



ISR - Softhand

Joana Rita Braga dos Santos

Design de sinergias adaptativas e de sistemas de parâmetros de entrada para o controlo de uma mão protética sub atuada

Dissertação apresentada no Departamento de Engenharia Eletrotécnica e Computadores da Faculdade de Ciências e Tecnologia da Universidade de Coimbra para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Biomédica

September 2014



UNIVERSIDADE DE COIMBRA



FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

JOANA RITA BRAGA DOS SANTOS

Design de sinergias adaptativas e de
sistemas de parâmetros de entrada para o
controlo de uma mão prótica sub atuada

Dissertação apresentada à Universidade de Coimbra
para cumprimento dos requisitos necessários à obtenção
do grau de Mestre em Engenharia Biomédica

Orientadores:

Dr. Mahmoud TAVAKOLI (ISR-DEEC)

Prof. Dr. Lino MARQUES (DEEC)

Coimbra, 2014

Este trabalho foi desenvolvido em colaboração com:



Esta cópia da tese é fornecida na condição de que quem a consulta reconhece que os direitos de autor são pertença do autor da tese e que nenhuma citação ou informação obtida a partir dela pode ser publicada sem a referência apropriada.

This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognize that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without proper acknowledgement.

Agradecimentos

A realização desta dissertação de mestrado contou com importantes apoios e incentivos sem os quais não se teria tornado uma realidade e a os quais estarei eternamente grata.

Quero agradecer ao Professor Dr. Lino Marques pela total disponibilidade e pelas palavras de incentivo. Quero também agradecer ao Dr. Mahmoud Tavakoli pela orientação, apoio, pelas opiniões e críticas, pela total colaboração aquando a solucionar de dúvidas e problemas que surgiram ao longo do projeto. Agradeço ainda toda a paciência e esforço nas nossas conversas em Português.

Quero agradecer a todos os presentes no laboratório de Sistemas Embebidos do Instituto de Sistemas e Robótica da Universidade de Coimbra, principalmente ao Tifi, Ricardo, Daniel, Bruno e Cabrita, por manterem a boa disposição e alegria no laboratório e por estarem sempre disponíveis para me auxiliarem.

No laboratório tenho desde mais a agradecer aos meus colegas e amigos durante este ano Tomé e Baptiste por aturarem os maus humores e as alegrias e por me acompanharem todas as 8473893749 vezes que quis ir à varanda do 2º andar porque “já não dava mais”. A vocês, o maior obrigado.

Agradeço aqueles que estiveram sempre comigo, Mariana, Natasha e Devesa. Mariana obrigada por me aturares todos os dias em casa com tudo o que aconteceu e não aconteceu, Natasha obrigada pelas pausas e invasões pelo ISR e Devesa obrigada por me guiares neste fase (Sr.Engenheiro!) e por leres e corrigires todo o meu lindo trabalho!

A estes 5 LINDIDOS do meu ♡ agradeço ainda todos os grandes momentos como

Veteranos! Á melhor latada, às melhores férias com as "snokiies", aos melhores jantares, às melhores saídas! ♡ U ALL

Agradeço também as meninas do HOTBERG por todos os momentos.

Agradeço também a "Vila do Conde, espaiada entre pinhais, rio e mar!" (José Régio, *Poesia I*) pela tranquilidade e paz de espírito. Agradeço às minhas meninas de Vila do Conde, Marta e Xana, por me fazerem relaxar e descontraír no difícil mês de Agosto!

Por fim agradeço á minha família, pois sem eles nada disto seria possível. Aos meus pais pelo apoio e paciência e ao meu irmão (João Augusto B. Santos aka *Gusto o maior*) que tudo fez para ver o nome dele aqui presente, incluindo ler uma tese de "quase 300 páginas"!

“Lo imposible so lo tarde um pouco mas”

“Man is the most intelligent of animals because he has hands” - Anaxagoras

Resumo

Esta dissertação estuda e projeta sinergias adaptativas para que uma mão robótica, a *ISR-Soft-hand* consiga agarrar o maior número de objetos possível.

A *ISR-Soft-hand* é um protótipo de uma mão robótica antropomórfica sub-atuada complacente, com articulações elásticas, para aplicações próstéticas. A complacência incluída nas articulações dos dedos, promove uma boa adaptabilidade da mão ao agarrar um vasto número de objetos com apenas três atuadores. Deste modo, é pretendido criar o mínimo de sinergias adaptativas que consigam agarrar o máximo de objetos possível.

Neste trabalho a mão robótica foi proposta a imitar uma lista de 33 preensões utilizadas no dia a dia, tendo as preensões realizáveis sido agrupadas em cinco grupos e conseqüentemente criadas cinco sinergias adaptativas e respetivos parâmetros de entrada para cada um dos grupos, respetivamente.

Além do projeto das sinergias adaptativas, foram ainda tiradas conclusões sobre a relação entre a complacência da mão e a simplicidade dos sistemas de controlo.

Por fim, foi feita uma análise visando o projecto de uma mão mais simples, com um número reduzido de atuadores e sinergias, sem que a funcionalidade da mão seja afetada, com o objetivo da aplicação de eléctrodos de Eletromiografia (EMG) para o futuro controlo da prótese robótica.

Palavras chave: Mão próstética sub atuada; Preensão; Sinergia; Sinergias Adaptativas.

Abstract

This dissertation focuses on the test and design of grasping synergies for the ISR-Soft hand to perform the maximum number of different grasps.

ISR-Soft hand is an anthropomorphic hand with embedded compliant elements and elastic joints for prosthesis applications. The embedded compliance in the joints of the fingers enables a good adaptability of the hand for grasping a wide range of objects with only 3 actuators. Thus, the aim is to design a minimum number of synergies to perform a maximum number of grasps.

Experiments were made with the ISR-Soft hand in order to imitate a list of 33 grasps which are being performed by humans in daily tasks, and based on the results the grasps were categorized into 5 groups and synergies and system inputs were designed for each group.

In addition, after the design of the adaptive synergies for the ISR-Soft hand, important conclusions were obtained regarding how the integrated compliance helps in a less complex control system.

A further analysis also suggested a simpler hand with reduced number of actuators and synergies, without significantly decreasing the hand's functionality, with the objective of implementation of electromyography electrodes for the control of the prosthetic hand.

Key Words: Underactuated prosthetic hand; Grasp; Synergy; Adaptive synergies.

Lista de acrónimos

CMC CarpoMetaCarpiana

DoA Grau de Atuação (*Degree of actuation*)

DoF Grau de Liberdade (*Degree of freedom*)

DV Dedos Virtuais

EMG Eletromiografia

IF InterFalangiana

IFD InterFalangiana Distal

IFP InterFalangiana Proximal

MCF MetaCarpoFalangiana

SNC Sistema Nervoso Central

Lista de Figuras

2.1	Ossos da mão humana (a cores) e respectivas articulações (a cinzento).	8
2.2	Movimentos realizáveis pela articulação MetaCarpoFalangiana (MCF).	9
2.3	Movimentos realizáveis pelas articulações InterFalangiana Proximal (IFP) e InterFalangiana Distal (IFD) de flexão/extensão.	10
2.4	Movimentos realizáveis pelas articulações MetaCarpoFalangiana (MCF) e CarpoMelaCarpiana (CMC) do polegar.	10
2.5	Representação dos 6 possíveis graus de liberdade de uma articulação.	12
2.6	Esquema das 19 articulações que compõe a <i>Pisa/IIT SoftHand</i>	13
2.7	Esquema de mão robótica com 3 dedos, simplificada, com a implementação de diferentes paradigmas de atuação.	15
2.8	Direções básicas da força realizada pela mão para a execução das <i>grasps</i> .	21
2.9	Sequência de movimentos da mão humana ao abrir um frasco. Evolução do uso da <i>power grasp</i> para <i>grasp</i> de precisão.	22
2.10	Visualização de cada um dos tipo de <i>grasps</i> : <i>Power</i> , precisão e intermédia.	22
2.11	Mãos protéticas disponíveis no mercado.	24
2.12	Mãos protéticas em desenvolvimento.	24
2.13	A Categorização de Schlesinger.	27
2.14	Lógica utilizada na categorização de Elliott e Connolly.	28
2.15	Categorização de Cutkosky	29
2.16	Representação das duas sinergias adaptativas principais.	31
2.17	Ilustração que mostra as estratégias de controlo aquando o uso de um ou dois eletrodos de EMG.	33

2.18	Processamento de sinal de um sistema de múltiplos sinais de entrada de EMG de superfície.	34
2.19	Distinção entre os dois algoritmos usados no processamento de sinal de múltiplos sinais de entrada de EMG de superfície.	35
2.20	Vários métodos de controlo de mãos protéticas robóticas.	36
3.1	Design conceptual da <i>ISR-Softhand</i>	38
3.2	Esquema onde se visualiza a relação de cada atuador com cada dedo e com a palma na <i>ISR-Softhand</i>	39
3.3	Esquema de montagem e comunicação entre os atuadores DYNAMIXEL AX-12A da ROBOTIS.	40
3.4	Opções possíveis para a configuração dos atuadores DYNAMIXEL da ROBOTIS.	41
3.5	Esquema de montagem real.	41
3.6	Interfaces existentes no software RoboPlus da ROBOTIS®.	42
4.1	Fluxograma de implementação inicial dos parâmetros de entrada na <i>ISR-Softhand</i>	44
4.2	Binário (τ), em unidades do software RoboPlus Manager, vs. Posição (θ), em graus ($^{\circ}$).	45
4.3	Objetos utilizados para testar cada uma das <i>grasps</i> e desenvolver as sinergias adaptativas para a <i>ISR-Softhand</i>	46
4.4	<i>ISR-Softhand</i> com o polegar em Abdução.	51
4.5	<i>ISR-Softhand</i> com o polegar em Adução.	51
4.6	Fotografias das <i>grasps</i> n.º 18 e 22, realizadas com a Sinergia I e V.	53
4.7	Estiramento dos tendões, em mm, em cada passo da Sinergia I	56
4.8	Figura onde se visualiza o problema do desenho mecânico (a vermelho) que impede o máximo fecho dos 3 dedos atuados pelo atuador 3.	57
4.9	Velocidade rotacional (rpm) de cada atuador em cada passo da Sinergia I.	58
4.10	<i>Grasp</i> de um ovo, realizado pelo passo 3 da Sinergia I.	58

4.11 Estiramento dos tendões, em milímetros, em cada uma das Sinergias II, III, IV e V. Nestas a velocidade rotacional de todos os atuadores é mantida nas 5,9 rpm. 61

Lista de Tabelas

2.1	Variáveis utilizadas para a análise dos sistemas de atuação.	14
2.2	Características gerais de 5 mãos próstéticas robóticas disponíveis no mercado.	25
2.3	Características gerais de 4 mãos próstéticas sub atuadas em desenvolvimento.	26
3.1	Características da <i>ISR-Soft hand</i>	39
3.2	Características do atuador DYNAMIXEL AX-12A da ROBOTIS. . .	40
4.1	Dimensões dos 21 objetos utilizados para testar cada uma das <i>grasps</i> e desenvolver as sinergias adaptativas para a <i>ISR-Soft hand</i>	46
4.2	Resumo da quantidade de <i>grasps</i> possíveis, aproximadas e impossíveis de realizar pela <i>ISR-Soft hand</i>	47
4.3	Representação das 33 <i>grasps</i> da lista de Feix et al. e a melhor imitação conseguida pela <i>ISR-Soft hand</i>	48
4.4	Número e percentagem de <i>grasps</i> realizáveis (em cada um dos grupos) e não realizáveis pela <i>ISR-Soft hand</i>	53
4.5	Sinergias passo a passo	54
4.6	Valores do estiramento do tendão, em milímetros, por cada atuador em cada um dos passos da Sinergia I.	56
4.7	<i>Grasps</i> que conseguem ser efetuadas por cada um dos passos da Sinergia I.	59

4.8	Valores do estiramento do tendão, em milímetros, por cada atuador em cada uma Sinergias II, III, IV e V. Atuador 1: polegar, Atuador 2: dedo indicador, Atuador 3: Restantes 3 dedos.	61
5.1	<i>Grasps</i> que seriam impossíveis de realizar com uma versão da <i>ISR-Softhand</i> com 2 atuadores.	65

Conteúdo

Agradecimentos	v
Resumo	ix
Abstract	xi
Lista de Acrónimos	xiii
Lista de Figuras	xv
Lista de Tabelas	xix
1 Introdução	1
1.1 Motivação e Enquadramento	1
1.2 Objetivos	3
1.3 Organização da tese	5
2 Conceitos Teóricos e Estado da arte	7
2.1 Conceitos teóricos	7
2.1.1 Anatomia da mão Humana	8
2.1.2 Mãos Robóticas	11
2.1.3 Sinergias	18
2.1.4 <i>Grasps</i>	19
2.2 Estado da arte	23
2.2.1 Mãos robóticas protéticas sub atuadas	23

2.2.2	Categorização das <i>grasps</i>	27
2.2.3	Estratégias de simplificação de controlo	30
3	ISR-Softhand	37
3.1	Características e <i>Design</i>	37
3.2	Atuadores	39
3.2.1	Atuador AX-12A	40
4	<i>Design</i>, Teste e Implementação das Sinergias Adaptativas	43
4.1	<i>Grasps</i>	43
4.1.1	Controlo de parâmetros de entrada e primeiros resultados . . .	45
4.2	Categorização das <i>grasps</i> e sinergias	50
4.3	Sinergias	53
4.3.1	Sinergia I.A & I.B	55
4.3.2	Sinergia II	58
4.3.3	Sinergia III	59
4.3.4	Sinergia IV	59
4.3.5	Sinergia V	60
5	Conclusões e Trabalho Futuro	63
5.1	Estratégia para reduzir o número de atuadores	63
5.1.1	2 Atuadores	64
5.1.2	1 Atuador	66
5.2	Estratégias para reduzir o número de sinergias	66
5.3	Conclusões	67
	Anexo Anexos	78
	Anexo A Publicações	81
	Anexo B Categorização de Feix et al.	83
	Anexo C Códigos Matlab[®]	89

Capítulo 1

Introdução

Este capítulo introduz esta dissertação, revelando a principal motivação e objetivo desta, e está organizado da seguinte forma:

1.1 Motivação e Enquadramento

1.2 Objetivos

1.3 Organização da tese

1.1 Motivação e Enquadramento

A amputação é um evento que muda a vida de qualquer pessoa, levando a que o paciente desenvolva problemas a nível psicológico e social. Os amputados do membro superior tendem a ter maiores taxas de depressão, dado que a este evento é particularmente traumático a nível pessoal. A maioria das amputações resulta de trauma, doença vascular periférica, infeções crónicas, lesões químicas, térmicas ou elétricas e tumores malignos. As amputações do membro superior ocorrem 75% das vezes devido a traumas e ocorrem principalmente entre homens jovens e sem qualquer problema de saúde [1].

