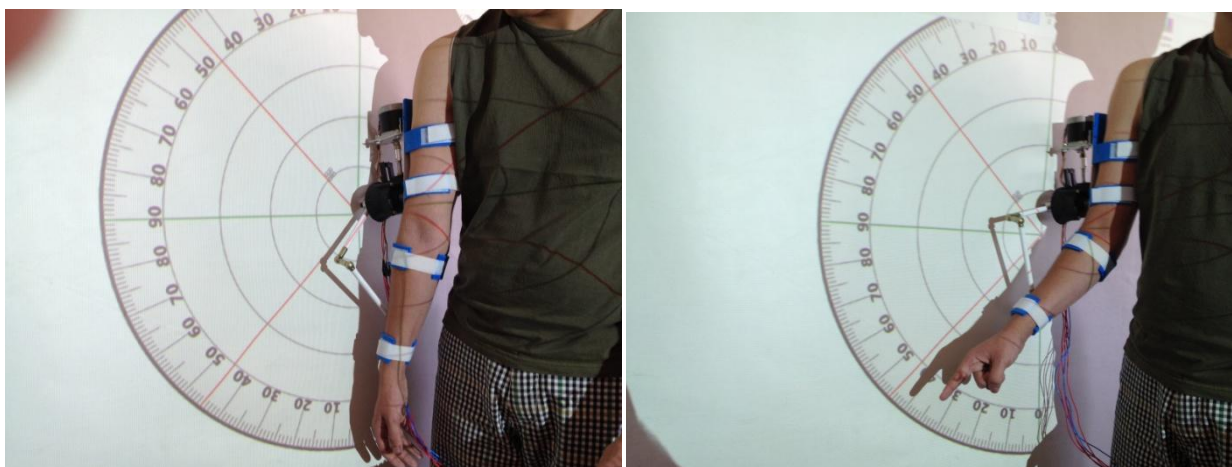


Figura 36 - Órtese acoplada ao membro superior

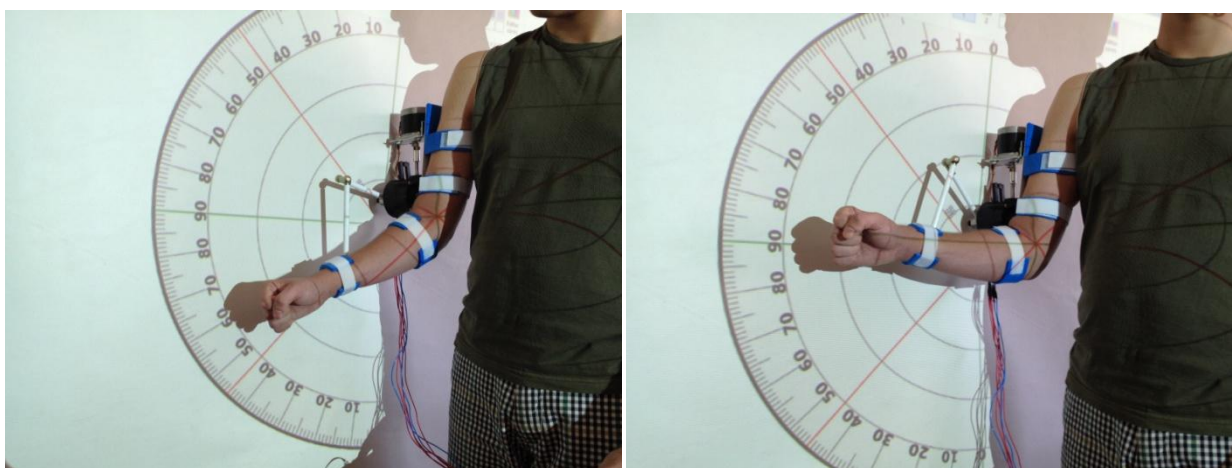
De forma a testar o controle e a cinemática direta do exoesqueleto, foi realizado um experimento de medição de deslocamento angular. A órtese foi acoplada a um membro superior, que por sua vez estava fixo a um sistema de medição angular. Ao acionar o dispositivo, o deslocamento angular do antebraço da estrutura, já acoplada com a cadeia passiva humana, pode ser medido, e confrontado com a indicação de deslocamento angular da IHM, provinda da equação da Figura 35. A (e)

Figura 37 mostra o resultado do experimento



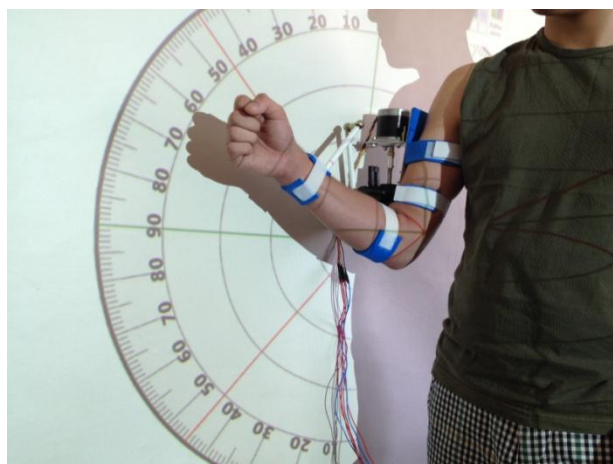
(a)

(b)



(c)

(d)



(e)

Figura 37 - Medição de deslocamento angular do dispositivo. A sequência de movimentação mostra o antebraço em 0 grau (a), 40 graus (b), 60 graus (c), 90 graus (d) e 120 graus (e).

Através deste experimento, foi possível avaliar qualitativamente o funcionamento do mecanismo. A precisão da modelagem não pôde ser avaliada devido às características simples do experimento. Melhores precisões poderiam ser obtidas por exemplo através da utilização de sensores que medem a posição angular do eixo do motor, como encoders ou potenciômetros.

No entanto, foi possível observar que o movimento da órtese respeita o movimento do membro superior, não gerando incômodos ou movimentos relativos entre as partes.

5. DISCUSSÃO

A construção do protótipo levou em consideração o conceito de mecanismo de cadeia passiva, que, quando acoplado ao membro superior humano, torna-se um exoesqueleto de mobilidade igual à mobilidade do membro superior, e portanto não gera esforços indesejáveis no pontos de fixação com o corpo humano. Isso se dá devido ao fato do exoesqueleto não utilizar uma junta rotacional simples, como é comumente visto nas órteses de reabilitação da literatura.

A utilização de um exoesqueleto de mobilidade exatamente igual ao membro superior humano soluciona um dos problemas levantados na introdução deste trabalho, no que diz respeito aos tratamentos de reabilitação do membro superior. Se corpo humano possui uma mobilidade diferente da mobilidade do exoesqueleto acoplado, existirão esforços nos pontos de fixação, devido aos desalinhamentos, que provocarão forças e torques que muitas vezes prejudicam o tratamento de reabilitação. Este trabalho propôs uma solução distinta, que garante a mobilidade exatamente igual entre o corpo humano e o exoesqueleto acoplado.

Além disso, no que diz respeito ao controle do mecanismo, foi preciso criar um modelo para a órtese e o membro superior humano acoplado, para que se pudesse relacionar a rotação do antebraço do paciente com a quantidade de pulsos necessários para o driver do motor alcançar este deslocamento angular. Para tal, modelou-se a junta Úmero-Rádio-Ulnar através do conceito de quadrilátero articulado, que propõe um dispositivo com variação do centro de rotação, semelhante à junta biológica. Tanto a junta biológica quanto o quadrilátero articulado se deslocam em torno de um eixo de rotação que varia com a posição do mecanismo. Desta forma, a modelagem por quadrilátero articulado foi proposta e analisada uma vez que o protótipo estava concluído.

Através dos resultados obtidos, foi possível verificar qualitativamente que o movimento da junta biológica do membro superior humano se desloca conforme previsto pelo software de controle. O experimento realizado com a órtese avalia o funcionamento e a precisão do controle, porém o experimento não possui grande precisão. Além disso, a modelagem do quadrilátero considerou um experimento de aquisição de pontos que também possui baixa precisão, o que poderia justificar a imprecisão do modelo real. Como um protótipo inicial, o funcionamento está de acordo com o esperado, porém sugere-se para trabalhos futuros, que os experimentos sejam refeitos, para que um controle mais preciso seja obtido. Estes pontos de melhoria são sugeridos como trabalhos futuros, juntamente com o refinamento do protótipo para se obter um dispositivo mais leve e rápido.

6. CONCLUSÕES

Nota-se que, embora um exoesqueleto de junta rotacional simples apresente maior simplicidade e menor custo do que este projeto, a órtese projetada garante a repetibilidade igual ao do membro superior, já que este faz o papel de cadeia passiva do mecanismo da órtese. O exoesqueleto proposto neste trabalho possui o diferencial de não gerar esforços indesejáveis nas fixações do dispositivo no membro superior do paciente. Isso se dá devido ao fato de a mobilidade do mecanismo ser exatamente igual à mobilidade do membro superior humano. Desta forma, este trabalho soluciona um dos problemas levantados na introdução deste projeto e portanto viabiliza sua utilização nas terapias de reabilitação.

O escopo deste trabalho não abrange o estudo comparativo entre as órteses de juntas rotacionais simples e a órtese desenvolvida neste projeto. Sugere-se, portanto, um trabalho futuro para estudar as vantagens e desvantagens do protótipo construído. Este trabalho, no entanto, constituiu no projeto de uma órtese de membro superior utilizando o conceito de quadrilátero articulado e também o estudo da modelagem da junta Úmero-Rádio-Ulnar por mecanismo de quadrilátero articulado.

Conforme visto no experimento realizado com o protótipo, conclui-se que o modelo adotado de quadrilátero articulado pode ser aplicado à junta biológica humana, porém deve ser modelado com maior precisão, para que se obtenha um modelo mais próximo do real. Ademais, o protótipo construído apresenta condições adequadas para ser utilizado em tratamentos de reabilitação humana, visto que sua arquitetura não gera esforços indesejáveis no membro superior do paciente. No entanto, algumas melhorias podem ser sugeridas para trabalhos futuros, como um refinamento do protótipo mecânico e uma modelagem mais precisa do quadrilátero articulado utilizado no controle do dispositivo.

7. REFERÊNCIAS BIBLIOGRÁFICAS

AMIGO, L. E.; FERNANDEZ, Q.; GIRALT, X.; CASALS, A.; AMAT, J., "Study of patient-orthosis interaction forces in rehabilitation therapies," *Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS & EMBS International Conference on*, vol., no., pp.1098,1103, 24-27 June 2012.

BENJAMIN, E. J. et al., Heart disease and stroke statistics – 2013 update, a report from the American Heart Association, *Circulation*, 2012.

DAVID, E. H.; KIELMAN, B. R.; HENSLEY, D. E., *Extension Deceleration Orthosis*. EP0439552 B1, 29 Mar 1995.

FILHO, M. A., 2006, *Derrame cerebral pode ser tratado, é o que neurologistas procuram mostrar - Jornal da Unicamp*. Disponível em: < http://www.unicamp.br/unicamp/unicamp_hoje/jornalPDF/ju323pg09.pdf >. Acesso em Junho, 2013.

COELHO, T. A. H, *Metodologia para Análise e Síntese de Mecanismos* - Material de apoio para PMR2430 - Universidade de São Paulo, Outubro, 2008.

COELHO, T. A. H, *Síntese de Mecanismos Planos de 4 Elos - Metodologia Analítica* - Material de apoio para PMR2430 - Universidade de São Paulo, 2009.