Para resolver este problema, vários estudos foram realizados ao longo dos anos para desenvolver tecnologias e instrumentação que melhore a qualidade de vida destes pacientes. Dentro destas, este trabalho foca-se nas mãos próstéticas robóticas. As mãos robóticas que mimetizam mãos humanas são desenvolvidas quer para fins

industriais ou próstéticos e são normalmente categorizadas como mão completamente atuadas [2], [3], [4], [5] ou mãos sub atuadas [6], [7], [8].

O primeiro grupo de mãos robóticas contem um maior número de atuadores, tornando-se maiores, mais pesadas, mais caras e de controlo complexo. Ainda assim podem agarrar um grande número de objetos com diferentes características. No entanto, devido aos complexos sistemas eletromecânico e de arquitetura de controle, não são a melhor opção para muitas aplicações onde é necessário que a mão seja simples, leve e barata.

Já o segundo grupo, de mãos sub atuadas, é composto normalmente por apenas 1 a 6 atuadores (comparado com os 34 músculos que compõe uma mão humana) e apresenta as seguintes vantagens em relação ao primeiro grupo:

- Estrutura eletromecânica mais simples.
- Baixo peso, tamanho e preço.
- Arquitetura de controlo mais simples, com um menor número de parâmetros de entrada.

Na categoria de mãos sub atuadas, investigadores continuam a tentar arranjar soluções inovadoras para reduzir a complexidade de controlo da mão tanto quanto possível. Belter et al. [9] discutiram vários trabalhos de investigação onde é notório a procura de próteses mais leves pelos pacientes. Por exemplo, um estudo mostra que 79% dos pacientes considera que a sua prótese é demasiado pesada [10].

Numa análise recente e compreensiva realizada em [11], os autores registaram 7:30h de atividades do dia a dia de duas donas de casa, mostrou-se que apenas 5 *grasps* (de um total de 33) constituíam 80% do total de *grasps* realizadas. Isto faz com que as mãos sub atuadas sejam mais apelativas, pois mostra que este tipo de atuação pode não ser menos funcional do que as mãos completamente atuadas, por exemplo, muitas mãos com três ou menos atuadores, em alguns casos até mesmo com apenas um, podem atingir um elevado número de *grasps* realizadas nas atividades do dia a dia. O principal desafio aqui é aumentar o possível número de pontos de

contacto com um número limitado de atuadores. Para resolver este problema vários estudos têm sido realizadas a nível da mão humana e seu controlo a nível do Sistema Nervoso Central (SNC), que podem ser aplicadas a nível da robótica.

Pesquisas realizadas por vários neurocientistas revelaram que para controlar o mecanismo de movimento de uma mão humana, o SNC não controla cada articulação e músculo individualmente, mas sim em padrões de movimentos predefinidos, intitulados de sinergias.

Neste projeto, pretende-se desenvolver sinergias adaptativas para uma mão robótica antropomórfica sub atuada adaptativa, com o menor número de parâmetros de entrada possível, na medida que esta realize o maior número de *grasps* possíveis. O menor número de parâmetros de entrada propõe-se pois pretende-se implementar o controlo por sensores de EMG, e estes não são capazes de oferecer um grande número de parâmetros de entrada [12].

A mão antropomórfica em questão é a a *ISR-Softhand* [13], uma mão robótica antropomórfica sub atuada que integra complacência nas suas articulações, o que oferece uma boa estabilidade ao agarrar uma larga variedade de objetos. Esta contém três atuadores, um que comanda a flexão do polegar, outro a flexão do dedo indicador e outro que realiza a flexão dos restantes três dedos. A abdução/adução do polegar é feita manualmente.

1.2 Objetivos

Com esta Dissertação de Mestrado pretende-se:

- Reproduzir com a *ISR-Softhand* (um sistema mecânico adaptativo) todas as *grasps* da categorização de Feix et al. (*Human Grasp data base*, disponível no Anexo B e online [14]).
- Categorizar as *grasps* efetuadas.
- Criar o mínimo número de sinergias adaptativas que realizem o máximo de *grasps*, com o menor número de parâmetros de entrada.

- Provar a eficiência da *ISR-Softhand*.
- Analisar e compreender o efeito da perda de um ou mais atuadores da *ISR-Softhand* em termos de:
 - Número de *grasps* possíveis.
 - Complexidade de controlo e número de sinergias.
- Analisar as sinergias adaptativas projetadas e compreender se a redução de um ou mais atuadores reduz a complexidade de controlo, podendo ver respondidas algumas questões:
 - Se a mão fosse controlada por apenas um sinal de entrada (um sinal de entrada que permitisse vários níveis de intensidade), quantas *grasps* seriam possíveis?
 - Para tal sistema quantos atuadores seriam necessários? Tendo em conta a maneira que as sinergias foram projetadas e como as articulações estão acopladas, se a mão fosse construída com menos atuadores afetaria a eficiência da mão em termos de número de *grasps* possíveis?

Sendo a *ISR-Softhand* um sistema adaptativo, permite a criação sinergias adaptativas, de maneira inteligente, para serem testadas e implementadas na mão robótica. O número reduzido de parâmetros de entrada proposto para este projeto, além de simplificar o controlo da *ISR-Softhand* (não sendo necessário controlar cada articulação ou atuador individualmente) tem também como propósito uma futura implementação de sensores de EMG no sistema de controlo da mão robótica.

1.3 Organização da tese

Esta tese está dividida em cinco capítulos. No Capítulo 1 é apresentada uma introdução que revela a principal motivação e objetivo deste projeto.

No Capítulo 2 são apresentados conceitos teóricos relevantes para a compreensão desta dissertação, bem como o estado da arte em relação às mãos prótesicas e sinergias já desenvolvidas.

No Capítulo 3 é descrita a mão robótica antropomórfica sub atuada utilizada neste projeto, a *ISR-Softhand*. Mostra as suas características mecânicas bem como as especificidades dos seus atuadores.

No Capítulo 4 são descritas e categorizadas as *grasps* possíveis de realizar pela *ISR-Softhand*, bem como as sinergias correspondentes.

Por fim, o Capítulo 5 conclui esta dissertação. Aqui são discutidas estratégias de simplificação da mão robótica, a nível de número de atuadores e sinergias, e como estas estratégias podem ser implementadas no futuro.

Capítulo 2

Conceitos Teóricos e Estado da arte

Este Capítulo está dividido em duas importantes secções, os conceitos teóricos e o estado da arte. O primeiro descreve os conceitos básicos para o entendimento desta dissertação e o segundo faz um apanhado dos desenvolvimentos históricos à volta dos temas abordados.

2.1 Conceitos teóricos

Nesta secção são aprofundados o seguintes conceitos:

2.1.1 Anatomia da mão humana

2.1.2 Mãos robóticas

2.1.21 Antropomorfismo

2.1.22 Destreza

2.1.23 Cinemática

2.1.24 Mecanismos de atuação

2.1.25 Software de programação

2.1.3 Sinergias

2.1.31 Sinergias rígidas

2.1.32 Sinergias *soft*

2.1.33 Sinergias adaptativas

2.1.4 *Grasps*¹

2.1.41 Direção da força aplicada

2.1.42 *Power grasps*, *grasps* intermédias ou de precisão.

¹termo inglês que se refere ao momento final do movimento efetuado quando se agarra um objeto

2.1.1 Anatomia da mão Humana

Esta sub-seção descreve de forma simplificada, a anatomia de uma mão humana, para que se consiga compreender as semelhanças entre o design mecânico da mão robótica utilizada nesta dissertação com uma mão humana.

A Figura 2.1 mostra a anatomia de uma mão humana, onde se pode ver em pormenor o nome dos ossos (a cores) e o nome das articulações que ligam os ossos (a cinzento). Esta figura servirá para ao longo do Capítulo 3 se perceber a estrutura da mão robótica e comprovar a sua igualdade a uma mão humana.

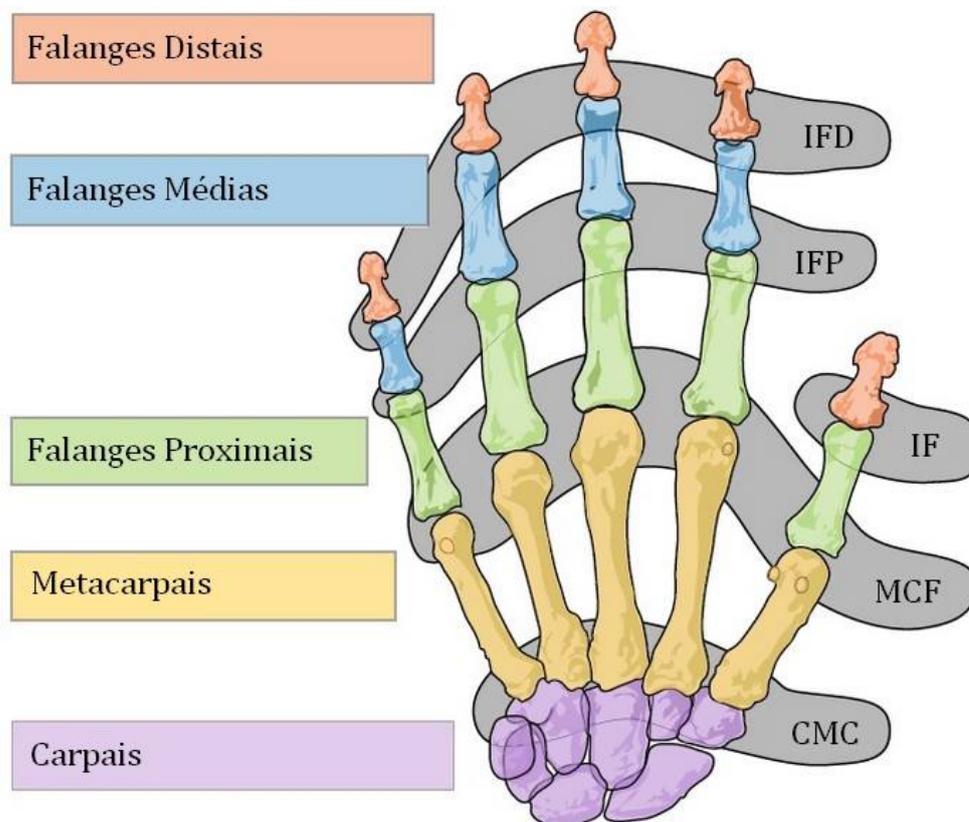


Figura 2.1: Ossos da mão humana (a cores) e respetivas articulações (a cinzento). Adaptado de [15].

A palma da mão é formada por cinco ossos metacarvais. A articulação CarpoMetaCarpiana (CMC) é aquela que permite a palma movimentar-se em arco. Devido ao pequeno movimento efetuado pela articulação CMC, apenas de 10 a 15 mm [16], a maioria dos modelos de mãos robóticas desenvolvidas são constituídas por uma palma rígida.

Em relação à articulação MetaCarpoFalangiana (MCF), esta permite movimentos de: flexão/extensão e abdução/adução de todos os dedos. Na Figura 2.2 está representado, esquematicamente da esquerda para a direita, o máximo de flexão (pode ir até aos 90° e aumenta desde o dedo indicador ao mindinho), o máximo de flexão ativa que varia deste os 30° e os 40° , e o movimento de abdução(B)/adução(B) do dedo indicador.

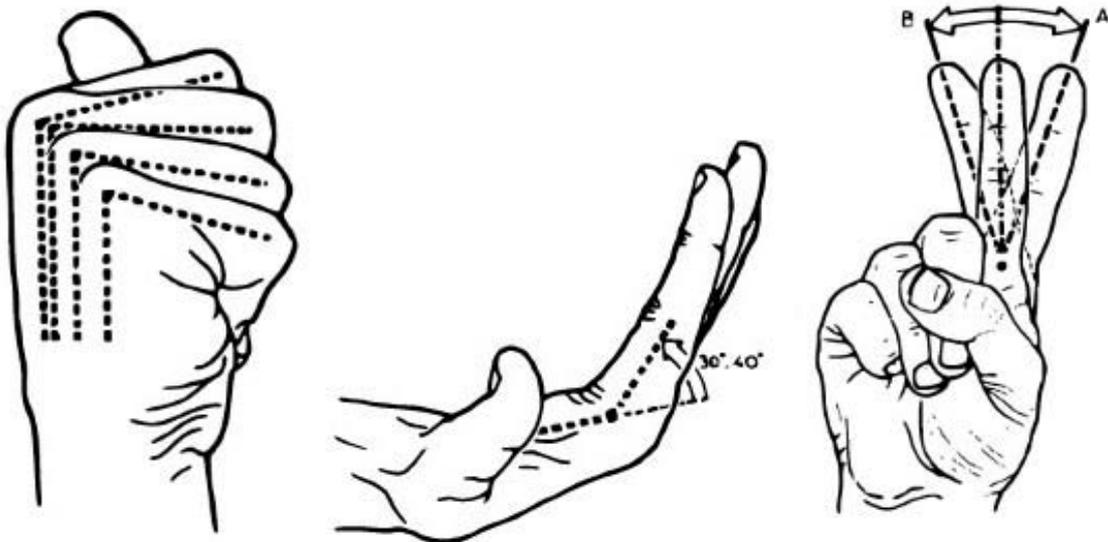


Figura 2.2: Movimentos realizáveis pela articulação MetaCarpoFalangiana (MCF). Da esquerda para a direita: Flexão da articulação MCF; Extensão dos dedos; Movimento de abdução/adução do dedo indicador [17].

De referenciar que quando se refere a dedos, refere-se aos dedos indicador, médio, anelar e mindinho. O polegar, pela sua diferenciação anatómica, como se visualiza na Figura 2.1, é comentado separadamente.

Dedos: Cada dedo apresenta 4 Graus de Liberdade (*Degrees of freedom*) (DoFs), pois além dos dois graus de liberdade conferidas pela articulação MCF, esta apresenta ainda mais duas articulações, a articulação InterFalangiana Proximal (IFP) e a articulação InterFalangiana Distal (IFD), que lhe conferem mais dois graus de liberdade. A primeira permite um movimento de flexão até aos 135° e a segunda permite um movimento de flexão até aos 90° [16] (característica importante para quando se agarra objetos com a ponta dos dedos) como se visualiza na Figura 2.3.

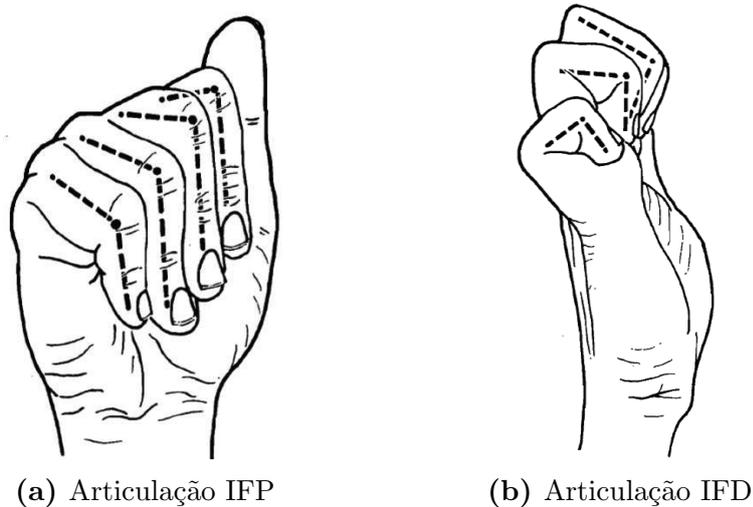


Figura 2.3: Movimentos realizáveis pelas articulações InterFalangiana Proximal (IFP) e InterFalangiana Distal (IFD) de flexão/extensão [17].

Polegar: O polegar tem um papel bastante importante na maioria das *grasps* efetuadas pelo Homem, sendo utilizado ativamente em 93,94% da categorização das principais *grasps* efetuadas pelo Homem, usada neste projeto (Anexo B).

O Polegar difere dos restantes dedos, pois apenas apresenta uma articulação InterFalangiana (IF) e tem mais um Grau de Liberdade (*Degree of freedom*) (DoF) do que os dedos, devido à diferente mobilidade que as articulações MCF e CMC conferem a este dedo, pois contêm dois eixos de deslocação perpendiculares, que lhes permite o movimento de flexão(Figura 2.4a)/extensão(Figura 2.4b) e abdução(Figura 2.4c)/adução(Figura 2.4d) [18].

A articulação IF tem apenas um grau de liberdade (permite apenas flexão/extensão da ponta do polegar). Em suma, o polegar apresenta 5 DoFs, permitindo-lhe adequar-se à sua importante funcionalidade.

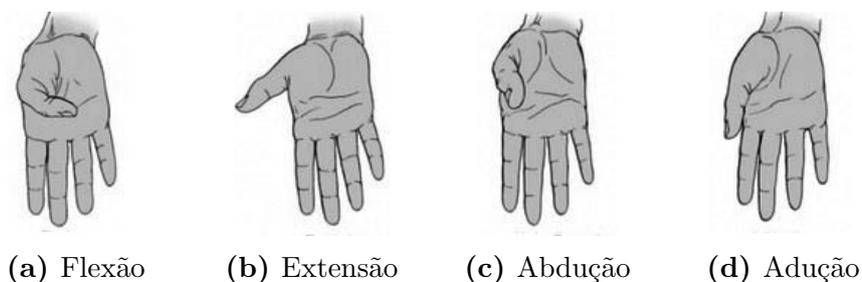


Figura 2.4: Movimentos realizáveis pelas articulações MetaCarpoFalangiana (MCF) e CarpoMelaCarpiana (CMC) do polegar [19].

2.1.2 Mãos Robóticas

Desde o século *XVIII* que a biomecânica da mão humana tem sido estudada para poder ser aplicada como modelo de mãos robóticas [6]. Os avanços na área da robótica permitiram então desenvolver um sistema capaz de replicar os movimentos realizáveis pela mão humana.

Esta sub-secção descreve de forma básica e compreensível vários conceitos essenciais aquando o desenvolvimento e controlo de mãos robóticas.

2.1.2.1 Antropomorfismo

O termo antropomorfismo descreve a "*capacidade da extremidade do robô em imitar uma mão humana, parcialmente ou totalmente, em relação à forma, tamanho, consistência e aspeto geral (cor, temperatura, etc)*" [20]. Resumidamente, uma mão robótica é tanto mais antropomórfica quanto mais idêntica a uma mão humana, sem que seja considerada a sua funcionalidade.

2.1.2.2 Destreza

O termo destreza "*relaciona-se com a funcionalidade da mão e não com a forma e outros fatores estéticos*" [20].

Ou seja, uma mão que possui destreza é aquela que consegue realizar uma maior quantidade de movimentos a que se propõe, é mais hábil. No entanto, muitas vezes para uma mão possuir destreza necessita de um design mecânico mais complexo, podendo recorrer ao uso de mais motores e articulações, tornando-se não utilizável a nível prótico. Um dos desafios aquando a conceção de uma mão robótica prótica é então conseguir um equilíbrio entre o nível de destreza e antropomorfismo.

2.1.2.3 Cinemática

O estudo do movimento pode ser dividido em dois grupos, a cinemática e dinâmica. Este trabalho foca-se principalmente em controlo de parâmetros de entrada relacionados com a cinemática da *ISR-SoftHand*, daí o foco principal neste ponto.

A cinemática refere-se ao movimento do robô, sem ter em consideração as forças e/ou binários [21](área em que se foca a dinâmica [22]). Ou seja, é o estudo da posição, velocidade e aceleração quer da(s) extremidade(s), quer das articulações do robô. Entenda-se como articulação a junção das partes rígidas de um robô. A respetiva geometria da articulação define o número de DoF do robô, e por sua vez a cinemática da articulação.

DoF é um termo normalmente utilizado para descrever a liberdade de movimento do robô pelo espaço tridimensional. Uma articulação para ser completamente versátil necessita de 6 DoFs (Figura 2.5) permitindo lhe movimentos de [23]:

- Rotação (θ):
 - Em torno do eixo de x ;
 - Em torno do eixo de y ;
 - Em torno do eixo de z .
- Translação:
 - No eixo de x ;
 - No eixo de y ;
 - No eixo de z .

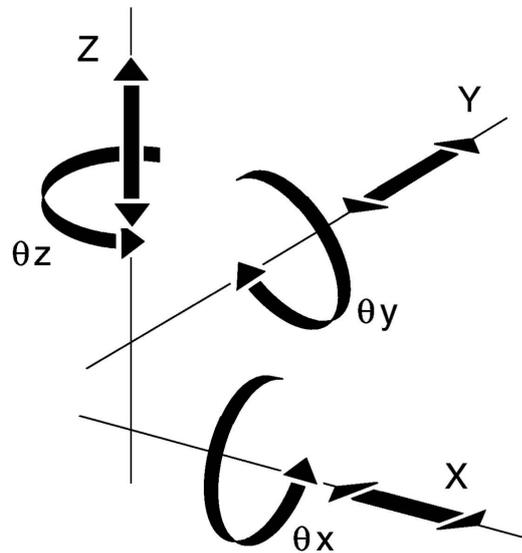


Figura 2.5: Representação dos 6 possíveis graus de liberdade de uma articulação. Adaptado de [24].

No entanto o termo de DoF por vezes ganha uma nova definição na área do desenvolvimento de mãos robóticas sub atuadas. Nesta área muitas vezes diz-se que o número de articulações de uma mão robótica define o seu número de DoFs. Por exemplo, a mão sub atuada PISA/IIT (descrita na secção 2.2.1.2) contém 19 articulações (Figura 2.6) e é comandada por apenas um atuador no entanto é descrita como tendo 19 DoFs. Seria de esperar que este mecanismo fosse considerado como tendo apenas 1 DoF tendo em conta que quando uma articulação depende de outra o seu grau de liberdade deixa de ser considerado.

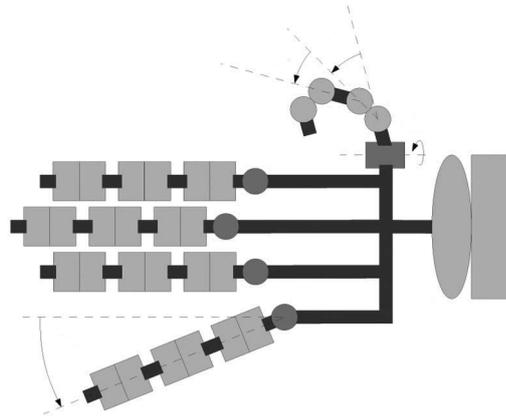


Figura 2.6: Esquema das 19 articulações que compõe a *Pisa/IIT SoftHand* [25].

Considerando esta definição, sempre que for referido o termo DoF aquando a descrição de mãos sub atuadas será mencionado se os autores se referem, ou não, ao número total de articulações ou ao número de articulações com movimento independente.

2.1.2.4 Mecanismo de atuação

A estrutura física do robô é denominada estrutura mecânica ou mecanismo do robô. Os motores e/ou outros elementos que provocam o movimento da estrutura mecânica são denominadas de atuador. Os atuadores fornecem força eletromotriz aos robôs. Existem três tipos principais de atuadores: hidráulicos, pneumáticos e eletromagnéticos [26].

Um mecanismo de atuação de um robô pode ser categorizado em 3 classes, pela relação do número de DoFs e ao número de Graus de Atuação (*Degrees of actuation*) (DoAs) (número de atuadores):

1. Mecanismo completamente atuado

$$DOA = DOF \tag{2.1}$$

2. Mecanismo redundantemente atuados

$$DOA > DOF \quad (2.2)$$

3. Mecanismo sub atuado

$$DOA < DOF \quad (2.3)$$

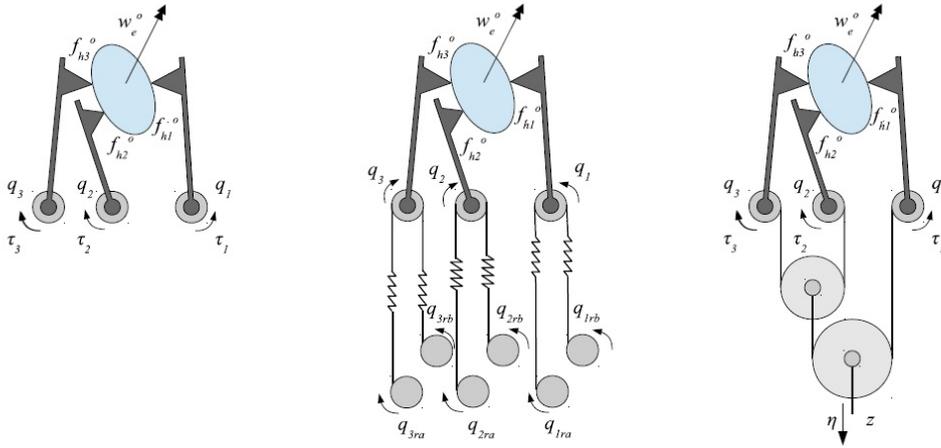
Na Figura 2.7 estão representados 6 tipos de atuação utilizados e desenvolvidos por investigadores na área das mãos robóticas protéticas [27]. Na Tabela 2.1 encontram-se as definições de cada uma das variáveis visíveis na Figura 2.7.

Tabela 2.1: Variáveis utilizadas para a análise dos sistemas de atuação.

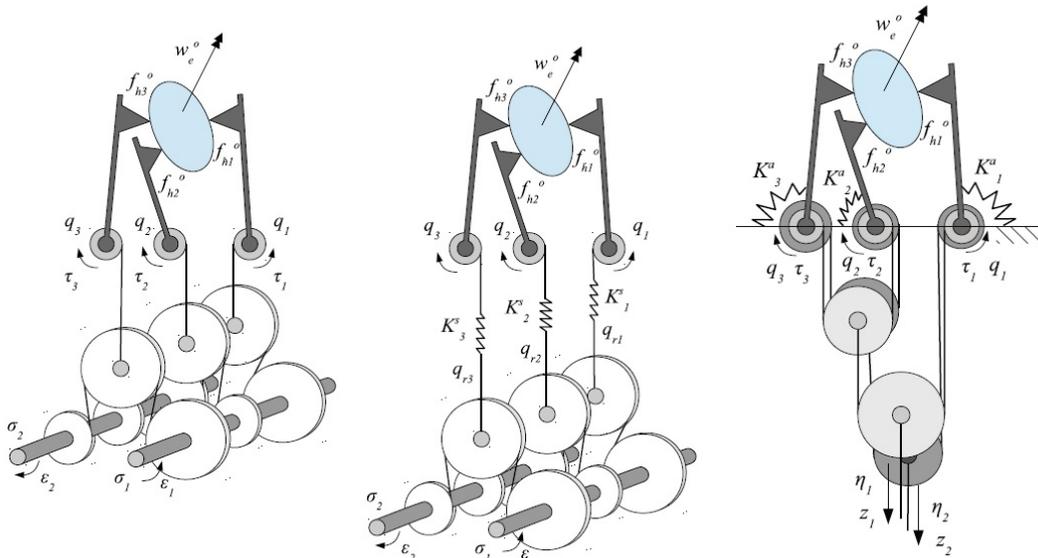
Variável	Definição
$W_e^o \in \mathbb{R}^6$	Força externa aplicada no objeto
$f_h^{c^o} \in \mathbb{R}^c$	Força de contacto, pela mão, no objeto
c	Número de zonas de contacto
$q \in \mathbb{R}^{\#q}$	Posição da articulação
$q_r \in \mathbb{R}^{\#q}$	Posição de referência da articulação
$\tau \in \mathbb{R}^{\#q}$	Binário da articulação
$\sigma \in \mathbb{R}^{\#\sigma}$	Posição da sinergia <i>soft</i>
$\varepsilon \in \mathbb{R}^{\#\sigma}$	Força da sinergia <i>soft</i>
$z \in \mathbb{R}^{\#z}$	Movimento da sinergia adaptativa
$\eta \in \mathbb{R}^{\#z}$	Força da sinergia adaptativa
K	Rigidez

Na Figura 2.7a observa-se o caso mais simples, onde cada articulação é controlada por um atuador (mecanismo completamente atuado, equação (2.1)). Aqui o controlo é simplificado, no entanto é um modelo caro, pesado (por ter que utilizar três atuadores) e de controlo complexo. Neste caso quanto mais rígido for o objeto que se pretende agarrar, mais fácil é o controlo da posição de contacto na extremidade do objeto, no entanto nem todos os objetos têm esta característica, podendo levar a danos no objeto se a força de contacto for excessiva [29].

Para evitar problemas deste tipo aquando a interação com objetos, elementos complacentes são incorporados no robô. A complacência nas mãos robóticas permite uma maior possibilidade de criação pontos de contacto da mão com objetos de diferentes



(a) Completamente atu- (b) Atuador de rigidez (c) Sub atuada adapta-
 ada [27]. variável [27]. tiva [27].



(d) Sinergia rígida [27]. (e) Sinergia *soft* [27]. (f) Sinergia adapta-
 tiva [28].