MANCINI M. C.; ALVES A. C. M.;SCHAPER C.; FIGUEIREDO E. M.; SAMPAIO R. F.; COELHO Z. A. C.; TIRADO M. G. A., "Gravidade da paralisia cerebral e desempenho funcional," *Revista Brasileira de Fisioterapia*, vol. 8, pp. 253 – 60, 2004.

MEDICAL ALLIANCE SA; NACE, R. A., *Knee Orthosis Swing Assist Mechanism*. US8376974 B2, 19 Feb 2013.

MIRANDA, A.B.W.; YASUTOMI, A.Y.; SOUIT, C.; FORNER-CORDERO, A., "Bioinspired mechanical design of an upper limb exoskeleton for rehabilitation and motor control assessment," *Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS & EMBS International Conference on*, vol., no., pp.1776,1781, 24-27 June 2012.

MYOMO, INC; MCBEAN, M. J.; NARENDRAN, N. K., *Powered Orthotic Device*. EP2079361 B1, 9 Jan 2013.

PONS, J.L., *Wearable Robots: Biomecatronic Exoskeletons*, Wiley & Sons, Ltd, ISBN 978-0-470-51294-4, 2008.

OLAYA A. F. R., *Sistema Robótico Multimodal para Análisis y Estudios en Biomecánica, Movimiento Humano y Control Neuromotor*. PhD thesis, University Carlos III, Madri, Spain, 2009.

PYLATIUK, C.; KARGOV, A.; GAISER, I.; WERNER, T.; SCHULZ, S.; BRETTHAUER, G., "Design of a flexible fluidic actuation system for a hybrid elbow orthosis," *Rehabilitation Robotics*, 2009. ICORR 2009. IEEE International Conference on , vol., no., pp.167,171, 23-26 June 2009.

RAGONESI, D.; RAHMAN, T.; SAMPLE, W.; AGRAWAL, S., "Dynamic simulation and experimental validation of an upper extremity powered orthosis," *Advanced Intelligent Mechatronics (AIM), 2010 IEEE/ASME International Conference on* , vol., no., pp.1,6, 6-9 July 2010.

SEKI, M.; MATSUMOTO, Y.; IIJIMA, H.; ANDO, T.; KOBAYASHI, Y.; FUJIE, M.G.; NAGAOKA, M., "Development of robotic upper limb orthosis with tremor suppressibility and elbow joint movability," *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on* , vol., no., pp.729,735, 9-12 Oct. 2011.

SICILIANO, B.; KHATIB, O. (eds.) *Springer Handbook of Robotics*, Springer, ISBN 978-3540239574, 2008.

TASHJIAN, RZ.; KATARINCIC, JA., *Complex Elbow Instability*, J Am Acad Orthop Surg, PP. 278-286, May 2006.

VANDERNIEPEN, I.; VAN HAM, R.; NAUDET, J.; VAN DAMME, M.; VANDERBORGHT, B.; VERSLUYS, R.; LEFEBER, D., "Novel Compliant Actuator for Safe and Ergonomic Rehabilitation Robots - Design of a Powered Elbow Orthosis," *Rehabilitation Robotics, 2007. ICORR 2007. IEEE 10th International Conference on* , vol., no., pp.790,797, 13-15 June 2007.

VANDERNIEPEN, I.; VAN HAM, R.; VAN DAMME, M.; LEFEBER, D., "Design of a powered elbow orthosis for orthopaedic rehabilitation using compliant actuation," *Bio-medical Robotics and Biomechatronics, 2008. BioRob 2008. 2nd IEEE RAS & EMBS International Conference on* , vol., no., pp.801,806, 19-22 Oct. 2008.

WORCESTER POLYTECHNIC INSTITUTE; HOFFMAN, A. H.; SCARSELLA, M. J.; TODDES, S. P.; ABRAMOVICH, D. N., *Two degree of freedom powered orthosis*. US8246559 B2, 21 Aug 2012.

APÊNDICE A - SIMULAÇÃO EM MATLAB

O Apêndice A apresenta o código em Matlab da simulação da síntese do mecanismo 4R utilizado neste trabalho.

```
%%%%%%%% TRABALHO DE CONCLUSÃO DE CURSO %%%%%%%%%
```

```
%%%%%%%% Curso de 140° do mecanismo %%%%%%%%%
```

```
%% Divide-se por 5 as coordenadas (em cm) dos pontos    %%  
%% para adaptar a ordem de grandeza do problema      %%
```

```
%% Pontos prescritos do experimento %%
```

```
P1 = (5.5 - 25.5i);  
P2 = (25.0 + 0i);  
P3 = (18.1 + 15.3i);
```

```
%% Centros O2 e O4 escolhidos %%
```

```
O2 = -2.00 +3i;  
O4 = 1 + 5i;
```

```
alpha2=78*pi/180;  
alpha3=117*pi/180;
```

```
%%%%%%%%%%%%%% LADO ESQUERDO %%%%%%%%%%%%%%%
```

```
v2 = -(P3-O2 - exp(1i*alpha3)*(P1-O2));  
v3 = P2-O2 - exp(1i*alpha2)*(P1-O2);  
v4 = -exp(1i*alpha2)*(P3-O2) + exp(1i*alpha3)*(P2-O2);  
modv2=abs(v2);  
modv3=abs(v3);  
modv4=abs(v4);
```

```
teta = acos((modv2^2 + modv4^2 - modv3^2)/(2*modv2*modv4));  
phi = acos((modv3^2 + modv4^2 - modv2^2)/(2*modv3*modv4));
```

```
%%%%%%%%%% Beta 2 e 3 %%%%%%%%%%%
```

```
beta2 = angle(v4) - angle(v2) + teta;  
beta3 = angle(v4) - angle(v3) - phi;
```

```
%%%%%%%%%% Beta 2' e 3' %%%%%%%%%%%
```

```
beta22 = angle(v4) - angle(v2) - teta;  
beta33 = angle(v4) - angle(v3) + phi;
```

```
M=[exp(1i*beta22)-1 , exp(1i*alpha2)-1; exp(1i*beta33)-1 , exp(1i*alpha3)-1];  
N=[P2-P1 ; P3-P1];  
esq = linsolve(M,N);
```

```

W=esq(1);
Z=esq(2);

A1=O2+W;
A2=O2+W*exp(1i*beta22);
A3=O2+W*exp(1i*beta33);

P11=A1+Z;
P22=A2+Z*exp(1i*alpha2);
P33=A3+Z*exp(1i*alpha3);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% LADO DIREITO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

u2 = -(P3-O4 - exp(1i*alpha3)*(P1-O4));
u3 = P2-O4 - exp(1i*alpha2)*(P1-O4);
u4 = -exp(1i*alpha2)*(P3-O4) + exp(1i*alpha3)*(P2-O4);
modu2=abs(u2);
modu3=abs(u3);
modu4=abs(u4);

teta2 = acos((modu2^2 + modu4^2 - modu3^2)/(2*modu2*modu4));
phi2 = acos((modu3^2 + modu4^2 - modu2^2)/(2*modu3*modu4));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Gama 2 e 3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
gama2 = angle(u4) - angle(u2) + teta2;
gama3 = angle(u4) - angle(u3) - phi2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Gama 2' e 3' %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
gama22 = angle(u4) - angle(u2) - teta2;
gama33 = angle(u4) - angle(u3) + phi2;

M=[exp(1i*gama22)-1 , exp(1i*alpha2)-1; exp(1i*gama33)-1 , exp(1i*alpha3)-1];
N=[P2-P1 ; P3-P1];
dir = linsolve(M,N);
U=dir(1);
S=dir(2);

B1=O4+U;
B2=O4+U*exp(1i*gama22);
B3=O4+U*exp(1i*gama33);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PLOTAGEM DO MECANISMO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

esquerdo=[real(O2) imag(O2) real(O2) imag(O2) real(O2) imag(O2) ;
          real(A1) imag(A1) real(A2) imag(A2) real(A3) imag(A3) ;
          real(P1) imag(P1) real(P2) imag(P2) real(P3) imag(P3) ];

direito= [real(O4) imag(O4) real(O4) imag(O4) real(O4) imag(O4) ;
          real(B1) imag(B1) real(B2) imag(B2) real(B3) imag(B3) ;
          real(P1) imag(P1) real(P2) imag(P2) real(P3) imag(P3) ];

hold all;

```

```

plot(real(O2), imag(O2), '*m', real(O4), imag(O4), '*y');

plot(esquerdo(:,1), esquerdo(:,2), 'ok', esquerdo(:,1), esquerdo(:,2), 'g-');
plot(direito(:,1), direito(:,2), 'ok', direito(:,1), direito(:,2), 'g-');

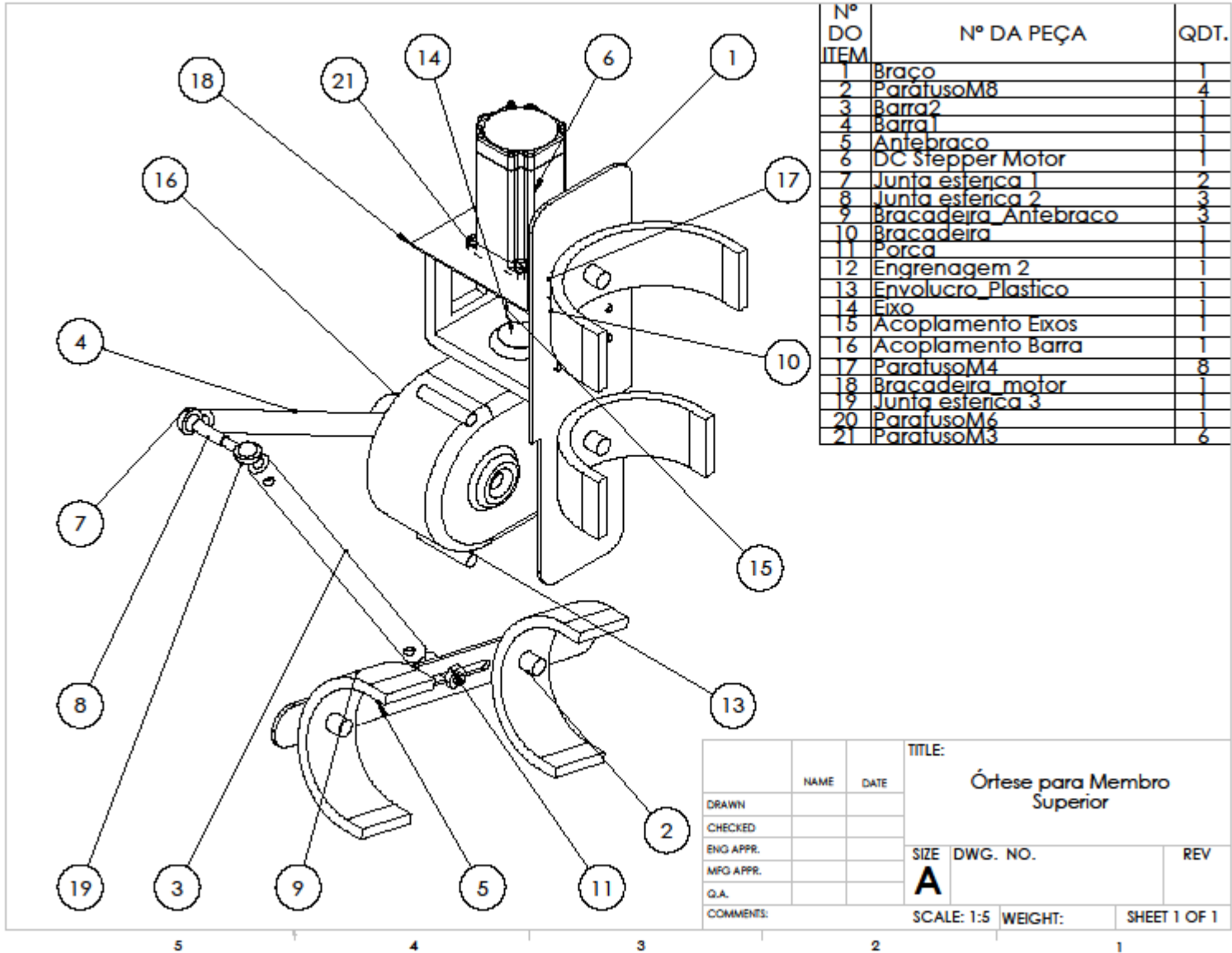
plot(esquerdo(:,3), esquerdo(:,4), 'ok', esquerdo(:,3), esquerdo(:,4), 'b-');
plot(direito(:,3), direito(:,4), 'ok', direito(:,3), direito(:,4), 'b-');

plot(esquerdo(:,5), esquerdo(:,6), 'ok', esquerdo(:,5), esquerdo(:,6), 'r-');
plot(direito(:,5), direito(:,6), 'ok', direito(:,5), direito(:,6), 'r-');

legend('O2', 'O4', '', 'Posição 1', '', 'Posição 1', '', 'Posição 2', '', 'Posição
2', '', 'Posição 3', '', 'Posição 3');
axis('equal');

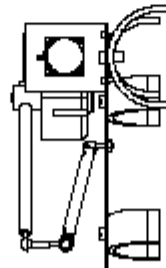
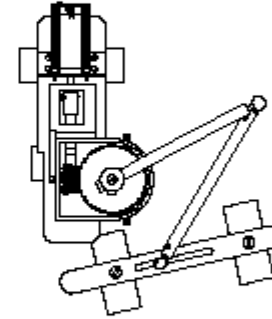
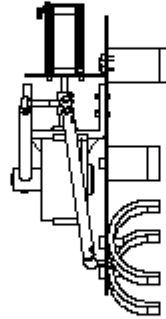
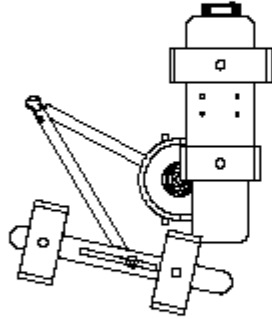
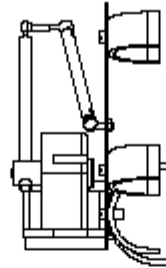
```