Figura 2.7: Esquema de mão robótica com 3 dedos, simplificada, com a implementação de diferentes paradigmas de atuação.

dimensões, com a mesma configuração de posição das articulações (q). Ou seja, ao adicionar complacência ao sistema é possível reduzir o número de articulações atuadas e permite reduzir a complexidade de controle.

Este comportamento complacente pode ser conseguido quer passiva como ativamente [30], permitindo controlar a força do atuador e levando conseqüentemente, a mão adaptar-se ao objeto. Na complacência ativa, o extremidade do robô é vista como

uma mola, onde a rigidez (K) é controlada pelo software utilizado para o controle do movimento [30]. Na complacência passiva, adiciona-se elementos complacentes ao sistema mecânico que variam a rigidez (K) do sistema, como é visível nas Figuras 2.7b, 2.7e e 2.7f.

Na Figura 2.7b existe uma adição de elementos complacentes (em pares, em cada uma das articulações) que variam a rigidez (K) do sistema. Aqui é dado ao sistema valores de posição de referência (q_r) e a sua diferença à verdadeira posição das articulações (q) cria uma complacência no sistema, fazendo com que a mão se adapte aos objetos. No entanto este modelo criado por Grebenstein et al. [31] não apresenta potencialidade de simplificação mecânica, pela necessidade da utilização de dois atuadores por cada articulação [25].

Na Figura 2.7e já existe um sistema de sub atuação, onde existe um menor número de atuadores do que articulações (ou seja, $\#q > \#\sigma$). Na prática, consiste em juntar complacência em série, ou seja no próprio atuador, no modelo da Figura 2.7d [27]. O modelo da Figura 2.7d, foi apresentado inicialmente proposto por Brown et Asada [32] representa o modelo adaptativo mais básico onde existe a combinação de vários elementos para a criação do máximo de movimentos com a redução do número de atuadores, ou seja, sinergias (termo explicado em pormenor mais á frente, na secção 2.1.3). Neste o número de DoFs é menor que DoAs, e as roldanas existente têm diferentes raios para dar diferentes movimentos a cada articulação, permitindo que a mão sub atuada seja o mais eficiente possível. No entanto este modelo não é muito utilizado, dado à rigidez do sistema, que pode ser um problema aquando a utilização de diferentes objetos com diferentes características, como já referido. [25].

Na Figura 2.7f adicionam-se elementos passivos ao modelo da Figura 2.7c em paralelo com as articulações, permitindo controlar forças. É um modelo de pequeno tamanho, leve e barato, mas os seus fundamentos teóricos ainda não estão muito bem explicados pela comunidade científica. No entanto a implementação deste *design* mecânico leva a bons resultados, como se comprova em diversos estudos realizados, por exemplo, por Grioli et al. [28], Catalano et al. [27], Ra et al. [33] ou Dollar et

al. [34].

As roldanas (Nas Figuras 2.7c, 2.7d, 2.7e e 2.7f) permitem que um controlo simplificado ou seja, apenas com o movimento de uma ou duas roldanas movem-se as 3 articulações, tornando a mão sub atuada (equação(2.3)) [6].

Na Figura 2.7c está representado o caso mais simples e eficaz de sub atuação, onde existe a distribuição do deslocamento por duas roldanas, onde apenas uma é atuada [27]. Aqui o sistema é capaz de se adaptar a diferentes objetos apenas pela implementação de roldanas com diferentes tamanhos conseguindo que a posição(q) e binário(τ) das articulações dependa do movimento (z) e força (η) conferida pela sinergia adaptativa, por um fator que pode ser pensado pelo utilizador aquando o *design* do sistema. No entanto neste modelo de atuação é necessário um estudo meticoloso do valor de rotação na roldana para que as forças de contacto com o objeto não sejam excessivas [25], daí ter surgido o modelo da Figura 2.7f.

O modelo representado na Figura 2.7f corresponde aquele que por objetivo se pretende implementar na mão próstética em desenvolvimento.

O termo sinergia, utilizado para descrever os modelos de atuação das Figuras 2.7c, 2.7d e 2.7e, será explorado na próxima secção(secção 2.1.3).

2.1.2.5 Software de programação

Os softwares de controlo de sistemas robóticos robustos com um elevado número de articulações, como é o caso das mãos robóticas, tendem a ser complexos. Esta complexidade deve-se à necessidade de controlar diversos sensores e atuadores em tempo real. O uso de arquiteturas já conhecidas/desenvolvidas pode simplificar esta tarefa [35]. Nesta dissertação foram utilizados atuadores da ROBOTIS, já com uma arquitetura de controlo incorporada e software próprio. No Capítulo 3 esta questão será abordada com maior pormenor.

2.1.3 Sinergias

O termo sinergia, já referido anteriormente, define a junção de vários elementos para uma realização de uma tarefa comum e pode ser usado em vários contextos.

Esta dissertação foca-se principalmente no controlo complexo da mão humana, um campo que vários investigadores na área da neurociência já tentaram analisar. É ainda considerado uma incógnita como é que o SNC consegue controlar os 21 DoFs da mão humana. Uma explicação para este fenómeno reside no facto de que em sistemas vivos complexos, como é o próprio ser humano, os elementos estão organizados em sinergias, ou seja, agrupamentos temporários de elementos estruturais (isto é, neurónios, músculos e articulações) para uma única finalidade [36], como o movimento da mão para agarrar um objeto.

Percebe-se então que uma sinergia pode ser vista como uma ferramenta de redução de complexidade de controlo, já utilizada na área da robótica, como referido na secção 2.1.2.4 (Figuras 2.7d (Sinergia rígida), 2.7e (Sinergia Suave) e 2.7f (Sinergia adaptativa)) permitindo controlo do maior número de elementos com os menores comandos de parâmetros de entrada possíveis [37].

Resumidamente, as sinergias quando aplicadas a mãos robóticas conseguem reduzir a complexidade de controlo, podendo torna-las mais leves e simples de manusear.

De seguida são descritas, de forma resumida, os três tipos de sinergias mais utilizados aquando o controlo de mãos robóticas protéticas.

2.1.3.1 Sinergias Rígidas

O caso mais simples da sinergia, onde existe a combinação de elementos de maneira a que seja necessário controlar menos parâmetros de entrada do que aqueles de saída. O problema é fazer com que o sistema se adapte ao objeto que está a agarrar, criando o máximo de pontos de contacto para que a *grasp* seja o mais estável possível, daí a necessidade de criar outros tipos de sinergias com elementos complacentes que permitam tal adaptabilidade.

2.1.3.2 Sinergias *soft*

Estas sinergias têm como base a implementação de complacência em série com o atuador na mão robótica. Esta sinergia foi implementada com o intuito de mimetizar a complacência de uma mão humana, que lhe é conferida pelo sistema músculo-tendinoso [37]. No entanto este modelo de atuação torna-se complexo na sua implementação prática, levando que à necessidade de juntar a complacência de outra maneira na mão robótica [25].

2.1.3.3 Sinergias Adaptativas

Permitem explorar eficazmente a aplicação de sinergias em mão robóticas sub atuadas. Consiste na adoção de elementos complacentes em paralelo com as articulações, permitindo que estas se adaptem ao objeto sem qualquer necessidade de controlar o nível de complacência. Ou seja, adiciona-se elementos complacentes nas próprias articulações da mão robótica (como borrachas, vistas como um elemento elástico).

Assim basta implementar pontos de contacto virtuais, que podem ser considerados dentro dos objetos, que a mão consegue adaptar-se a vários objetos com forma e tamanho idêntico, sem que a mão exerça força excessiva que danifique o objeto [25].

2.1.4 *Grasps*

A mão humana é pela sua anatomia um sistema complexo, tornando-se mais complexo o estudo do seu funcionamento. Esta pode ser usada numa vasta variedade de tarefas do dia a dia do Homem, desde agarrar objetos até expressar emoções.

Dado que esta dissertação se foca no *design* de sinergias adaptativas para que a mão consiga agarrar o máximo de objetos possíveis, esta subsecção vai descrever os termos básicos usados no *grasping* ².

Primeiramente os movimentos realizados pela mão humana com a interação com objetos foram classificados em dois grandes grupos [38]: Movimentos preênsais e movimentos não preênsais. Um movimento preênsil acontece quando um objeto é agarrado

²termo inglês que define todo o movimento que envolva o agarrar de objetos

parcial ou totalmente dentro da mão humana de forma estável. Um movimento não preênsil é aquele onde não se agarra o objeto mas consegue-se manipulado, tal como empurrá-lo ou levantá-lo, com toda a mão ou até com apenas os dedos/polegar individualmente.

Para o *design* das sinergias adaptativas e estudo da eficiência da *ISR-SoftHand* apenas foram tomados como relevantes os movimentos preênsis, logo todos as restantes funcionalidades da mão humana que não incluam a *grasp* de um objeto não foram tidas em conta.

Todas as categorizações desenvolvidas em relação a estes movimentos serão vistas em detalhe na Secção 2.2.2. Nesta secção apenas se descreverá os diferentes tipos de definições existentes dentro dos movimentos preênsis, em relação à direção da força aplicada e em relação à maneira como são agarrados os objetos.

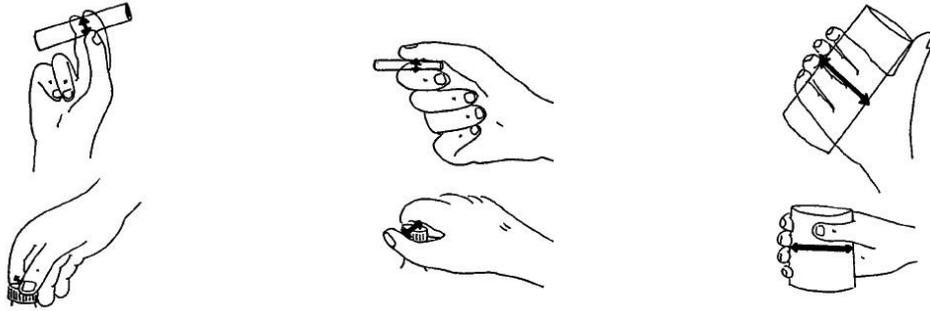
2.1.4.1 Direção da força aplicada

Existem 3 direções básicas de aplicação de força num objeto de forma a que ele seja agarrado com a maior estabilidade [39]. Nas Figuras 2.8a, 2.8c e 2.8b estão representadas as 3 direções de aplicação de força referidas.

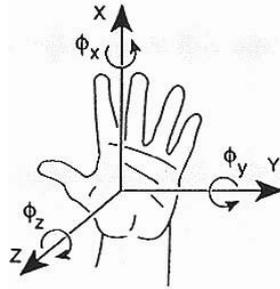
A força aplicada pelas pontas dos dedos ocorre na direção paralela à palma da mão (Figura 2.8a) correspondente ao eixo dos x representado na Figura 2.8d e envolve normalmente o uso da zona desde a palma à ponta dos dedos e do polegar.

A força aplicada pela zona lateral dos dedos (Figura 2.8b) ocorre na direção transversal à palma, correspondente ao eixo dos y na Figura 2.8d.

A força aplicada pela palma da mão (Figura 2.8c) ocorre na direção perpendicular à palma, correspondente ao eixo dos z visualizado na Figura 2.8d e envolve sempre a interação da palma ao agarrar objetos.



(a) Pontas dos dedos [39]. (b) Zona lateral dos dedos [39]. (c) Palma das mãos [39].



(d) Sistema de eixos (x , y e z) em relação à mão humana [17].

Figura 2.8: Direções básicas da força realizada pela mão para a execução das *grasps*.

2.1.4.2 *Power grasps*, *grasps* intermédias ou *grasps* de precisão

As *grasps* em relação à maneira que se agarra os objetos estão descritas em 3 grandes grupos [38]: *Power³ grasps*, *grasps* de precisão ou *grasps* intermédias.

Este tipo de classificação é a mais importante, pois é nesta onde a maioria dos autores se baseia para a classificação das *grasps*, como se irá perceber mais à frente na secção 2.2.2.

A Figura 2.9 mostra como a postura da mão muda ao longo do movimento de abertura da mão. No início, quando o frasco encontra fechado é necessário mais força e mais área de contacto da mão com o frasco, ou seja, é necessário utilizar uma *power grasp*. Quando a tampa fica mais solta a *grasp* fica diferente, para que se possa abrir o frasco mais rapidamente, não sendo necessária tanta força e área de contacto, utilizando-se então uma *grasp* de precisão.

³termo inglês propositadamente usado dado a sua pobre tradução

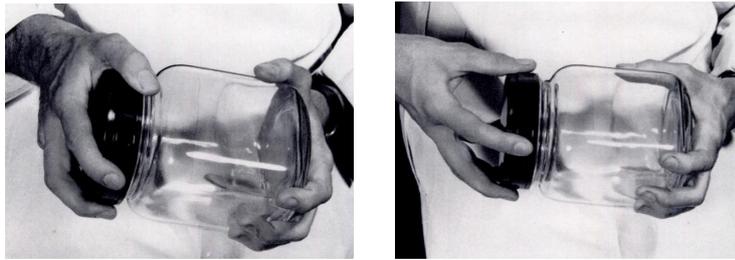


Figura 2.9: Sequência de movimentos da mão humana ao abrir um frasco. Evolução do uso da *power grasp* para *grasp* de precisão [38].

A ideia de distinção de tipos de *grasps* foi inicialmente descrita por Landsmeer [40], que distinguiu entre “agarrar com força” e “manusear com precisão”. Este definiu “agarrar com força” quando existe uma relação rígida entre o objeto e a mão, ou seja, quando a única maneira de mover o objeto é movendo o braço. No “manuseamento com precisão” o movimento do objeto pode ser realizado por movimentos da própria mão (ou seja, movimentos dos dedos, polegar ou palma).

No entanto nesta dissertação não se dá importância ao movimentar do objeto, apenas à *grasp* em si, estática e estável. Dado isto, assume-se que as *grasps* de precisão efetuadas com a *ISR-SoftHand* são estáveis.

Kamakura [41] introduziu mais tarde o termo de *grasps* intermédias. Aqui existe uma igual contribuição de *power grasp* e de *grasp* de precisão.

De seguida, na Figura 2.10 estão representados três esquemas correspondentes aos três tipos de *grasps*:

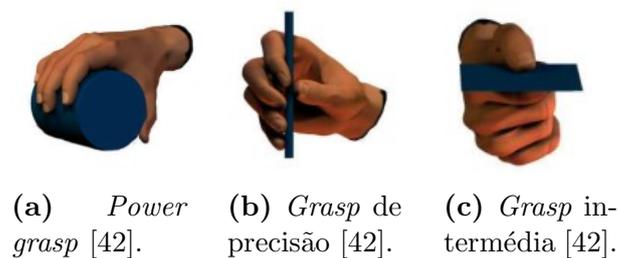


Figura 2.10: Visualização de cada um dos tipos de *grasps*: *Power*, precisão e intermédia.

2.2 Estado da arte

Nos últimos tempos têm sido desenvolvidos trabalhos e investigações relacionadas com o design e controlo de mãos protéticas robóticas.

Esta secção será então focada no estado da arte existente em relação às seguintes matérias:

2.2.1 Mãos robóticas protéticas sub atuadas

2.2.1.1 Mãos robóticas protéticas no mercado

2.2.1.2 Mãos robóticas protéticas em desenvolvimento

2.2.2 Categorização das *grasps*

2.2.2.1 Schlesinger (1919) [43]

2.2.2.2 Napier et al. (1956) [38]

2.2.2.3 Kapandji et al. (1982) [44]

2.2.2.4 Elliot and Conolly (1984) [45]

2.2.2.5 Cutkosky (1989) [46]

2.2.2.6 Feix et al. (2011) [16] [14]

2.2.3 Estratégias de simplificação de controlo

2.2.3.1 Sinergias adaptativas

2.2.3.2 Controlo por EMG

2.2.1 Mãos robóticas protéticas sub atuadas

Já desde os meados dos anos 80 que tem sido estudado o fabrico de mãos robóticas baseadas no funcionamento da biomecânica da mão humana [2]. Este tipo de mãos robóticas antropomórficas desenvolvidas para a indústria ou para fins protéticos podem ser completamente atuadas (como [2], [3], [4] ou [5]) ou sub atuadas (como [6], [7], [8]).

Na categoria de mãos sub atuadas, os investigadores têm vindo a tentar encontrar soluções inovadoras para a redução da complexidade da mão. Além da simplicidade de controlo, hoje em dia também se procura o desenvolvimento de uma mão com cada vez menos atuadores, em alguns casos mesmo com apenas um, que consigam realizar o máximo de *grasps* que usualmente são realizados nas tarefas do dia a dia [11]. O maior problema das mãos sub atuadas, é conseguir o máximo de pontos de contacto da mão com o objeto. Para isso existe então, o estudo da mão humana e suas sinergias.

2.2.1.1 Mãos robóticas protéticas no mercado

Já existem no mercado várias mãos protéticas robóticas disponíveis. Cinco dessas mãos estão descritas na Tabela 2.2.1.2 bem como as suas principais características. Estas podem ser vistas na Figura 2.11.



Figura 2.11: Mãos protéticas disponíveis no mercado.

2.2.1.2 Mãos robóticas protéticas em desenvolvimento

Em relação às mãos robóticas ainda em desenvolvimento foram escolhidas cinco desenvolvidas nos últimos 10 anos e estão descritas na Tabela 2.2.1.2. Na Figura 2.12 podem visualizar-se estas mãos robóticas.

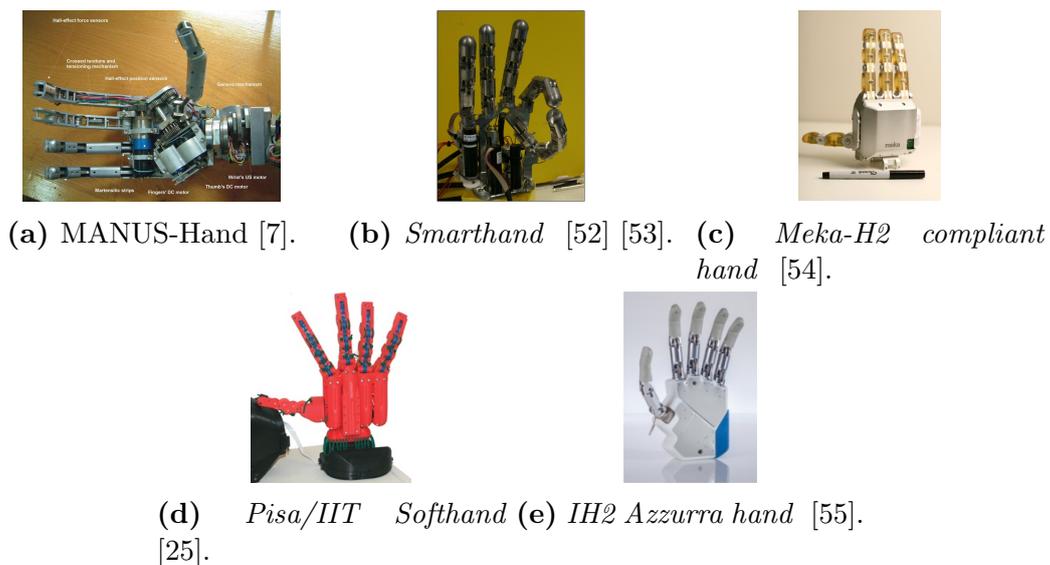


Figura 2.12: Mãos protéticas em desenvolvimento.

Tabela 2.2: Características gerais de 5 mãos próstéticas robóticas disponíveis no mercado.

	Criadores	DoF	DoA	Características importantes
Sensorhand [47]	Ottobock inc.	1	1	O atuador move o polegar, o dedo indicador e o anelar. Os restantes dedos são para a tornar antropomórfica. Não tem a capacidade de se adaptar a objetos (Figura 2.11a).
Michelan-gelo [48] [56]	Ottobock inc.	2	2	1 atuador para fechar todos os dedos e polegar e outro que muda a posição de abdução/adução (ab/ad) do polegar. Não tem a capacidade de se adaptar a objetos (Figura 2.11b).
iLIMB [49] [57]	Touch Bionics	6	5	1 atuador por cada dedo, que atua na articulação MCF. O movimento de ab/ad do polegar é manual, podendo adotar duas posições com 90° entre elas. Tem a capacidade de se adaptar a objetos (Figura 2.11c).
Bebionic [50] [57]	RSL Steeper	6	5	1 atuador por cada dedo. O movimento de ab/ad do polegar é manual. A diferença em relação à iLIMB está na localização dos motores, nesta estão localizados na zona do metacarpo(Figura 2.11d).
Vicent hand [51] [57]	Vincent System	6	6	1 atuador por cada dedo, o polegar tem 2 atuadores, um para a sua flexão e outro para o movimento de ab/ad. Tem a capacidade de se adaptar a objetos (Figura 2.11e).

Tabela 2.3: Características gerais de 4 mãos próstéticas sub atuadas em desenvolvimento.

		Características importantes	
	DoF	DoA	
MANUS-hand [7](2004)	3	3	Inclui 3 atuadores e uma roldana. Um atuador para o movimento do polegar, um para os restantes dedos e outro para o movimento do pulso. (Figura 2.12a).
Smarthand [52] [53] (2009)	16*	4	4 atuadores, todos colocados dentro da palma da mão. 3 atuadores comandam o movimento do polegar, dedo indicador e os restantes dedos, independentemente. O quarto atuador atua no movimento de abdução/adução (ab/ad) do polegar. A flexão/extensão (flex/ext) da articulação MCF está ligada à extensão do motor de Corrente Contínua. (Figura 2.12b).
Meka-H2 compliant hand [54] (2009)	12*	5	Acoplamento de molas (elemento complacente) em série com o atuador, ou seja, existe uma mola entre o tendão e o motor. Tem apenas 4 dedos, não sendo completamente antropomórfica. Cada um dos dedos é controlado por um destes atuadores e outro controla o movimento de ab/ad do polegar (Figura 2.12c).
Pisa/IIT Softband [25] (2014)	19*	1	1 atuador controla todos os dedos. Tem complacência no <i>design</i> mecânico que lhe confere um alto nível de adaptabilidade, não deixando de ser muito robusta. Baseia-se em sinergias adaptativas para se conseguir adaptar ao máximo de objetos possíveis (Figura 2.12d).
IH2 Azzurra hand [55] (2014)	11*	5	1 atuador controla o movimento de ab/ad do polegar, outro a sua flex/ext, outro a flex/ext do dedo indicador, outro a flex/ext do dedo médio e por fim, o 5º atuador acopla os movimentos de a flex/ext dos restantes dois dedos. Contem sensores de força e posição, dedos complacentes e cada dedo é sub atuado (Figura 2.12e).

* Neste caso DoF refere-se ao número de articulações, como explicado na secção 2.1.2.3.

2.2.2 Categorização das *grasps*

Como já referido na secção 2.1.4, a categorização das *grasps* já vem a ser estudada à bastantes anos, sendo a primeira categorização conhecida datada de 1919 [43], tornando-se ao longo do tempo mais completas e claras por um vasto número de investigadores. Nesta secção serão descritas algumas das principais categorizações existentes e será dado ênfase àquela utilizada para testar e implementar as sinergias adaptativas na *ISR-Softhand*.

2.2.2.1 Schlesinger (1919) [43]

A primeira categorização realizada, onde são datados seis diferentes *grasps*: cilíndrica, pontas dos dedos, gancho, palmar ,esférica e lateral (Figura 2.13). As *grasp* eram na sua maioria definidas pelo tipo de objetos que se agarrava.

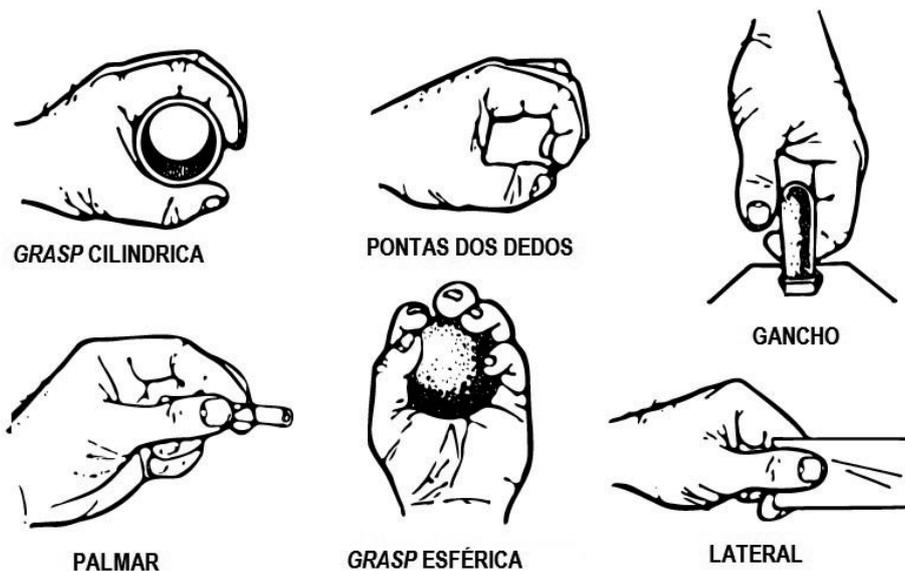


Figura 2.13: A Categorização de Schlesinger. Adaptade de [58].

2.2.2.2 Napier et al. (1956) [38]

Primeira classificação que tem em conta a função da *grasp* e não à sua aparência. Foram os pioneiros na introdução dos termos de *power grasp* e *grasp* de precisão (como já descrito na secção 2.1.4).

2.2.2.3 Kapandji et al. (1982) [44]

Nesta categorização existem várias sub-categorias de *grasps*, consoante estas são ou não estáticas. No entanto, como esta dissertação se foca na criação de sinergias para *grasps* estáticas, apenas esta sub-categoria é considerada.

As *grasps* são classificadas pela zona da mão que está em contacto com o objeto, que podem ser, quer os dedos, a palma ou ambos, formando *grasps* simétricas. Dentro destas categorias existem ainda sub-categorias em relação ao número de dedos que se utiliza. Nesta categorização estão incluídas 24 *grasps*.

2.2.2.4 Elliott and Connolly (1984) [45]

Aqui são descritas três classes gerais de movimentos de manipulação com a mão humana: sinergias simples, sinergias recíprocas e padrões sequenciais (Figura 2.14). As sinergias simples são aquelas em que todos os dedos e o polegar estão envolvidos na *grasp* e se movem em conjunto. Nas sinergias recíprocas os quatro dedos movem-se juntos mas o polegar tem um movimento independente. Nos padrões sequenciais os dedos e polegar tem padrões de movimento independente mas numa sequência repetitiva.

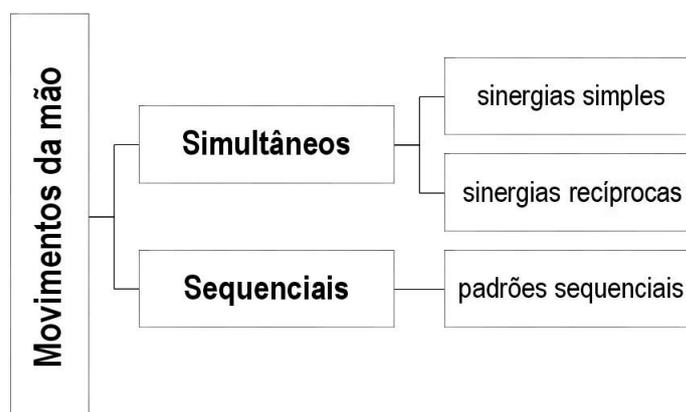


Figura 2.14: Lógica utilizada na categorização de Elliott e Connolly.

2.2.2.5 Cutkosky (1989) [46]

Nesta categorização 16 *grasps* foram tidas em conta. Foi desenvolvida com base em estudos em parceria com mecânicos profissionais e no trabalho já realizado por

Schlesinger e Napier. A Categorização de Cutkosky está representada na Figura 2.15, é primeiramente dividida em duas categorias principais, as *power grasps* e *grasps de precisão*, de seguida as *grasps* são categorizadas consoante a forma do objeto e sua funcionalidade.