APÊNDICE B - DESENHOS DE CONJUNTO DO PROTÓTIPO



Nº DO ITEM	Nº DA PEÇA	QDT.
1	Braço	1
2	ParafusoM8	4
3	Barra2	1
4	Barra1	1
5	Antebraco	1
6	DC Stepper Motor	1
7	Junta esterica 1	2
8	Junta esterica 2	3
9	Bracadeira Antebraco	3
10	Bracadeira	1
11	Porca	1
12	Engrenagem 2	1
13	Envolvuro Plastico	1
14	Fixo	1
15	Acoplamento Eixos	1
16	Acoplamento Barra	1
17	ParafusoM4	8
18	Bracadeira motor	1
19	Junta esterica 3	1
20	ParafusoM6	1
21	ParafusoM3	6

NAME		DATE		TITLE:	
				Órtese para Membro Superior	
DRAWN		SIZE	DWG. NO.	REV	
CHECKED		A			
ENG APPR.		SCALE: 1:5	WEIGHT:	SHEET 1 OF 1	
MFG APPR.					
Q.A.					
COMMENTS:					



	NAME	DATE
DRAWN		
CHECKED		
ENG APPR.		
MFG APPR.		
Q.A.		

TITLE: Órtese - Vistas Laterais		
SIZE	DWG. NO.	REV
A		
SCALE: 1:5	WEIGHT:	SHEET 1 OF 1

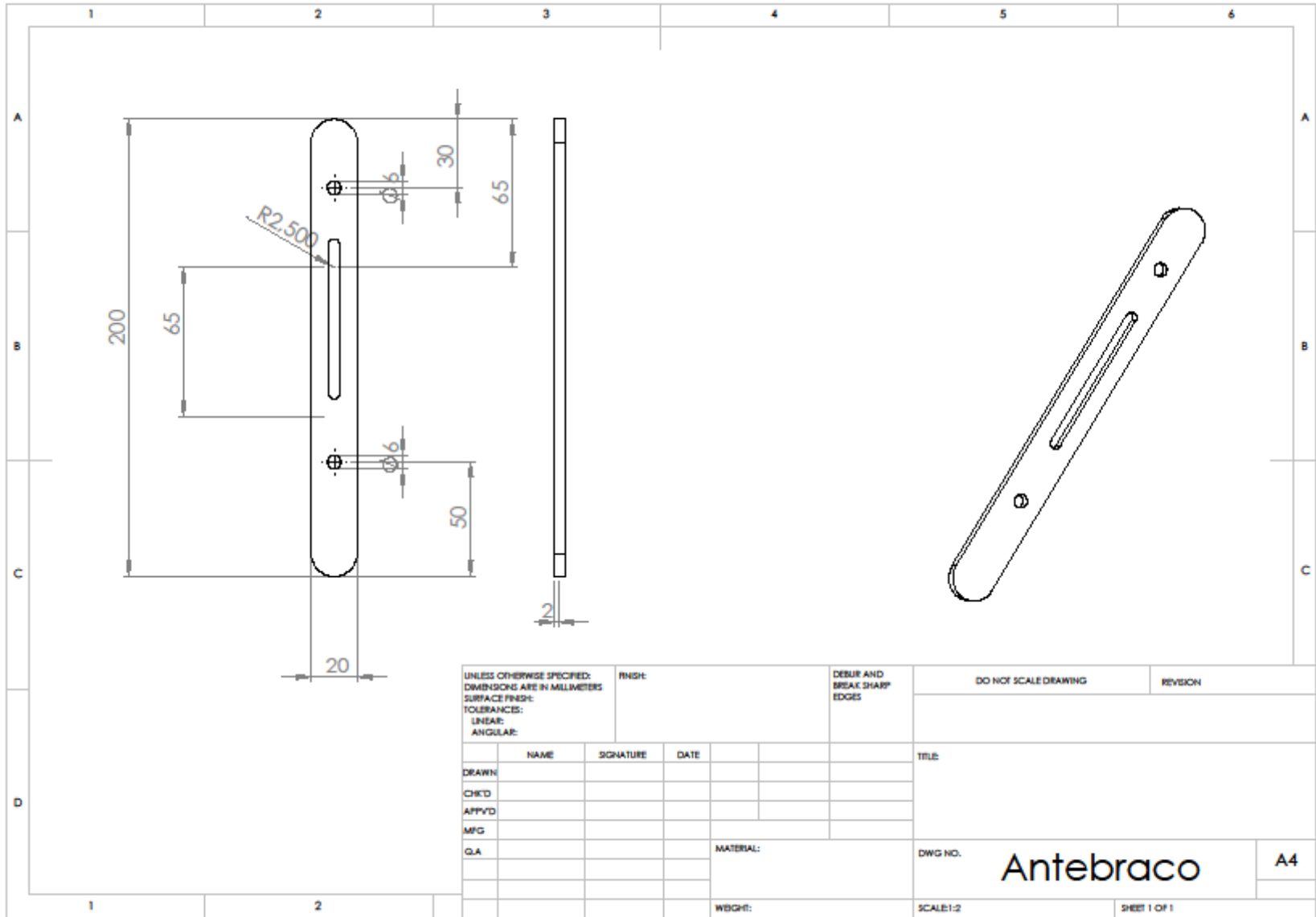
5

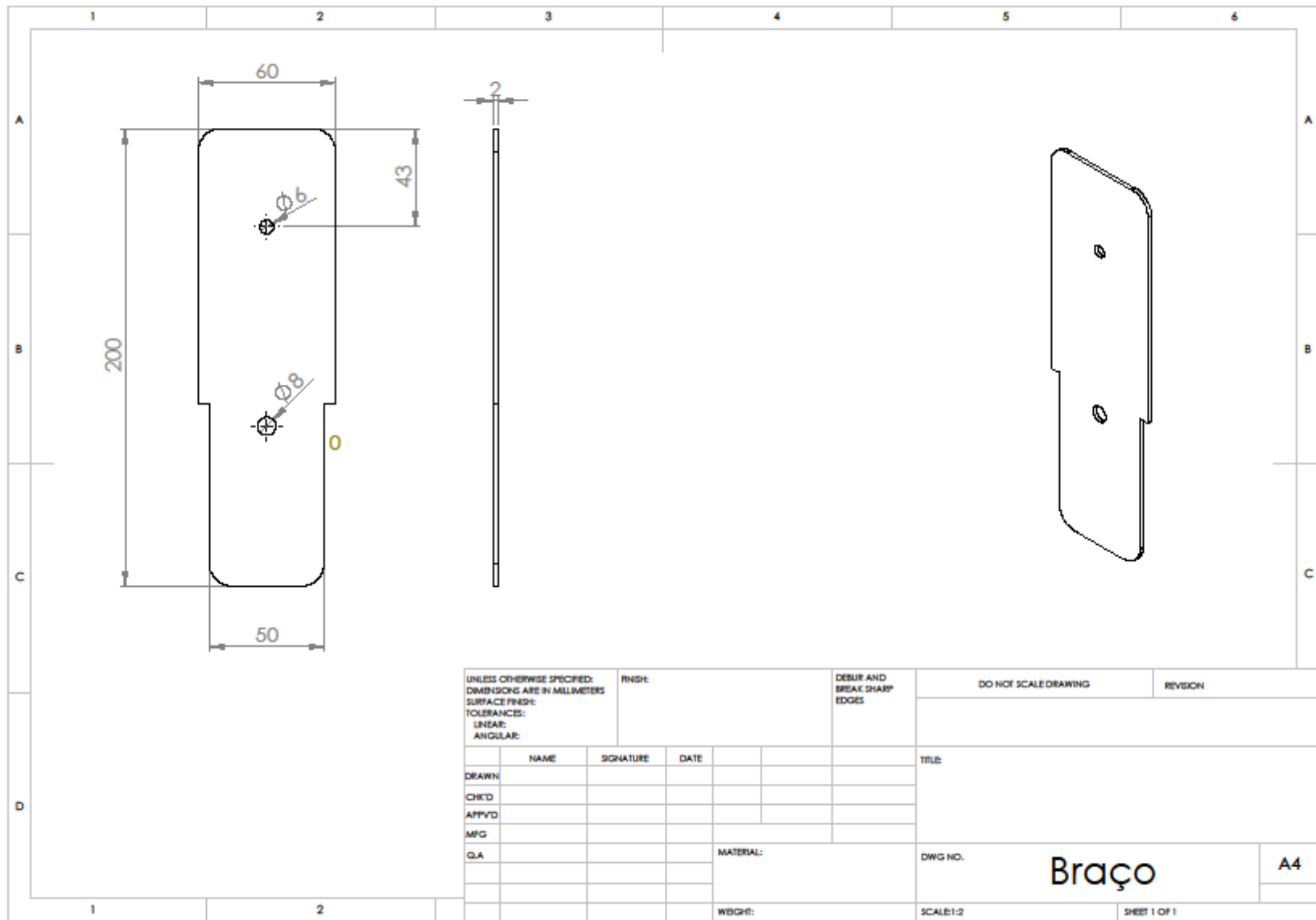
4

3

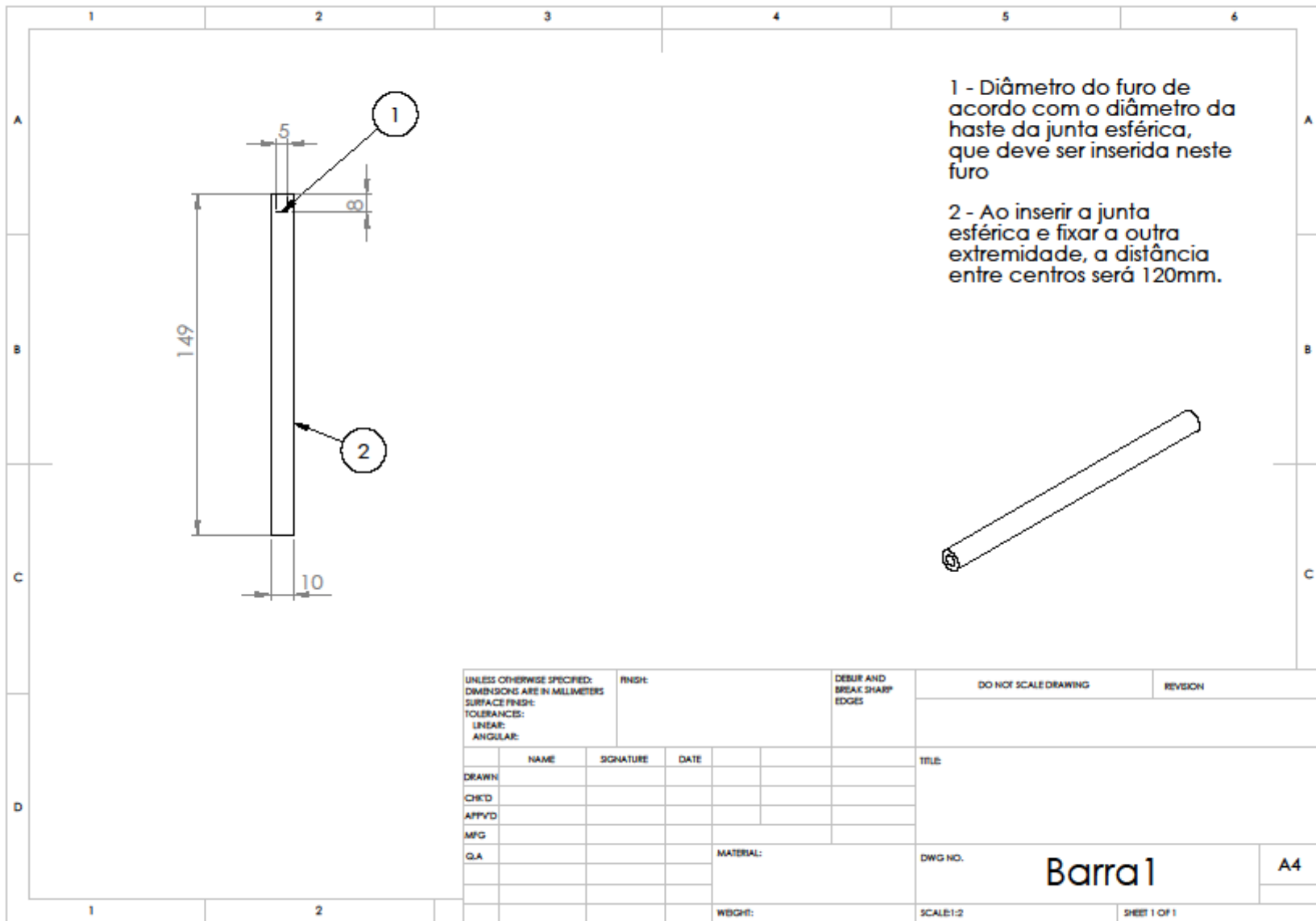
2

1

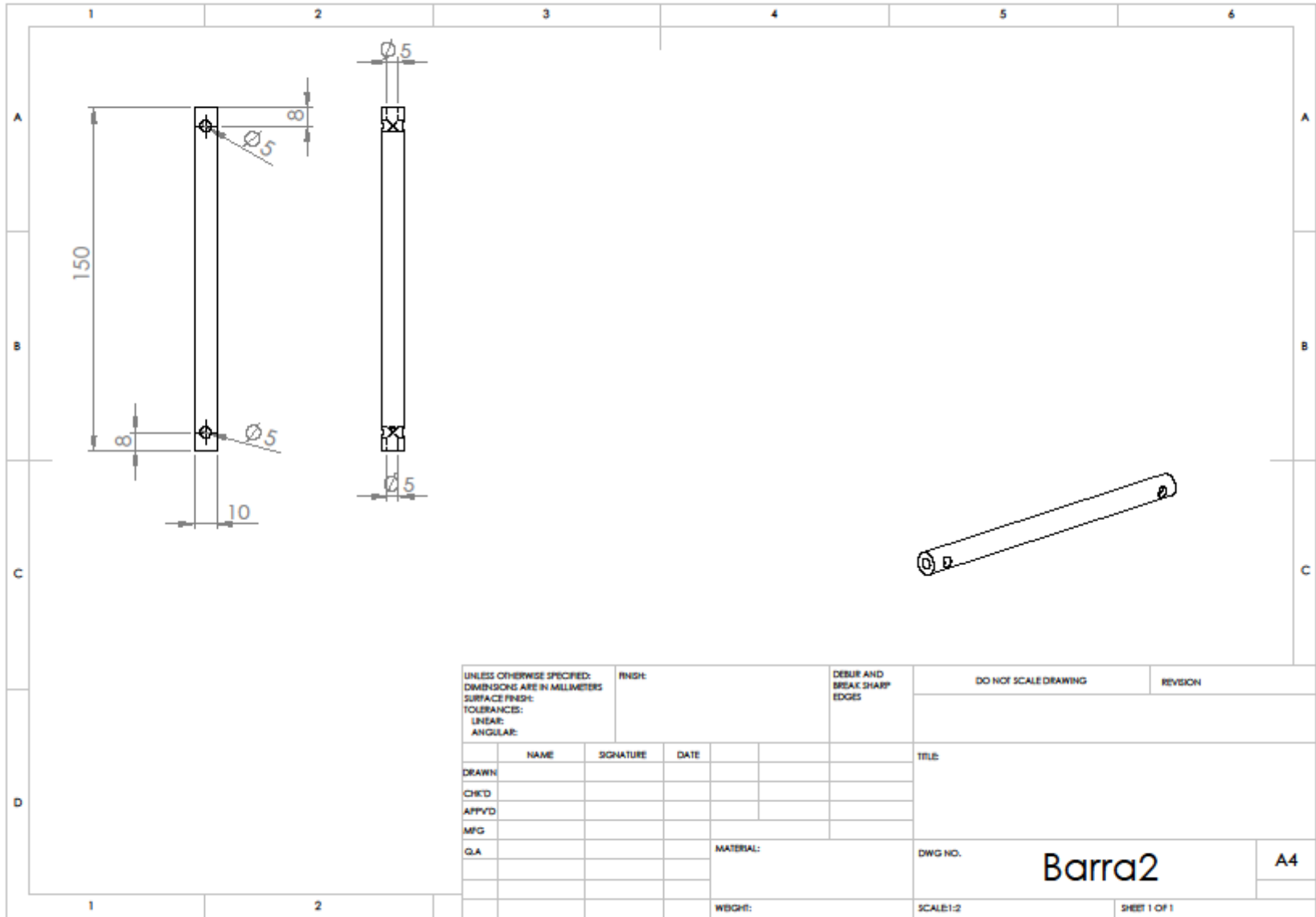


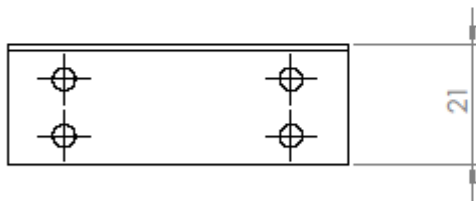
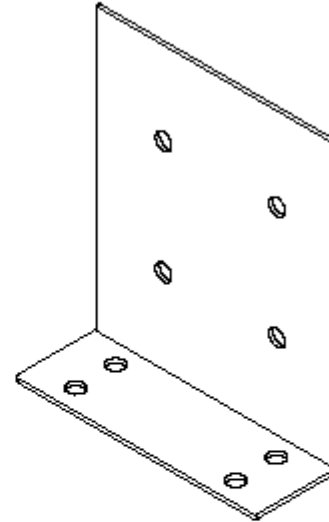
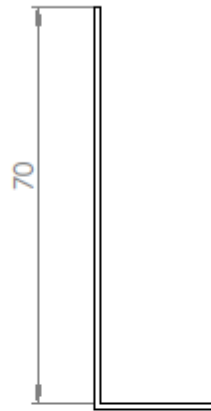
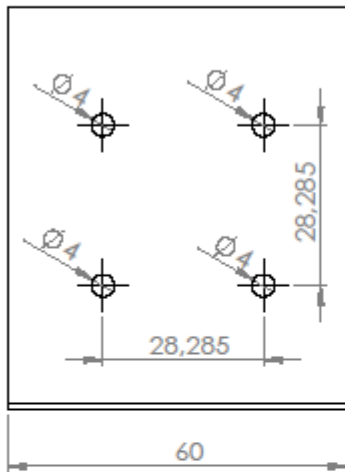


UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS				FINISH:		DEBUR AND BREAK SHARP EDGES		DO NOT SCALE DRAWING		REVISION	
SURFACE FINISH:											
TOLERANCES:											
LINEAR:											
ANGULAR:											
		NAME	SIGNATURE	DATE				TITLE			
DRAWN											
CHKD											
APPVD											
MFG											
Q.A						MATERIAL:		DWG NO.		A4	
								SCALE:1:2		SHEET 1 OF 1	
						WGHT:					



UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:				FINISH:		DEBUR AND BREAK SHARP EDGES		DO NOT SCALE DRAWING		REVISION	
										TITLE:	
DRAWN:				NAME		SIGNATURE		DATE			
CHKD:											
APPVD:											
MFG:											
Q.A:								MATERIAL:		DWG NO.	
										Barra1	
								WGHT:		A4	
								SCALE:1:2		SHEET 1 OF 1	





PROPRIETARY AND CONFIDENTIAL
 THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

		UNLESS OTHERWISE SPECIFIED:		NAME	DATE			
		DIMENSIONS ARE IN INCHES		DRAWN		TITLE:		
		TOLERANCES:		CHECKED				
		FRACTIONAL: ±		ENG APPR.				
		ANGULAR: MACH ± BEND ±		MFG APPR.				
		TWO PLACE DECIMAL ±		Q.A.				
		THREE PLACE DECIMAL ±		COMMENTS:		SIZE	DWG. NO.	REV
		INTERPRET GEOMETRIC TOLERANCING PER:				Brazadeira_motor		
		MATERIAL:				SCALE: 1:1	WEIGHT:	SHEET 1 OF 1
		FINISH:						
NEXT ASSY	USED ON	APPLICATION						
		DO NOT SCALE DRAWING						