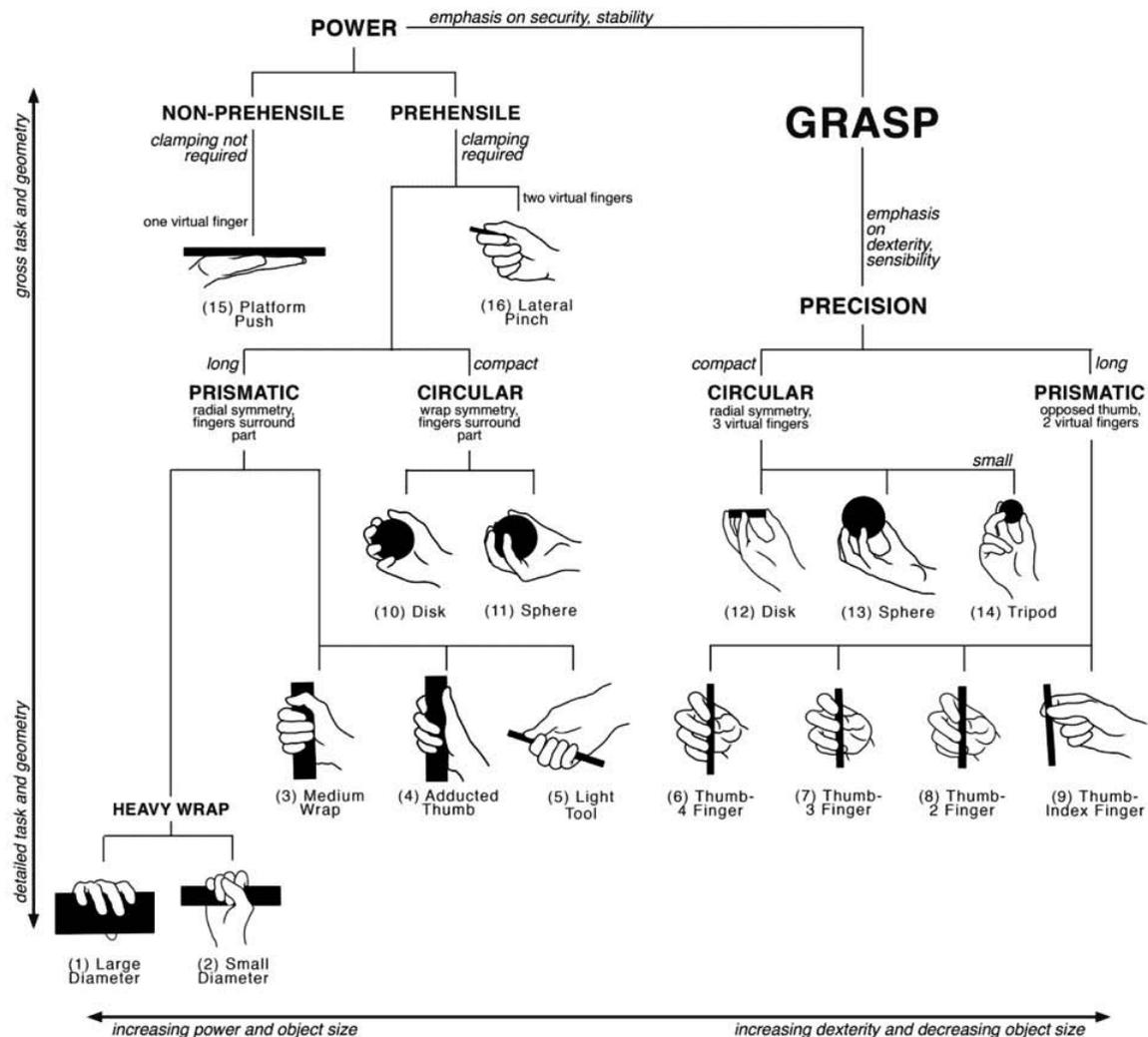


Figura 2.15: Categorização de Cutkosky [46].

2.2.2.6 Feix et al. (2011) [16] [14]

A categorização realizada por Feix é a mais recente, classificando um total de 33 *grasps*. Estas são distinguidos pelo: tipo, direção da força aplicada, posição do polegar e pelo conceito de Dedos Virtuais (DV)⁴. Tem as principais características:

⁴DV: Quando mais do que um dedo funciona em conjunto como uma unidade funcional.

- Os tipos de *grasps* são definidos como descrito na secção 2.1.4: *Power*, precisão e intermédia.
- A direção de força aplicada pode também ser definida como descrito na secção 2.1.4 em: Ponta dos dedos, palma e lateral.
- A posição do polegar, de abdução ou adução, define duas sub-categorias.
- Os DV podem ser: DV1, o primeiro DV; DV2, o segundo DV, que atua em direção contrária a DV1; DV3, em algumas *grasps* são utilizados 3 dedos virtuais.

Das categorizações mencionadas, esta última é a utilizada para testar a eficiência das sinergias adaptativas desenvolvidas para a ISR-Soft-hand. Esta pode ser consultada no Anexo B ou on-line em [14].

2.2.3 Estratégias de simplificação de controlo

2.2.3.1 Sinergias adaptativas

Nesta dissertação é proposta uma abordagem de design integrado da mecânica e do controlo com objetivo de alcançar os melhores resultados, em termos simplicidade de *design* e desempenho, ao incorporar um “controlo inteligente” no próprio desenho mecânico da *ISR-Soft-hand*.

Investigadores na área da neurociência, como Santello et al. [59], Weiss e Flanders [60] e Latash et al. [61], demonstraram que para controlar a complacência da mão humana o SNC não controla cada articulação individualmente, o que acontece é uma utilização de alguns padrões de movimento predefinidos, ou sinergias [62].

Estas sinergias podem ser incorporadas a vários níveis, a partir do agrupamento físico dos músculos e tendões até ao SNC. Em virtude desta organização, são formados vários conjuntos de ativação básica neuro-muscular, análogo ao conceito de sistema de eixos aquando a definição de um vetor [63], ou seja, um número mínimo de elementos linearmente independentes conjugados geram uma série de acontecimentos, neste caso,

o conjunto de todos os movimentos. Tal “sistema de eixos” é referido como o espaço das sinergias posturais, ou o espaço *eigengrasp* [64].

Recentemente, diferentes abordagens na área da robótica tentou tirar proveito da ideia de sinergias, com o objetivo de reproduzir um “conjunto coordenado e ordenado de movimentos” similar aos movimentos da mão humana.

Outro ponto de apoio às sinergias e incentivo à sua utilização advém do facto de o ser humano efetuar um número “limitado” de movimentos no seu dia a dia. Num estudo realizado por Santello et al. [59] uma série de 5 indivíduos foi levado a agarrar 57 diferentes objetos, daqui conclui-se que apenas duas sinergias (representadas na Figura 2.16) representavam 84% do total dos movimentos efetuados e o conjunto de três sinergias representava 94% do total de movimentos.

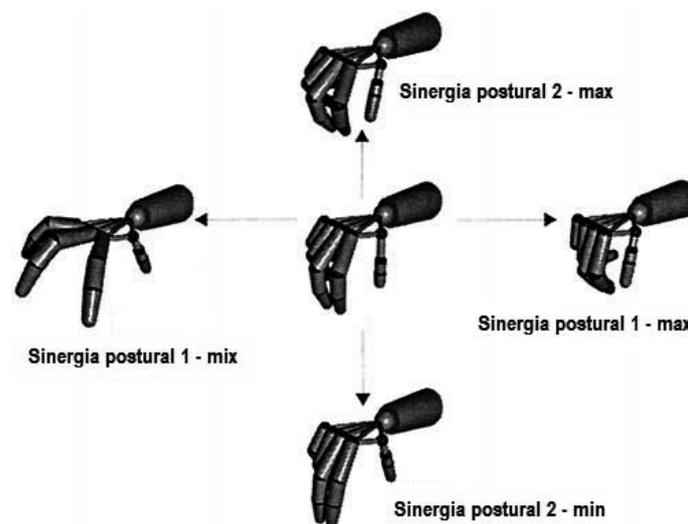


Figura 2.16: Representação das 2 sinergias adaptativas principais. As posturas da esquerda para a direita representam o máximo(max) e o mínimo(min) valor da primeira sinergia postural principal. As posturas de cima para baixo representam o máximo (max) e o mínimo (min) valor da segunda sinergia postural principal [59].

Com o intuito incorporar parte da “inteligência” típica da mão humana numa mão robótica, uma das possibilidades promissoras é a recriação destes padrões, chamados de sinergias, no sistema mecânico e eletrónico da mão robótica. Esta abordagem já foi usada por vários investigadores, quer numa abordagem puramente cinemática [64], que levava a um modelo inconsistente de distribuição de forças de contacto, quer numa abordagem do desenho mecânico. Um dos modelos bastante usados é chamado

modelo de sinergia *soft* (introduzido por Bicchi et al [37] e Gabbicini et al [65]), este modelo explica como as sinergias conseguem controlar e gerar forças internas necessárias para agarrar um objeto. Se a mão humana fosse considerado um sistema rígido, o limitado número de sinergias significaria um limitado número de pontos de contacto nos objetos que um ser humano agarraria no seu dia-a-dia. Ou seja, se a forma dos objetos se modificasse ligeiramente já não era possível realizar uma *grasp* estável. Mas um ponto fulcral em relação às sinergias é a que a mão humana não é um sistema rígido, esta contém complacência do sistema músculo-esquelético e na própria pele. Isto resultou então na definição da sinergia *soft* [27], dado que a complacência da mão permite vários pontos de contacto com a mesma configuração das articulações da mão humana e diferentes objetos.

Esta integração de complacência em mãos robóticas antropomórficas vai ajudar a aumentar a eficiência das mesmas, permitindo que agarrem um maior número de objetos com mecanismos de sub atuação. Como mencionado na secção 2.1.2.4, a complacência integrada no *design* mecânico de mãos robóticas têm a vantagem de se poder implementar sinergias adaptativas.

Esta definição de sinergia adaptativa advém de vários trabalhos de investigação (como [28], [27], [25]) com objetivo de criar mãos robóticas mais simples, com um pequeno número de atuadores sem que o número de DoF seja afetado.

A *ISR-SoftHand*, através da utilização de articulações elásticas, é capaz de agarrar objetos de forma adaptativa podendo então utilizar as chamadas sinergias adaptativas aquando o controlo dos parâmetros de entrada.

2.2.3.2 Controlo por EMG

Desde os meados dos anos 40 que os avanços a nível do controlo na robótica têm vindo a ser aplicados no controlo de mãos protéticas [66].

A EMG de superfície, onde sensores são colocados na superfície da pele do antebraço, é provavelmente o método não invasivo mais simples para adquirir os sinais de entrada para controlar mãos robóticas protéticas. Este método já é usado para o

controle de próteses desde meados dos anos 70 [67]. Mas a sua desvantagem, quando comparado com a EMG invasiva, é que os sensores superficiais apenas podem obter a ativação muscular em geral, em oposição à atividade de fibras específicas, o que acontece na utilização da EMG intra muscular [68].

Existem já várias mãos robóticas no mercado que são controladas pelo sinal da EMG não invasiva, como a DMC Plus [69], a *iLIMB ultra revolution* [70](Figura 2.20a), Michaelangelo [48] e a *RSL Steeper Bebionic* [50], [12], [66]. Todos estes exemplos são simples de manusear, não invasivos, têm um design atraente e são controlados por dois sinais de entrada de EMG [71] com um controlo proporcional.

O tipo de controlo que utiliza dois sinais de entrada, também chamado de controlo proporcional, pode controlar pelo menos uma grandeza de saída da prótese, como a força, a velocidade ou a posição, com um intervalo finito controlado pelo utilizador [71](Figura 2.17b). Neste caso o sinal de EMG é normalmente adquirido por dois diferentes grupos musculares que pode resultar da abertura/fecho da mão robótica. Este método é o de mais fácil implementação quando comparado com o controlo com um único sinal de EMG(Figura 2.17a).

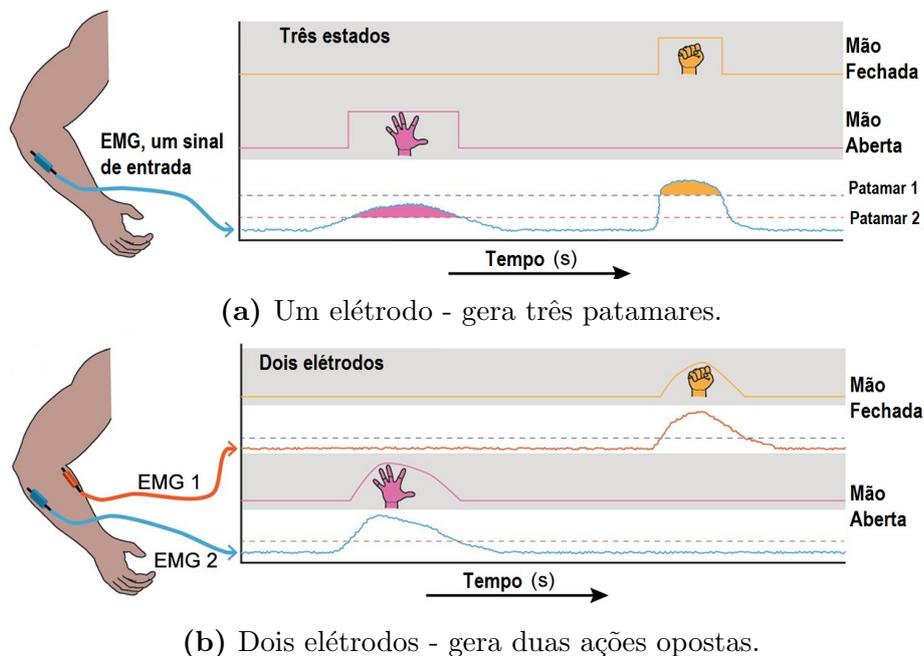


Figura 2.17: Ilustração que mostra as estratégias de controlo aquando o uso de um ou dois eletrodos de EMG. Adaptado de [12].

Neste um sinal é dividido em patamares, dependendo do nível de contração mus-

cular. Se forem definidos mais do que dois patamares, podem então ser realizados movimentos mais suaves e complexos, tal e qual faria uma mão humana [67].

No entanto, com o controlo mais simples, por dois sinais de EMG também é possível de realizar movimentos mais complexos se a este controlo for adicionado o uso de sinergias [72].

Em paralelo com a investigação e desenvolvimento dos métodos de controlo com um e dois sinais de EMG, está a também ser desenvolvido o controle com vários sinais e entrada [12]. Aqui é necessário para classificar a atividade de cada músculo para controlar diferentes funções da mão humana e, em seguida, usar essa atividade para controlar uma mão prótica de uma forma mais simples e intuitiva [73]. Existem dois tipos de processamento de sinal que normalmente são implementados neste controlo: Algoritmo de classificação e algoritmo de regressão, como se visualiza na Figura 2.18.

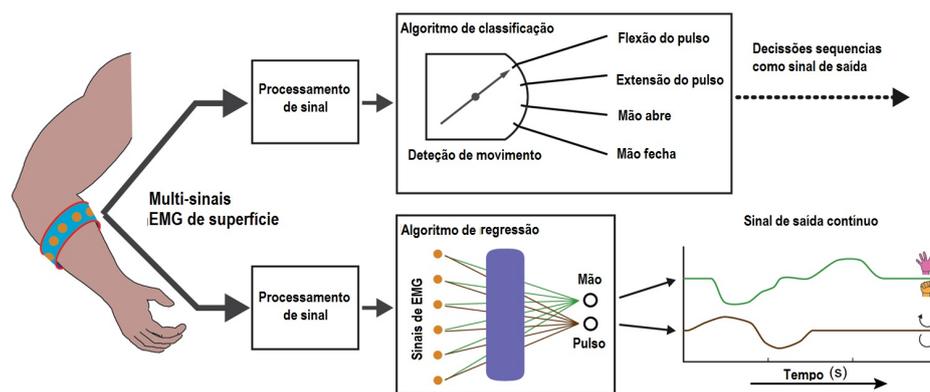


Figura 2.18: Processamento de sinal de um sistema de múltiplos sinais de entrada de EMG de superfície. Adaptado de [12].