5

4

3

2

1

APÊNDICE C - CÓDIGO PARA INTERFACE HOMEM-MÁQUINA

C1 - Comunicacao.java

```
package tcc;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.OutputStream;
import gnu.io.CommPortIdentifier;
import gnu.io.SerialPort;
import gnu.io.SerialPortEvent;
import gnu.io.SerialPortEventListener;
import java.util.Enumeration;

public class Comunicacao implements SerialPortEventListener {

    SerialPort serialPort;
    /** The port we're normally going to use. */
    private static final String PORT_NAMES[] = {
        "/dev/tty.usbserial-A9007UX1", // Mac OS X
        "/dev/ttyACM0", // Linux
        "COM3", // Windows
    };
    /**
     * A BufferedReader which will be fed by a InputStreamReader
     * converting the bytes into characters
     * making the displayed results codepage independent
     */
    private BufferedReader input;
    /** The output stream to the port */
    private OutputStream output;
    /** Milliseconds to block while waiting for port open */
    private static final int TIME_OUT = 2000;
    /** Default bits per second for COM port. */
    private static final int DATA_RATE = 9600;

    Controlador control = null;
    public void setControl(Controlador _control){
        control = _control;
    }

    public void initialize() {
```

```

        CommPortIdentifier portId = null;
        Enumeration portEnum =
CommPortIdentifier.getPortIdentifiers();

        //First, Find an instance of serial port as set in
PORT_NAMES.
        while (portEnum.hasMoreElements()) {
            CommPortIdentifier currPortId = (CommPortIdentifier)
portEnum.nextElement();
            for (String portName : PORT_NAMES) {
                if (currPortId.getName().equals(portName)) {
                    portId = currPortId;
                    break;
                }
            }
        }
        if (portId == null) {
            System.out.println("Could not find COM port.");
            return;
        }

        try {
            // open serial port, and use class name for the
appName.
            serialPort = (SerialPort)
portId.open(this.getClass().getName(),
                TIME_OUT);

            // set port parameters
            serialPort.setSerialPortParams(DATA_RATE,
                SerialPort.DATABITS_8,
                SerialPort.STOPBITS_1,
                SerialPort.PARITY_NONE);

            // open the streams
            input = new BufferedReader(new
InputStreamReader(serialPort.getInputStream()));
            output = serialPort.getOutputStream();

            // add event listeners
            serialPort.addEventListener(this);
            serialPort.notifyOnDataAvailable(true);
        } catch (Exception e) {
            System.err.println(e.toString());
        }
    }

/**
 * This should be called when you stop using the port.
 * This will prevent port locking on platforms like Linux.

```



```

    */
    public synchronized void close() {
        if (serialPort != null) {
            serialPort.removeEventListener();
            serialPort.close();
        }
    }

    /**
     * Handle an event on the serial port. Read the data and print
it.
     */
    public synchronized void serialEvent(SerialPortEvent oEvent) {
        if (oEvent.getEventType() == SerialPortEvent.DATA_AVAILABLE)
    {
        try {
            String inputLine=input.readLine();
            System.out.println("Recebi " + inputLine);
            if
(inputLine.substring(0,5).equals(":FIM!")){
                control.fim();
            }
            if
(inputLine.substring(0,9).equals(":RESPOSTA")){
                if (inputLine.charAt(11) == '@'){
                    con-
trol.atualizarPosicao(inputLine.substring(10,11));
                    if ((inputLine.charAt(13) ==
'!')){
                        con-
trol.atualizarRepeticao(inputLine.substring(12,13));
                    }
                    if ((inputLine.charAt(14) ==
'!')){
                        con-
trol.atualizarRepeticao(inputLine.substring(12,14));
                    }
                    if ((inputLine.charAt(15) ==
'!')){
                        con-
trol.atualizarRepeticao(inputLine.substring(12,15));
                    }
                }
                else if (inputLine.charAt(12) ==
'@'){
                    con-
trol.atualizarPosicao(inputLine.substring(10,12));
                    if ((inputLine.charAt(14) ==
'!')){

```

```

                                con-
trol.atualizarRepeticao(inputLine.substring(13,14));
                                }
                                if ((inputLine.charAt(15) ==
'!')){
                                con-
trol.atualizarRepeticao(inputLine.substring(13,15));
                                }
                                if ((inputLine.charAt(16) ==
'!')){
                                con-
trol.atualizarRepeticao(inputLine.substring(13,16));
                                }
                                }
                                else if (inputLine.charAt(13) ==
'@'){
                                con-
trol.atualizarPosicao(inputLine.substring(10,13));
                                if ((inputLine.charAt(15) ==
'!')){
                                con-
trol.atualizarRepeticao(inputLine.substring(14,15));
                                }
                                if ((inputLine.charAt(16) ==
'!')){
                                con-
trol.atualizarRepeticao(inputLine.substring(14,16));
                                }
                                if ((inputLine.charAt(17) ==
'!')){
                                con-
trol.atualizarRepeticao(inputLine.substring(14,17));
                                }
                                }
                                else if (inputLine.charAt(14) ==
'@'){
                                con-
trol.atualizarPosicao(inputLine.substring(10,14));
                                if ((inputLine.charAt(16) ==
'!')){
                                con-
trol.atualizarRepeticao(inputLine.substring(15,16));
                                }
                                if ((inputLine.charAt(17) ==
'!')){
                                con-
trol.atualizarRepeticao(inputLine.substring(15,17));
                                }
                                if ((inputLine.charAt(18) ==
'!')){

```

```

                                con-
trol.atualizarRepeticao(inputLine.substring(15,18));
                                }
                                }
                                } catch (Exception e) {
                                //System.err.println(e.toString());
                                }
                                }
                                // Ignore all the other eventTypes, but you should consider
the other ones.
                                }

public void send(String msg) {
    try {
        output.write(msg.getBytes());
    }
    catch(Exception ex)
    {
    }
}

}

```

C2 - Controlador.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package tcc;

/**
 *
 * @author pc
 */
public class Controlador {

    private IHM ihm = null;
    private Comunicacao com = null;

    public void setIHM(IHM _ihm){
        ihm = _ihm;
    }

    public void setComm(Comunicacao _com){
        com = _com;
    }

    public void send(String msg){
        com.send(msg);
    }

    public void atualizarPosicao(String posicao){
        ihm.atualizarPosicao(posicao);
    }

    public void atualizarRepeticao(String repeticao){
        ihm.atualizarRepeticao(repeticao);
    }

    public void fim(){
        ihm.fim();
    }

}
```

C3 - Monitor.java

```
package tcc;

public class Monitor extends Thread {

    private Controlador control = null;

    public void setControl(Controlador _control){
        control = _control;
    }

    public void initialize() {
        while (true) {
            try {
                control.send(":POS!");
                try { sleep(1000); } catch(Exception e) {}

            } catch(Exception e) {
                // e.printStackTrace();
            }
        }
    }
}
```

C4 - TCC.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package tcc;

import java.awt.Color;

/**
 *
 * @author pc
 */
public class TCC {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {

        Comunicacao com = new Comunicacao();
        IHM ihm = new IHM();
        Controlador control = new Controlador();
        Monitor monitor = new Monitor();

        control.setIHM(ihm);
        control.setComm(com);

        ihm.setControl(control);
        com.setControl(control);
        monitor.setControl(control);

        com.initialize();

        ihm.setVisible(true);

        monitor.initialize();
    }
}
```

C5 - IHM.java

```
package tcc;

import java.awt.Color;

/**
 *
 * @author pc
 */
public class IHM extends javax.swing.JFrame {

    private Controlador control = null;

    public IHM() {
        initComponents();
        this.getContentPane().setBackground(Color.WHITE);
        jDialog1.setSize(300, 200);
        jDialog1.setUndecorated(true);
        jDialog2.setSize(300, 200);
        jDialog2.setUndecorated(true);
        jDialog3.setSize(300, 200);
        jDialog3.setUndecorated(true);
        jDialog4.setSize(300, 200);
        jDialog4.setUndecorated(true);
        //jButton8.setVisible(false);
        //jButton7.setVisible(false);
    }

    public void setControl(Controlador _control){
        control = _control;
    }

    public void atualizarPosicao(String posicao){
        jLabel6.setText(posicao + " graus");
    }

    public void atualizarRepeticao(String repeticao){
        //if (jButton6.getText().equals("Iniciar")){
        //jLabel8.setText(repeticao + " repetiÃ§Ãµes");
        //}
        //else
        jLabel8.setText(String.valueOf(Integer.parseInt(repeticao)-0) +
" repetiÃ§Ãµes");
    }
}
```

```

public void fim(){
jButton6.setText("Iniciar");
}

/**
 * This method is called from within the constructor to initialize
the form.
 * WARNING: Do NOT modify this code. The content of this method is
always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated
Code">//GEN-BEGIN: initComponents
private void initComponents() {

    jDialog1 = new javax.swing.JDialog();
    jLabel11 = new javax.swing.JLabel();
    jDialog2 = new javax.swing.JDialog();
    jLabel12 = new javax.swing.JLabel();
    jDialog3 = new javax.swing.JDialog();
    jLabel13 = new javax.swing.JLabel();
    jDialog4 = new javax.swing.JDialog();
    jLabel14 = new javax.swing.JLabel();
    jRadioButton1 = new javax.swing.JRadioButton();
    jRadioButton2 = new javax.swing.JRadioButton();
    jLabel5 = new javax.swing.JLabel();
    jLabel6 = new javax.swing.JLabel();
    jLabel7 = new javax.swing.JLabel();
    jLabel8 = new javax.swing.JLabel();
    jComboBox1 = new javax.swing.JComboBox();
    jButton4 = new javax.swing.JButton();
    jButton5 = new javax.swing.JButton();
    jRadioButton3 = new javax.swing.JRadioButton();
    jRadioButton4 = new javax.swing.JRadioButton();
    jRadioButton5 = new javax.swing.JRadioButton();
    jRadioButton6 = new javax.swing.JRadioButton();
    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jLabel9 = new javax.swing.JLabel();
    jLabel10 = new javax.swing.JLabel();
    jTextField1 = new javax.swing.JTextField();
    jTextField2 = new javax.swing.JTextField();
    jButton6 = new javax.swing.JButton();
    jComboBox2 = new javax.swing.JComboBox();
    jButton7 = new javax.swing.JButton();
    jLabel15 = new javax.swing.JLabel();
    jButton8 = new javax.swing.JButton();

```



```

jComboBox3 = new javax.swing.JComboBox();
jLabel17 = new javax.swing.JLabel();
jLabel16 = new javax.swing.JLabel();

jDialog1.setDefaultCloseOperation(javax.swing.WindowConstants.DO_NOTHING_ON_CLOSE);
jDialog1.setType(java.awt.Window.Type.POPUP);

jLabel11.setText("Ilustração do Tratamento 1");

javax.swing.GroupLayout jDialog1Layout = new
javax.swing.GroupLayout(jDialog1.getContentPane());
jDialog1.getContentPane().setLayout(jDialog1Layout);
jDialog1Layout.setHorizontalGroup(

jDialog1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jDialog1Layout.createSequentialGroup()
        .addGap(70, 70, 70)
        .addComponent(jLabel11)
        .addGap(171, Short.MAX_VALUE))
    );
jDialog1Layout.setVerticalGroup(

jDialog1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jDialog1Layout.createSequentialGroup()
        .addGap(57, 57, 57)
        .addComponent(jLabel11)
        .addGap(124, Short.MAX_VALUE))
    );
jLabel12.setText("Ilustração do Tratamento 2");

javax.swing.GroupLayout jDialog2Layout = new
javax.swing.GroupLayout(jDialog2.getContentPane());
jDialog2.getContentPane().setLayout(jDialog2Layout);
jDialog2Layout.setHorizontalGroup(

jDialog2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jDialog2Layout.createSequentialGroup()
        .addGap(65, 65, 65)
        .addComponent(jLabel12)
        .addGap(219, Short.MAX_VALUE))
    );
jDialog2Layout.setVerticalGroup(

```

```

jDialog2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jDialog2Layout.createSequentialGroup()
        .addGap(72, 72, 72)
        .addComponent(jLabel12)
        .addContainerGap(214, Short.MAX_VALUE)
    );
jLabel13.setText("Ilustração do Tratamento 3");

javax.swing.GroupLayout jDialog3Layout = new
javax.swing.GroupLayout(jDialog3.getContentPane());
jDialog3.getContentPane().setLayout(jDialog3Layout);
jDialog3Layout.setHorizontalGroup(

jDialog3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jDialog3Layout.createSequentialGroup()
        .addGap(66, 66, 66)
        .addComponent(jLabel13)
        .addContainerGap(218, Short.MAX_VALUE)
    );
jDialog3Layout.setVerticalGroup(

jDialog3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jDialog3Layout.createSequentialGroup()
        .addGap(81, 81, 81)
        .addComponent(jLabel13)
        .addContainerGap(205, Short.MAX_VALUE)
    );
jLabel14.setText("Ilustração do Tratamento Personalizado");

javax.swing.GroupLayout jDialog4Layout = new
javax.swing.GroupLayout(jDialog4.getContentPane());
jDialog4.getContentPane().setLayout(jDialog4Layout);
jDialog4Layout.setHorizontalGroup(

jDialog4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jDialog4Layout.createSequentialGroup()
        .addGap(53, 53, 53)
        .addComponent(jLabel14)
        .addContainerGap(171, Short.MAX_VALUE)
    );
jDialog4Layout.setVerticalGroup(

```

```

jDialog4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jDialog4Layout.createSequentialGroup()
        .addGap(82, 82, 82)
        .addComponent(jLabel14)
        .addContainerGap(204, Short.MAX_VALUE)
    );

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setBackground(new java.awt.Color(255, 255, 255));

jRadioButton1.setBackground(new java.awt.Color(255, 255, 255));
jRadioButton1.setSelected(true);
jRadioButton1.setText("Modo Manual");
jRadioButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt)
    {
        jRadioButton1ActionPerformed(evt);
    }
});

jRadioButton2.setBackground(new java.awt.Color(255, 255, 255));
jRadioButton2.setText("Modo Autom\u00e1tico");
jRadioButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt)
    {
        jRadioButton2ActionPerformed(evt);
    }
});

jLabel15.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
jLabel15.setText("Posi\u00e7\u00e3o Atual");

jLabel16.setFont(new java.awt.Font("Tahoma", 0, 36)); // NOI18N
jLabel16.setText("? graus");

jLabel17.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
jLabel17.setText("Repeti\u00e7\u00e3o Atual");

jLabel18.setFont(new java.awt.Font("Tahoma", 0, 36)); // NOI18N
jLabel18.setText("? repeti\u00e7\u00f5es");

jComboBox1.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Velocidade 1", "Velocidade 2", "Velocidade 3", "Velocidade 4", "Velocidade 5" }));

```

```

        JComboBox1.addActionListener(new java.
va.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt)
{
                JComboBox1ActionPerformed(evt);
            }
        });

jButton4.setText("Jog+");
jButton4.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mousePressed(java.awt.event.MouseEvent evt) {
        jButton4MousePressed(evt);
    }
    public void mouseReleased(java.awt.event.MouseEvent evt) {
        jButton4MouseReleased(evt);
    }
});
jButton4.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(java.awt.event.ActionEvent evt)
{
        jButton4ActionPerformed(evt);
    }
});

jButton5.setText("Jog-");
jButton5.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mousePressed(java.awt.event.MouseEvent evt) {
        jButton5MousePressed(evt);
    }
    public void mouseReleased(java.awt.event.MouseEvent evt) {
        jButton5MouseReleased(evt);
    }
});

jRadioButton3.setBackground(new java.awt.Color(255, 255, 255));
jRadioButton3.setSelected(true);
jRadioButton3.setText("Tratamento 1");
jRadioButton3.setEnabled(false);
jRadioButton3.addActionListener(new java.
va.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt)
{
        jRadioButton3ActionPerformed(evt);
    }
});

jRadioButton4.setBackground(new java.awt.Color(255, 255, 255));
jRadioButton4.setText("Tratamento 2");
jRadioButton4.setEnabled(false);

```

```

        jButton4.addActionListener(new java.
va.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt)
            {
                jButton4ActionPerformed(evt);
            }
        });

        jButton5.setBackground(new java.awt.Color(255, 255, 255));
        jButton5.setText("Tratamento 3");
        jButton5.setEnabled(false);
        jButton5.addActionListener(new java.
va.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt)
            {
                jButton5ActionPerformed(evt);
            }
        });

        jButton6.setBackground(new java.awt.Color(255, 255, 255));
        jButton6.setText("Personalizado");
        jButton6.setEnabled(false);
        jButton6.addActionListener(new java.
va.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt)
            {
                jButton6ActionPerformed(evt);
            }
        });

        jLabel1.setText("Ãngulo MÃximo:");
        jLabel1.setEnabled(false);

        jLabel2.setText("Ãngulo MÃnimo:");
        jLabel2.setEnabled(false);

        jLabel3.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
        jLabel3.setForeground(new java.awt.Color(255, 0, 0));
        jLabel3.setText("?");
        jLabel3.setEnabled(false);
        jLabel3.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseEntered(java.awt.event.MouseEvent evt) {
                jLabel3MouseEntered(evt);
            }
            public void mouseExited(java.awt.event.MouseEvent evt) {
                jLabel3MouseExited(evt);
            }
        });

        jLabel4.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N

```

```

jLabel4.setForeground(new java.awt.Color(255, 0, 0));
jLabel4.setText("?");
jLabel4.setEnabled(false);
jLabel4.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseEntered(java.awt.event.MouseEvent evt) {
        jLabel4MouseEntered(evt);
    }
    public void mouseExited(java.awt.event.MouseEvent evt) {
        jLabel4MouseExited(evt);
    }
});

jLabel9.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel9.setForeground(new java.awt.Color(255, 0, 0));
jLabel9.setText("?");
jLabel9.setEnabled(false);
jLabel9.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseEntered(java.awt.event.MouseEvent evt) {
        jLabel9MouseEntered(evt);
    }
    public void mouseExited(java.awt.event.MouseEvent evt) {
        jLabel9MouseExited(evt);
    }
});

jLabel10.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel10.setForeground(new java.awt.Color(255, 0, 0));
jLabel10.setText("?");
jLabel10.setEnabled(false);
jLabel10.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseEntered(java.awt.event.MouseEvent evt) {
        jLabel10MouseEntered(evt);
    }
    public void mouseExited(java.awt.event.MouseEvent evt) {
        jLabel10MouseExited(evt);
    }
});

jTextField1.setText("140");
jTextField1.setEnabled(false);
jTextField1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt)
    {
        jTextField1ActionPerformed(evt);
    }
});

jTextField2.setText("000");
jTextField2.setEnabled(false);

```

```

        jTextField2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt)
            {
                jTextField2ActionPerformed(evt);
            }
        });

        jButton6.setText("Iniciar");
        jButton6.setEnabled(false);
        jButton6.addActionListener(new java.awt.event.ActionListener()
        {
            public void actionPerformed(java.awt.event.ActionEvent evt)
            {
                jButton6ActionPerformed(evt);
            }
        });

        jComboBox2.setModel(new javax.swing.DefaultComboBoxModel(new
String[] { "Velocidade 1", "Velocidade 2", "Velocidade 3", "Velocidade
4", "Velocidade 5" }));
        jComboBox2.setEnabled(false);
        jComboBox2.addActionListener(new java
        .awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt)
            {
                jComboBox2ActionPerformed(evt);
            }
        });

        jButton7.setText("Parar");
        jButton7.setEnabled(false);
        jButton7.addActionListener(new java.awt.event.ActionListener()
        {
            public void actionPerformed(java.awt.event.ActionEvent evt)
            {
                jButton7ActionPerformed(evt);
            }
        });

        jLabel15.setText("Nmero de Repeties:");
        jLabel15.setEnabled(false);

        jButton8.setText("Home");
        jButton8.setEnabled(false);

        jComboBox3.setModel(new javax.swing.DefaultComboBoxModel(new
String[] { "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11",
"12", "13", "14", "15", "16", "17", "18", "19", "20", "25", "30", "35",
"40", "45", "50", "55", "60", "65", "70", "75", "80", "85", "90" }));

```



```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                .addComponent(jLabel3))

.addGroup(layout.createSequentialGroup())

.addComponent(jRadioButton6)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addComponent(jLabel10))

.addGroup(layout.createSequentialGroup())
                                .addGap(1, 1, 1)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
TRAILING)

.addComponent(jRadioButton4)

.addComponent(jRadioButton5))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)

.addComponent(jLabel4)

.addComponent(jLabel9)))
                                .addGap(13, 13, 13)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, lay-
out.createSequentialGroup())
                                .addGap(191, 191, 191)

.addComponent(jComboBox3, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                                .addGap(18, 18, 18)
                                .addComponent(jButton7,
javax.swing.GroupLayout.PREFERRED_SIZE, 78,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(layout.createSequentialGroup())
                                .addGap(246, 246, 246)

```

```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)

.addComponent(jButton8)

.addComponent(jButton6, javax.swing.GroupLayout.PREFERRED_SIZE, 78,
javax.swing.GroupLayout.PREFERRED_SIZE))))

.addGroup(layout.createSequentialGroup())

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
                                .addComponent(jLabel16)

.addGroup(layout.createSequentialGroup())
                                .addComponent(jButton4)
                                .addGap(18, 18, 18)
                                .addComponent(jButton5)
                                .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING, false)

.addComponent(jComboBox2, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

.addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED_SIZE, 204,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel15))

.addGroup(layout.createSequentialGroup())
                                .addGap(14, 14, 14)

.addComponent(jLabel17))
                                .addGap(112, 112, 112))))
                                .addGroup(layout.createSequentialGroup())
                                    .addGap(163, 163, 163)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
                                .addComponent(jLabel2)
                                .addComponent(jLabel1))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)

```

```

        .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE, 33,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jTextField2,
javax.swing.GroupLayout.PREFERRED_SIZE, 33,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(315, 315, 315))
        .addContainerGap()
        .addGroup(layout.createSequentialGroup())

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
        .addComponent(jLabel6)
        .addComponent(jLabel5))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
        .addComponent(jLabel7)
        .addComponent(jLabel8))
        .addGap(122, 122, 122)))
);
layout.setVerticalGroup(
    lay-
out.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel17,
javax.swing.GroupLayout.PREFERRED_SIZE, 15,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jLabel16)
            .addGap(38, 38, 38)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
            .addComponent(jRadioButton1)
            .addComponent(jButton5)
            .addComponent(jButton4)
            .addComponent(jComboBox1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jButton8))
            .addGap(18, 18, 18)