Com o algoritmo de classificação, são identificadas as funções ativas dos músculos em contacto com os sensores de EMG, resultando numa sequência de decisões em constante atualização, no entanto este processo requer também um controlo proporcional (idêntico ao uso de dois elétrodos) [71] para que seja possível mimetizar a mão humana, tornando o controlo mais complexo.

Para o algoritmo de regressão, uma função “traduz” a informação proveniente dos sinais de EMG numa cinemática continua que pode ser usada para o controlo de mãos robóticas próticas [12].

Este tipo de controlo, com múltiplos sinais de entrada de EMG de superfície per-

mite movimentos mais complexos como rotação do pulso e movimentos independentes de diferentes atuadores. No entanto como esta técnica depende do processamento de sinal.

Com o algoritmo de classificação (Figura 2.19a) o movimento é feito sequencialmente, mas não pode ser realizado simultaneamente. Com o algoritmo de regressão o movimento é possível de realizar simultaneamente no tempo (Figura 2.19b), no entanto envolve um estudo complexo a nível da função que “traduz” a informação proveniente dos sinais de EMG numa cinemática contínua.

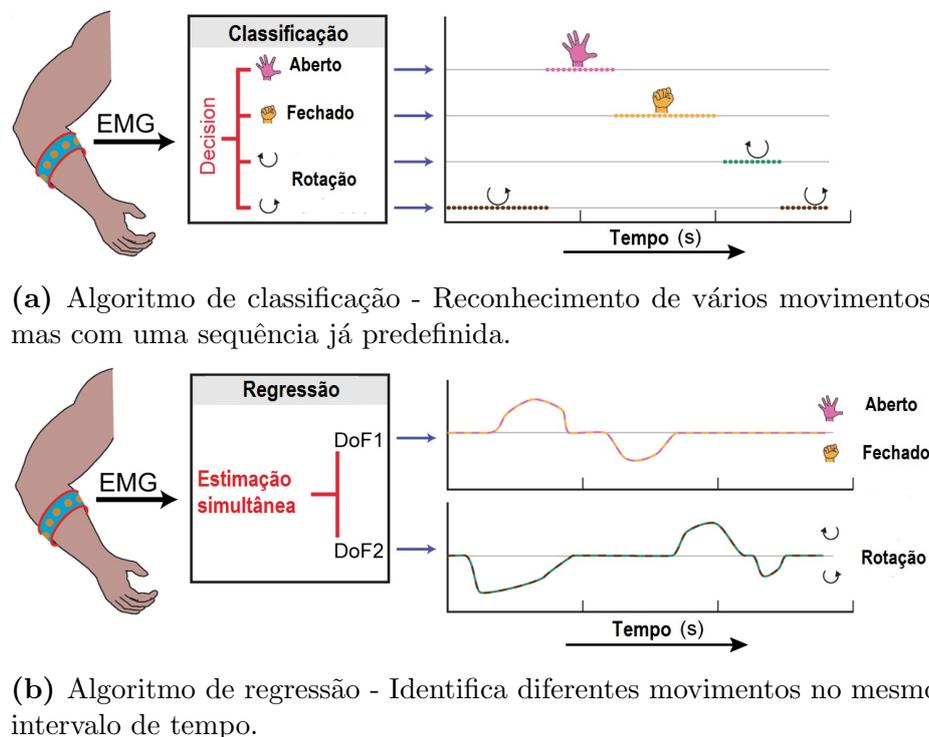


Figura 2.19: Distinção entre os dois algoritmos usados no processamento de sinal de múltiplos sinais de entrada de EMG de superfície. Adaptado de [12].

No entanto já existem outros métodos que permitem conjugar o simples controlo de EMG com outras tecnologias como aplicações para *smartphones* (como o caso da *i-LIMB ultra revolution* [70], visível na Figura 2.20b)) ou o uso da tecnologia de IMU⁵ controlada pelo pé (usado no *DEKA arm* [74], como visível na Figura 2.20c) [75]. Nesta dissertação pretende-se então desenvolver sinergias adaptativas, com o menor número de parâmetros de entrada, para uma futura implementação um controlo sim-

⁵*Inertial Measurement Unit* - Capaz de recolher medidas de velocidade, orientação e forças gravitacionais

ples (de um dois sinais de EMG) com sensores de superfície, para a *ISR-Softhand*, uma mão antropomórfica sub atuada que contem complacência acoplada nas articulações. Todo o desenvolvimento deste trabalho será descrita em pormenor no próximo capítulo (Capitulo 3).



(a) Uso de controlo por dois sinais de EMG pela i-LIMB ultra revolution [70].
(b) Uso de controlo por aplicação por smartphone pela i-LIMB ultra revolution [70].



(c) Uso de controlo pela utilização da IMU(*Inertial Measurement Unit*) no pé pelo DEKA arm [74].

Figura 2.20: Vários métodos de controlo de mãos prostéticas robóticas [75].

Capítulo 3

ISR-Softand

Neste capítulo será dada uma breve descrição da *ISR-Softand*. A *ISR-Softand* consiste num sistema eletromecânico adaptativo de baixo custo que mimetiza uma mão humana. Este capítulo aborda os seguintes temas relativos à mão protética:

3.1 Características e *Design*

3.2 Atuadores

3.2.1 Atuadores AX-12A

3.1 Características e *Design*

A Figura 3.1 mostra um esquema da *ISR-softand*. Na *ISR-softand* cada dedo é composto por articulações MCF e IFP e não incluem a articulação IFD. O polegar é particularmente diferente, sendo composto pelas articulações CMC e MCF, dado que na própria anatomia esta não inclui a articulação IFP. Com isto, a mão em questão fica com um total de 10 articulações.

À exceção da articulação CMC do polegar, todas as restantes são compostas por materiais elásticos. Cada dedo contem um tendão que dobra a articulação MCF e IFP em simultâneo, resultando na flexão do dedo.

A articulação IFP foi projetada de maneira a ser mais rígida em relação à MCF, deste modo quando a articulação MCF está bloqueada (ou seja, quando é efetuado o primeiro contacto com o objeto) a força do tendão é totalmente aplicada na articulação

IFP, levando à flexão do dedo de maneira a que circunda o objeto. Esta é uma das importantes características da *ISR-softand*, uma vez que não permite que toda mão se feche por completo antes de agarrar o objeto.

Em relação à articulação CMC do polegar, esta tem a característica de poder sofrer rotação dos 0° aos 90° manualmente, permitindo que o dedo polegar possa ter duas posições fixas, ou em paralelo (posição de adução) ou em perpendicular (posição de abdução) em relação à palma da mão, como é visível na Figura 3.1.

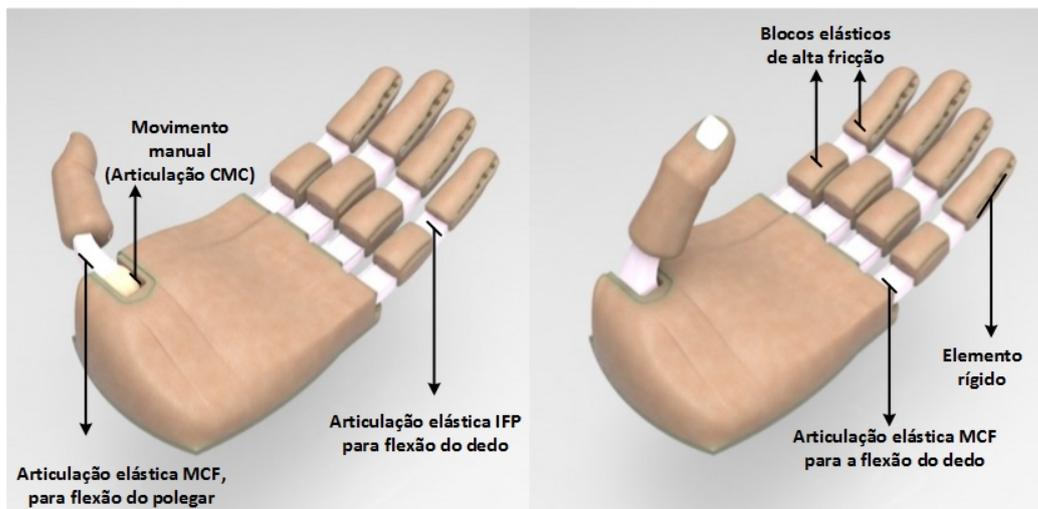


Figura 3.1: Design conceitual da *ISR-Softand*. Cada dedo é composto por uma articulação MCF e uma articulação IFP, não incluindo a articulação IFD. O polegar é composto por uma articulação CMC e uma MCF. Com exceção da articulação não atuada do polegar para efetuar o movimento de abdução/adução(CMC) todas as restantes articulações são fabricadas com materiais elásticos.

Isto torna a mão mais simples, evitando o uso de um outro atuador para o movimento de Abdução/Adução do polegar. Este conceito já é realizado por outras mãos protéticas, como é o caso da *iLIMB* [49] e da *Bebionic* [50].

Visto a importância das *graps* de precisão, esta mão foi projetada de modo a ter um atuador que controle o polegar, outro que controle o dedo indicador e por fim um restante que controle os restantes três dedos (médio, anelar e mindinho), tudo isto pode ser visualizado na Figura 3.2. Ao ter na sua constituição menos atuadores que graus de liberdade (correspondente ao número de articulações) esta mão é chamada de sub atuada.

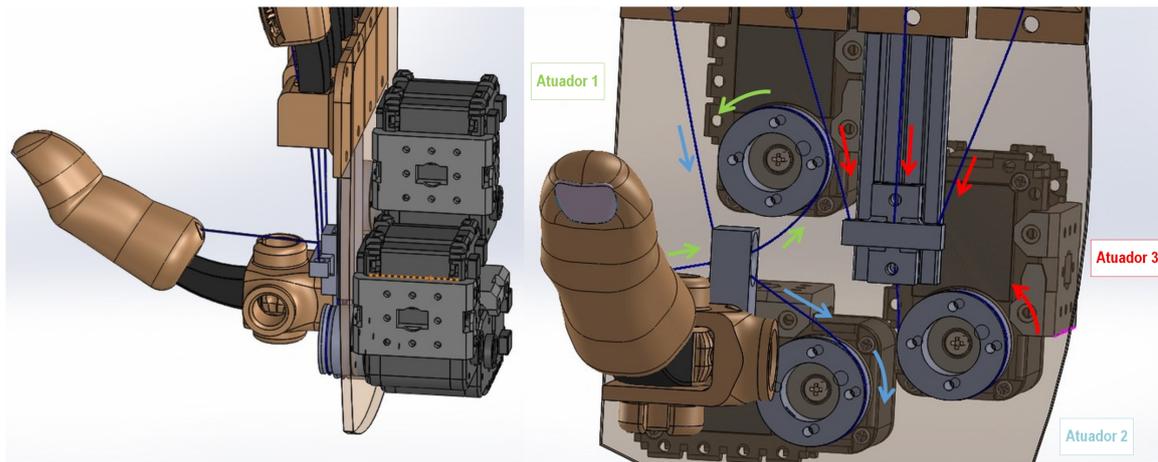


Figura 3.2: Esquema onde se visualiza a relação de cada atuador com cada dedo e à palma na *ISR-Softhand*. À esquerda visualiza-se a disposição dos atuadores em relação a cada dedo, à direita visualiza-se que existem dois tendões que cada um é puxado por um atuador individualmente (a verde o que puxa o polegar e a azul o que puxa o indicador) e um outro atuador que puxa os restantes 3 dedos (a vermelho)

A Tabela 3.1 mostra as principais características do primeiro protótipo da *ISR-Softhand*.

Tabela 3.1: Características da *ISR-Softhand*

DoFs	10*
DoAs	3
Dimensões (mm)	200*91*27
Massa (g)	530
Max. de massa que levanta (g)	1800

* Neste caso DoFs refere-se ao número de articulações, como explicado na secção 2.1.2.3.

3.2 Atuadores

A *ISR-Softhand* tem na sua constituição, como já referido, 3 atuadores. Estes atuadores são do modelo *Dynamixel AX-12A* e foram adquiridos através da ROBOTIS. As suas principais características são o seu baixo peso (54,6 g) e o seu baixo preço ($\approx 32\text{€}$, se comprado individualmente). No início deste trabalho foi importante conhecer as suas características e o seu funcionamento, para que de seguida fosse então possível implementar as sinergias adaptativas.

3.2.1 Atuador AX-12A

Este tipo de atuadores, disponíveis em [76], têm um ID único e não necessitam de que todos os atuadores estejam ligados ao controlador, pois possuem um protocolo TTL que os permite ligar em cadeia (com se vê na Figura 3.3), facilitando a sua montagem.

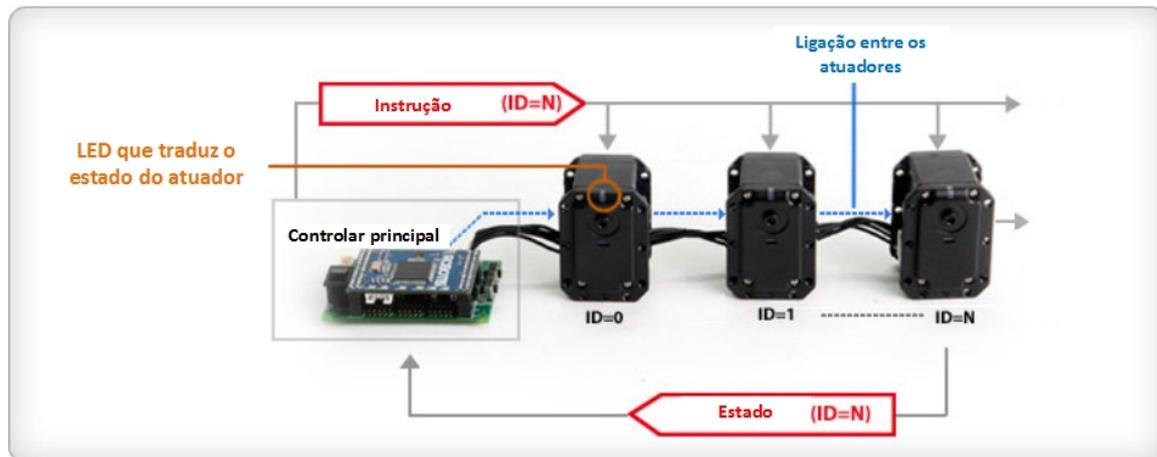


Figura 3.3: Esquema de montagem e comunicação entre os atuadores DYNAMIXEL AX-12A da ROBOTIS. Adaptado de [77].