```

```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
            .addComponent(jComboBox2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jButton6))
        .addGroup(layout.createSequentialGroup())

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
            .addComponent(jRadioButton2)
            .addComponent(jRadioButton3)
            .addComponent(jLabel3))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
            .addComponent(jRadioButton4)
            .addComponent(jLabel4))))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, lay-
out.createSequentialGroup())
        .addGap(11, 11, 11)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
            .addComponent(jLabel15)
            .addComponent(jButton7)
            .addComponent(jComboBox3,
javax.swing.GroupLayout.PREFERRED_SIZE, 23,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(26, 26, 26))
        .addGroup(layout.createSequentialGroup())

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
            .addComponent(jRadioButton5)
            .addComponent(jLabel9))

```

```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
            .addComponent(jRadioButton6)
            .addComponent(jLabel10)))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
            .addComponent(jLabel2)
            .addComponent(jTextField2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
            .addComponent(jLabel1)
            .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
            .addComponent(jLabel5)
            .addComponent(jLabel7))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(11, 11, 11)
                .addComponent(jLabel6))
            .addGroup(layout.createSequentialGroup()
                .addGap(18, 18, 18)
                .addComponent(jLabel8)))
            .addContainerGap(45, Short.MAX_VALUE)
);

pack();
} // </editor-fold> // GEN-END: initComponents

private void
jRadioButton2ActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST:event_jRadioButton2ActionPerformed

```

```

    if (jRadioButton2.isSelected()){
        jRadioButton1.setSelected(false);
    }
    jRadioButton2.setSelected(true);

    jButton4.setEnabled(false);
    jButton5.setEnabled(false);
    jComboBox1.setEnabled(false);

    jRadioButton3.setEnabled(true);
    jRadioButton4.setEnabled(true);
    jRadioButton5.setEnabled(true);
    jRadioButton6.setEnabled(true);
    if (jRadioButton6.isSelected()){
        jTextField1.setEnabled(true);
        jTextField2.setEnabled(true);
        jLabel1.setEnabled(true);
        jLabel2.setEnabled(true);
    }
    jLabel3.setEnabled(true);
    jLabel4.setEnabled(true);
    jLabel9.setEnabled(true);
    jLabel10.setEnabled(true);
    jButton6.setEnabled(true);
    jButton7.setEnabled(true);
    jComboBox2.setEnabled(true);
    jComboBox3.setEnabled(true);
    jLabel15.setEnabled(true);
    jButton8.setEnabled(true);
    control.send(":SETAR -MOD0 AUTOMATICO!");
} //GEN-LAST:event_jRadioButton2ActionPerformed

private void
jRadioButton1ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_jRadioButton1ActionPerformed
    if (jRadioButton1.isSelected()){
        jRadioButton2.setSelected(false);
    }
    jRadioButton1.setSelected(true);
    jButton4.setEnabled(true);
    jButton5.setEnabled(true);
    jComboBox1.setEnabled(true);
    jRadioButton3.setEnabled(false);
    jRadioButton4.setEnabled(false);
    jRadioButton5.setEnabled(false);
    jRadioButton6.setEnabled(false);
    jTextField1.setEnabled(false);
    jTextField2.setEnabled(false);
    jLabel3.setEnabled(false);
    jLabel4.setEnabled(false);

```

```

        jLabel9.setEnabled(false);
        jLabel10.setEnabled(false);
        jButton6.setEnabled(false);
        jButton7.setEnabled(false);
        jButton8.setEnabled(false);
        jComboBox2.setEnabled(false);
        jComboBox3.setEnabled(false);
        jLabel15.setEnabled(false);
        jLabel11.setEnabled(false);
        jLabel12.setEnabled(false);
        control.send(":SETAR -MODO MANUAL!");

    }//GEN-LAST:event_jRadioButton1ActionPerformed

    private void
jRadioButton3ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_jRadioButton3ActionPerformed
        jButton3.setSelected(true);
        jButton4.setSelected(false);
        jButton5.setSelected(false);
        jButton6.setSelected(false);
        jLabel11.setEnabled(false);
        jLabel12.setEnabled(false);
        jTextField1.setEnabled(false);
        jTextField2.setEnabled(false);
        control.send(":SETAR -TRATAMENTO 000 140!");
    }//GEN-LAST:event_jRadioButton3ActionPerformed

    private void
jRadioButton4ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_jRadioButton4ActionPerformed
        jButton4.setSelected(true);
        jButton3.setSelected(false);
        jButton5.setSelected(false);
        jButton6.setSelected(false);
        jTextField1.setEnabled(false);
        jTextField2.setEnabled(false);
        jLabel11.setEnabled(false);
        jLabel12.setEnabled(false);
        control.send(":SETAR -TRATAMENTO 000 090!");
    }//GEN-LAST:event_jRadioButton4ActionPerformed

    private void
jRadioButton5ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_jRadioButton5ActionPerformed
        jButton5.setSelected(true);
        jButton4.setSelected(false);
        jButton3.setSelected(false);
        jButton6.setSelected(false);
        jTextField1.setEnabled(false);

```

```

        jTextField2.setEnabled(false);
        jLabel1.setEnabled(false);
        jLabel2.setEnabled(false);
        control.send(":SETAR -TRATAMENTO 090 140!");
    }//GEN-LAST:event_jRadioButton5ActionPerformed

    private void
jRadioButton6ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_jRadioButton6ActionPerformed
        jButton6.setSelected(true);
        jButton4.setSelected(false);
        jButton5.setSelected(false);
        jButton3.setSelected(false);
        jTextField1.setEnabled(true);
        jTextField2.setEnabled(true);
        jLabel1.setEnabled(true);
        jLabel2.setEnabled(true);
        control.send(":SETAR -TRATAMENTO " + jTextField2.getText() + "
" + jTextField1.getText() + "!");
    }//GEN-LAST:event_jRadioButton6ActionPerformed

    private void jLabel3MouseEntered(java.awt.event.MouseEvent evt)
{//GEN-FIRST:event_jLabel3MouseEntered
        if (jLabel3.isEnabled()){
            dialog1.setLocation(jLabel3.getLocationOnScreen().x + 20,
jLabel3.getLocationOnScreen().y);
            dialog1.setVisible(true);
        }
    }//GEN-LAST:event_jLabel3MouseEntered

    private void jLabel3MouseExited(java.awt.event.MouseEvent evt)
{//GEN-FIRST:event_jLabel3MouseExited
        dialog1.setVisible(false);
    }//GEN-LAST:event_jLabel3MouseExited

    private void jLabel4MouseEntered(java.awt.event.MouseEvent evt)
{//GEN-FIRST:event_jLabel4MouseEntered
        if (jLabel4.isEnabled()){
            dialog2.setLocation(jLabel4.getLocationOnScreen().x + 20,
jLabel4.getLocationOnScreen().y);
            dialog2.setVisible(true);
        }
    }//GEN-LAST:event_jLabel4MouseEntered

    private void jLabel4MouseExited(java.awt.event.MouseEvent evt)
{//GEN-FIRST:event_jLabel4MouseExited
        dialog2.setVisible(false);
    }//GEN-LAST:event_jLabel4MouseExited

```



```

        private void jLabel9MouseEntered(java.awt.event.MouseEvent evt)
    { //GEN-FIRST:event_jLabel9MouseEntered
        if (jLabel9.isEnabled()) {
            jDialog3.setLocation(jLabel9.getLocationOnScreen().x + 20,
jLabel9.getLocationOnScreen().y);
            jDialog3.setVisible(true);
        }
    } //GEN-LAST:event_jLabel9MouseEntered

        private void jLabel9MouseExited(java.awt.event.MouseEvent evt)
    { //GEN-FIRST:event_jLabel9MouseExited
        jDialog3.setVisible(false);
    } //GEN-LAST:event_jLabel9MouseExited

        private void jLabel10MouseEntered(java.awt.event.MouseEvent evt)
    { //GEN-FIRST:event_jLabel10MouseEntered
        if (jLabel10.isEnabled()) {
            jDialog4.setLocation(jLabel10.getLocationOnScreen().x + 20,
jLabel10.getLocationOnScreen().y);
            jDialog4.setVisible(true);
        }
    } //GEN-LAST:event_jLabel10MouseEntered

        private void jLabel10MouseExited(java.awt.event.MouseEvent evt)
    { //GEN-FIRST:event_jLabel10MouseExited
        jDialog4.setVisible(false);
    } //GEN-LAST:event_jLabel10MouseExited

        private void jButton4ActionPerformed(java.awt.event.ActionEvent
evt) { //GEN-FIRST:event_jButton4ActionPerformed
        // TODO add your handling code here:
    } //GEN-LAST:event_jButton4ActionPerformed

        private void jButton4MousePressed(java.awt.event.MouseEvent evt)
    { //GEN-FIRST:event_jButton4MousePressed
        control.send(":JOG+!");
    } //GEN-LAST:event_jButton4MousePressed

        private void jButton4MouseReleased(java.awt.event.MouseEvent evt)
    { //GEN-FIRST:event_jButton4MouseReleased
        control.send(":NOJOG!");
    } //GEN-LAST:event_jButton4MouseReleased

        private void jButton5MousePressed(java.awt.event.MouseEvent evt)
    { //GEN-FIRST:event_jButton5MousePressed
        control.send(":JOG-!");
    } //GEN-LAST:event_jButton5MousePressed

        private void jButton5MouseReleased(java.awt.event.MouseEvent evt)
    { //GEN-FIRST:event_jButton5MouseReleased

```

```

        control.send(":NOJOG!");
    }//GEN-LAST:event_jButton5MouseReleased

    private void jButton1ActionPerformed(java.awt.event.ActionEvent
    evt) { //GEN-FIRST:event_jButton1ActionPerformed
        if (jComboBox1.getSelectedIndex() == 0){
            control.send(":SETAR -VELOCIDADE 1250!");
        }
        if (jComboBox1.getSelectedIndex() == 1){
            control.send(":SETAR -VELOCIDADE 1000!");
        }
        if (jComboBox1.getSelectedIndex() == 2){
            control.send(":SETAR -VELOCIDADE 750!");
        }
        if (jComboBox1.getSelectedIndex() == 3){
            control.send(":SETAR -VELOCIDADE 500!");
        }
        if (jComboBox1.getSelectedIndex() == 4){
            control.send(":SETAR -VELOCIDADE 250!");
        }
    } //GEN-LAST:event_jButton1ActionPerformed

    private void jButton6ActionPerformed(java.awt.event.ActionEvent
    evt) { //GEN-FIRST:event_jButton6ActionPerformed
        if (jButton6.getText().equals("Iniciar")){
            control.send(":INICIAR!");
            jButton6.setText("Pausar");
        }
        else if (jButton6.getText().equals("Pausar")){
            control.send(":PAUSAR!");
            jButton6.setText("Iniciar");
        }
    } //GEN-LAST:event_jButton6ActionPerformed

    private void jButton7ActionPerformed(java.awt.event.ActionEvent
    evt) { //GEN-FIRST:event_jButton7ActionPerformed
        control.send(":PARAR!");
        if (jButton6.getText().equals("Pausar")){
            jButton6.setText("Iniciar");
        }
    } //GEN-LAST:event_jButton7ActionPerformed

    private void jButton2ActionPerformed(java.awt.event.ActionEvent
    evt) { //GEN-FIRST:event_jButton2ActionPerformed
        if (jComboBox2.getSelectedIndex() == 0){
            control.send(":SETAR -VELOCIDADE 1250!");
        }
        if (jComboBox2.getSelectedIndex() == 1){
            control.send(":SETAR -VELOCIDADE 1000!");
        }
    }

```

```

        if (jComboBox2.getSelectedIndex() == 2){
            control.send(":SETAR -VELOCIDADE 750!");
        }
        if (jComboBox2.getSelectedIndex() == 3){
            control.send(":SETAR -VELOCIDADE 500!");
        }
        if (jComboBox2.getSelectedIndex() == 4){
            control.send(":SETAR -VELOCIDADE 250!");
        }
    } //GEN-LAST:event_jComboBox2ActionPerformed

    private void jTextField2ActionPerformed(java.awt.event.ActionEvent
    evt) { //GEN-FIRST:event_jTextField2ActionPerformed
        control.send(":SETAR -TRATAMENTO " + jTextField2.getText() + "
    " + jTextField1.getText() + "!");
    } //GEN-LAST:event_jTextField2ActionPerformed

    private void jComboBox3ActionPerformed(java.awt.event.ActionEvent
    evt) { //GEN-FIRST:event_jComboBox3ActionPerformed
        control.send(":SETAR -REPETICOES " +
    jComboBox3.getSelectedItem().toString() + "!");
    } //GEN-LAST:event_jComboBox3ActionPerformed

    private void jTextField1ActionPerformed(java.awt.event.ActionEvent
    evt) { //GEN-FIRST:event_jTextField1ActionPerformed
        control.send(":SETAR -TRATAMENTO " + jTextField2.getText() + "
    " + jTextField1.getText() + "!");
    } //GEN-LAST:event_jTextField1ActionPerformed

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel
    setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay
    with the default look and feel.
         * For details see
    http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.htm
    l
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
    javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {

    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        }
    }

```

```

    }
    } catch (ClassNotFoundException ex) {
        ja-
va.util.logging.Logger.getLogger(IHM.class.getName()).log(java.util.log
ging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        ja-
va.util.logging.Logger.getLogger(IHM.class.getName()).log(java.util.log
ging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        ja-
va.util.logging.Logger.getLogger(IHM.class.getName()).log(java.util.log
ging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        ja-
va.util.logging.Logger.getLogger(IHM.class.getName()).log(java.util.log
ging.Level.SEVERE, null, ex);
    }
}
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new IHM().setVisible(true);
    }
});
}
// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton jButton4;
private javax.swing.JButton jButton5;
private javax.swing.JButton jButton6;
private javax.swing.JButton jButton7;
private javax.swing.JButton jButton8;
private javax.swing.JComboBox jComboBox1;
private javax.swing.JComboBox jComboBox2;
private javax.swing.JComboBox jComboBox3;
private javax.swing.JDialog jDialog1;
private javax.swing.JDialog jDialog2;
private javax.swing.JDialog jDialog3;
private javax.swing.JDialog jDialog4;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel16;
private javax.swing.JLabel jLabel17;
private javax.swing.JLabel jLabel2;

```

```
private javax.swing.JLabel jLabel3;  
private javax.swing.JLabel jLabel4;  
private javax.swing.JLabel jLabel5;  
private javax.swing.JLabel jLabel6;  
private javax.swing.JLabel jLabel7;  
private javax.swing.JLabel jLabel8;  
private javax.swing.JLabel jLabel9;  
private javax.swing.JRadioButton jButton1;  
private javax.swing.JRadioButton jButton2;  
private javax.swing.JRadioButton jButton3;  
private javax.swing.JRadioButton jButton4;  
private javax.swing.JRadioButton jButton5;  
private javax.swing.JRadioButton jButton6;  
private javax.swing.JTextField jTextField1;  
private javax.swing.JTextField jTextField2;  
// End of variables declaration//GEN-END:variables  
  
}
```

APÊNDICE D - CÓDIGO EM C++ PARA ATMEGA

```
/******
```

```
Escola Politécnica da Universidade de São Paulo  
Departamento de Engenharia Mecatrônica e Sistemas Mecânicos  
PMR 2500 - 2550: Trabalho de Conclusão de Curso
```

```
Guilherme Namindome  
Murilo Oliveira
```

```
*****/
```

```
const int pinostep = 12;  
const int pinodir = 11;  
int velocidade = 750;  
int posicao = 0;  
int angmin = 0;  
int angmax = 140;  
int repeticoes = 3;  
int i = 0;  
int j = 0;  
String comando = "";  
String modo = "MANUAL";  
String estado = "PARADO";  
boolean flaginiciar = false;  
boolean flagpausar = false;  
boolean flagparar = false;  
boolean flagjogmais = false;  
boolean flagjogmenos = false;  
boolean flagnojog = false;  
boolean direcao = LOW;
```

```

void setup(){
  Serial.begin(9600);
  comando.reserve(30);
  Serial.flush();
  pinMode(pinostep, OUTPUT);
  pinMode(pinodir, OUTPUT);
  angmin = fteta(0);
  angmax = fteta(140);
}

void loop() {
  //stepper
  if (modo == "MANUAL" && !flagnojog){
    if (flagjogmais){
      digitalWrite(pinostep, HIGH);
      delayMicroseconds(velocidade);
      digitalWrite(pinostep, LOW);
      delayMicroseconds(velocidade);
      posicao = posicao + 1;
    }
    else if (flagjogmenos){
      digitalWrite(pinostep, HIGH);
      delayMicroseconds(velocidade);
      digitalWrite(pinostep, LOW);
      delayMicroseconds(velocidade);
      posicao = posicao - 1;
    }
  }
  else if (modo == "AUTOMATICO"){
    if (flaginiciar && !flagpausar){
      if ((estado == "PARADO" || estado == "ESTENDENDO") && posicao >
angmin && posicao < angmax){
        //estender
        if (direcao == HIGH){

```

```

    direcao = LOW;
    digitalWrite(pinodir, direcao);
    delayMicroseconds(velocidade);
}
digitalWrite(pinostep, HIGH);
delayMicroseconds(velocidade);
digitalWrite(pinostep, LOW);
delayMicroseconds(velocidade);
posicao = posicao - 1;
}
else if ((estado == "PARADO" || estado == "FLEXIONANDO") && posicao > angmin && posicao < angmax){
    //flexionar
    if (direcao == LOW){
        direcao = HIGH;
        digitalWrite(pinodir, direcao);
        delayMicroseconds(velocidade);
    }
    digitalWrite(pinostep, HIGH);
    delayMicroseconds(velocidade);
    digitalWrite(pinostep, LOW);
    delayMicroseconds(velocidade);
    posicao = posicao + 1;
}
else if ((estado == "PARADO" || estado == "ESTENDENDO") && posicao <= angmin){
    //trocar - flexionar
    if (direcao == LOW){
        direcao = HIGH;
        digitalWrite(pinodir, direcao);
        delayMicroseconds(velocidade);
    }
    digitalWrite(pinostep, HIGH);
    delayMicroseconds(velocidade);
}

```



```

digitalWrite(pinostep, LOW);
delayMicroseconds(velocidade);
posicao = posicao + 1;
if (posicao > angmin){
    estado = "FLEXIONANDO";
    if (i < repeticoes){
        i++;
    }
    else {
        flaginiciar = false;
        Serial.println(":FIM!");
        delayMicroseconds(velocidade);
        estado = "PARADO";
    }
}
}
else if ((estado == "PARADO" || estado == "FLEXIONANDO") && posicao >= angmax){
    //trocar - estender
    if (direcao == HIGH){
        direcao = LOW;
        digitalWrite(pinodir, direcao);
        delayMicroseconds(velocidade);
    }
    estado = "ESTENDENDO";
    digitalWrite(pinostep, HIGH);
    delayMicroseconds(velocidade);
    digitalWrite(pinostep, LOW);
    delayMicroseconds(velocidade);
    posicao = posicao - 1;
}
}
}
}
}

```

```

void serialEvent() {
    while (Serial.available()) {
        char inChar = (char)Serial.read();
        comando += inChar;
        if (inChar == '!') {
            tratarcomando();
        }
    }
}

void tratarcomando(){
    //tratamento da string de comando
    Serial.println(comando);
    delayMicroseconds(velocidade);
    if (comando == ":INICIAR!") {
        flaginiciar = true;
        flagpausar = false;
        flagparar = false;
    }
    else if (comando == ":PAUSAR!") {
        flagpausar = true;
        flaginiciar = false;
    }
    else if (comando == ":PARAR!") {
        flagparar = true;
        flaginiciar = false;
        flagpausar = false;
        i = 0;
    }
    else if (comando == ":SETAR -MODO MANUAL!") {
        modo = "MANUAL";
    }
    else if (comando == ":SETAR -MODO AUTOMATICO!") {

```

```

modo = "AUTOMATICO";
}
else if (comando.substring(0, 18) == ":SETAR -VELOCIDADE"){
if (comando[20] == '!'){
    velocidade = (comando[19] - '0');
}
if (comando[21] == '!'){
    velocidade = (comando[19] - '0')*10 + (comando[20] - '0');
}
else if (comando[22] == '!'){
    velocidade = (comando[19] - '0')*100 + (comando[20] - '0')*10 +
(comando[21] - '0');
}
else if (comando[23] == '!'){
    velocidade = (comando[19] - '0')*1000 + (comando[20] - '0')*100 +
(comando[21] - '0')*10 + (comando[22] - '0');
}
}
else if (comando == ":JOG+!"){
flagjogmais = true;
flagnojog = false;
flagjogmenos = false;
if (direcao == LOW){
    direcao = HIGH;
    digitalWrite(pinodir, direcao);
    delayMicroseconds(velocidade);
}
}
else if (comando == ":JOG-!"){
flagjogmenos = true;
flagnojog = false;
flagjogmais = false;
if (direcao == HIGH){
    direcao = LOW;
}
}

```

```

        digitalWrite(pinodir, direcao);
        delayMicroseconds(velocidade);
    }
}
else if (comando == ":NOJOG!") {
    flagnojog = true;
    flagjogmais = false;
    flagjogmenos = false;
}
else if (comando == ":POS!") {
    comando = ":RESPOSTA " + String(fpasso(posicao)) + "@" + String(i)
+ "!";
    Serial.println(comando);
    delayMicroseconds(velocidade);
}
else if (comando.substring(0, 18) == ":SETAR -TRATAMENTO") {
    angmin = (comando[19] - '0')*100 + (comando[20] - '0')*10 + (coman-
do[21] - '0');
    angmax = (comando[23] - '0')*100 + (comando[24] - '0')*10 + (coman-
do[25] - '0');
    angmin = fteta(angmin);
    angmax = fteta(angmax);
}
else if (comando.substring(0, 18) == ":SETAR -REPETICOES") {
    if (comando[20] == '!') {
        repeticoes = (comando[19] - '0');
    }
    if (comando[21] == '!') {
        repeticoes = (comando[19] - '0')*10 + (comando[20] - '0');
    }
}
comando = "";
}
}

```

```

int fteta(int teta){
    int passo;
    passo = (int)((2E-7)*(teta^5)-(4E-
5)*(teta^4)+(0.0022)*(teta^3)+(0.1251)*(teta^2)+(113.93)*(teta));
    //passo = teta;
    return passo;
}

int fpasso(int passo){
    int teta;
    teta = (int)((1E-20)*(passo^5)-(8E-16)*(passo^4)+(1E-11)*(passo^3)-
(2E-7)*(passo^2)+(0.0089)*passo);
    //teta = passo;
    return teta;
}

```