Cada atuador é composto por um motor de corrente contínua, uma caixa de velocidades, um potenciômetro instalado no eixo de saída para um controlo de posição (entre os 0° e os 300°) e um mecanismo que recebe energia diretamente do motor e transmite movimento para, por exemplo, a caixa de velocidades e o potenciômetro.

As principais características deste atuador estão descritas na Tabela 3.2.

Tabela 3.2: Características do atuador DYNAMIXEL AX-12A da ROBOTIS, adaptado de [76].

Modelo	AX-12A
Peso (g)	54,6
Dimensões (mm)	35x50x40
Porta de comunicação	TTL
Tensão de funcionamento(V)	[9,0 12,0] V
Sensor de Posição (Resolução)	300° 1024
Binário (N.m)	11,5 @ 12V; 1,5A
Velocidade (rpm)	59* @ 12V

*0.169s/60°.

Em relação à configuração e programação do atuador, este pode ser controlado de duas maneiras, ou por controladores específicos do fabricante ou com o uso de softwares comuns de programação, como se visualiza na Figura 3.4. Neste caso o controlador inicialmente utilizado foi o CM-510, associado a um software de controlo bastante simples (o Roboplus, Figura 3.6), passando de seguida ao USB2Dynamixel para o uso do software Matlab[®] para a configuração dos atuadores.



Figura 3.4: Opções possíveis para a configuração dos atuadores DYNAMIXEL da ROBOTIS. Adaptado de [77].

Na Figura 3.5 está representado o esquema de montagem utilizado neste projeto, onde é visível o controlador CM-510 e o USB2Dynamixel.



Figura 3.5: Esquema de montagem real.

Em relação ao software de controlo associado ao CM-510, este está representado na Figura 3.6 e é constituído por 3 interfaces principais, simples e de fácil acesso.

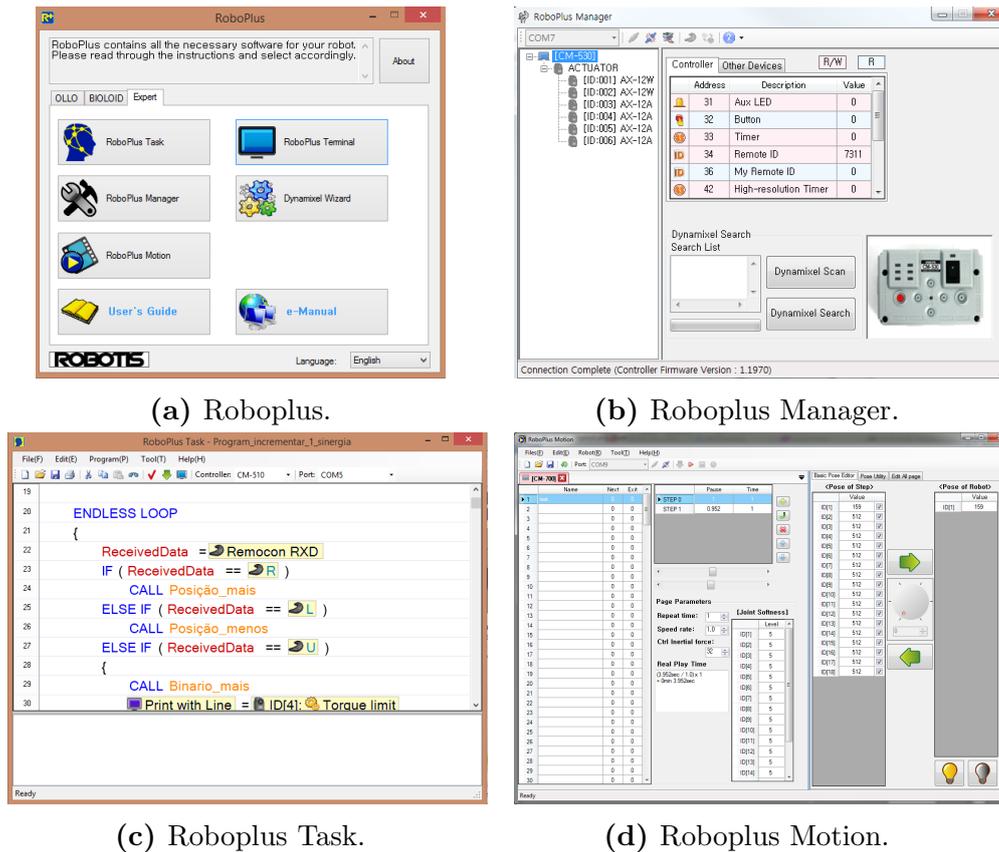


Figura 3.6: Interfaces existentes no software RoboPlus da ROBOTIS[®]. Adaptado de [77].

A partir do menu principal (Figura 3.6a) é possível aceder às três interfaces principais, são estas o RoboPlus Manager (Figura 3.6b), o RoboPlus Task (Figura 3.6c) e o RoboPlus Motion (Figura 3.6d). O primeiro permite configurar e inspecionar o estado do atuador, podendo o utilizador mudar parâmetros como o seu ID, a posição, a velocidade e o binário do atuador. É também neste onde são definidos os valores de temperatura em que o atuador se desliga automaticamente, para não sofrer danos. No segundo é possível programar o atuador, com uma linguagem simples e de fácil interpretação. O terceiro permite a criação de um programa simples onde se definem movimentos e velocidades do atuador e permite que este se movam continuamente segundo os comandos dados.

Neste projeto não foi utilizado o RoboPlus Motion, apenas o RoboPlus Task para testar e implementar posições e de seguida foi criado um código em Matlab com cada uma das sinergias adaptativas (Anexo C).

Capítulo 4

Design, Teste e Implementação das Sinergias Adaptativas

Neste capítulo será descrito o trabalho desenvolvida no *design* teste e implementação das sinergias adaptativas na *ISR-softand*. Tratou-se de um trabalho evolutivo entre experimentação e *design*, para que o controlo da prótese robótica fosse o mais simples possível e sucedessem o maior número de *grasps* com o menor número de sinergias.

4.1 *Grasps*

4.1.1 Controlo dos parâmetros de entrada e primeiros resultados

4.2 Categorização das *grasps* e sinergias

4.3 Sinergias

4.3.1 Sinergia I.A & I.B

4.3.2 Sinergia II

4.3.3 Sinergia III

4.3.4 Sinergia IV

4.3.5 Sinergia V

4.1 *Grasps*

Com a finalidade de implementar sinergias adaptativas com a *ISR-Softand*, primeiramente foram testadas as 33 *grasps* da categorização de Feix et al. [42]. Na fase inicial deste projeto tentou-se realizar o máximo de *grasps* possíveis pelo controlo ma-

nual e independente de cada um dos atuadores. Este controlo consistiu em conhecer a resposta dos atuadores mencionados anteriormente em relação a três parâmetros de entrada: posição (θ), binário (τ) e a diferença temporal (Δt) entre atuadores, com é visível no fluxograma da Figura 4.1. De relembrar que a complacência da *ISR-Softhand*, esta incorporada no próprio desenho mecânico não sendo necessário ter em conta este parâmetro aquando o controlo da posição de cada um dos atuadores, através do software de controlo.

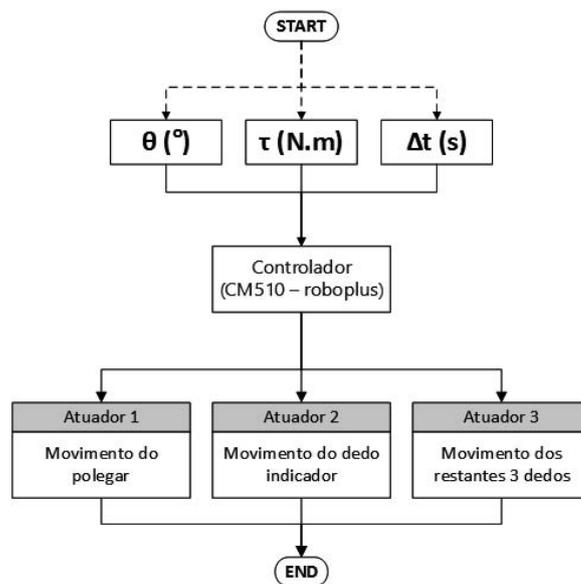


Figura 4.1: Fluxograma de implementação inicial dos parâmetros de entrada na *ISR-Softhand*.

Em relação ao binário (τ), depois de vários testes concluiu-se que este depende da posição (como de visualiza na Figura 4.2, onde está representado o valor da posição, θ , em graus, e do binário, τ , valor retirado do roboPlus Manager), ou seja, apenas usando o comando da posição varia-se automaticamente o binário exercido pelo atuador.

Em relação ao controlo da diferença temporal (Δt), este está relacionado diretamente com o controlo da velocidade, isto é, colocando os atuadores a diferentes velocidades consegue-se incorporar um intervalo de tempo entre o movimento dos atuadores. A variação da velocidade de cada atuador pode tornar o movimento do polegar e dos dedos mais suave, na medida em que a mão evita um contacto brusco com o objeto, o que poderia tornar a *grasp* menos estável, e por consequência, esta poderia não ser possível de realizar.

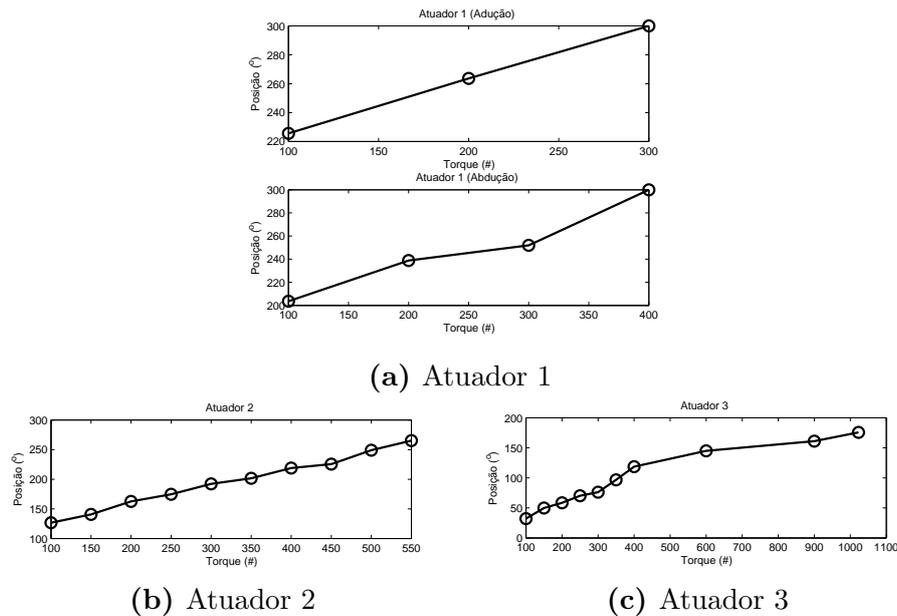


Figura 4.2: Binário (τ), em unidades do software RoboPlus Manager, vs. Posição (θ), em graus ($^{\circ}$).

Depois de conhecidas as principais características dos parâmetros de entrada a controlar e da tentativa de realização das 33 *grasps*, foi possível classifica-las em três grupos: possíveis, aproximadas e impossíveis. Compreenda-se como *grasps* aproximadas aquelas que não conseguem imitar perfeitamente o esquema disponível na categorização de Thomas et al. [42], mas que são possíveis de realizar pela *ISR-Softhand* e estáveis. Por outras palavras, é possível agarrar o objeto mas a imitação antropomórfica da *grasps* não é conseguida.

4.1.1 Controlo de parâmetros de entrada e primeiros resultados

Inicialmente foi apenas controlado o parâmetro da posição (θ) para controlar cada uma das *grasps*. Os restantes parâmetros foram controlados para o *design* das sinergias adaptativas. Para testar cada uma das *grasps* e desenvolver as sinergias adaptativas foram usados 21 objetos, mostrados na Figura 4.1. A Tabela 4.1 indica as dimensões dos objetos representados.

A Tabela 4.3 mostra a melhor imitação de cada uma das *grasps* pela *ISR-Softhand*. Estas *grasps* foram conseguidas pela implementação do código em Matlab[®], dis-



Figura 4.3: Objetos utilizados para testar cada uma das *grasps* e desenvolver as sinergias adaptativas para a *ISR-SoftHand*.

	Altura ($\pm 0,01$ mm)	Largura ($\pm 0,01$ mm)	Espessura ($\pm 0,01$ mm)	Diâmetro ($\pm 0,01$ mm)
Bola de ténis	-	-	-	64,50 mm
Bola esponja (meia)	-	-	-	87,00 mm
Bola média	-	-	-	48,80 mm
Bola ping-pong	-	-	-	40,00 mm
Caderno	210,00 mm	148,00 mm	16,00 mm	-
Canetas (azul verde e laranja)	140,00 mm	-	-	9,55 mm
Cartão	85,60 mm	54,00	9,55 mm	-
CD	-	-	1,5 mm	112,00 mm
Chave de fendas	167,00 mm	-	-	0,27-0,94 mm
Copo	14,00 mm	-	-	61,50 mm
Desodorizante	18,00 mm	-	-	44,91 mm
Lápis	140,00 mm	-	-	7,50 mm
Parafuso	21,16 mm	-	-	1,90 mm
Prato plástico	-	-	0,55 mm	205,00 mm
Rolo de linha	23,75 mm	-	-	14,60 mm
Tesoura	147,00 mm	60,00 mm	-	-
Tubo fino	125 mm	-	-	16,20 mm
Tubo grosso	300,00 mm	-	-	75,10 mm
Tubo médio	400,00 mm	-	-	35,00 mm

Tabela 4.1: Dimensões dos 21 objetos utilizados para testar cada uma das *grasps* e desenvolver as sinergias adaptativas para a *ISR-SoftHand*.

ponível no Anexo C.

Como se visualiza na Tabela 4.3 da lista de 33 *grasps*, apenas uma *grasp* é completamente impossível de realizar pela *ISR-SoftHand*, pois necessitava da atuação do movimento de abdução/adução entre o dedo indicador e o dedo médio.

Das restantes 32 *grasps*, 20 são possíveis de realizar pela *ISR-SoftHand*, 11 são aproximadas e uma é impossível de realizar com este protótipo, como é visível em resumo na Tabela 4.2.

De seguida é dada uma breve explicação da razão da aproximação/inexequibilidade das *grasps*:

Tabela 4.2: Resumo da quantidade de *grasps* possíveis, aproximadas e impossíveis de realizar pela *ISR-Softand*.

Imitação	Número de <i>grasps</i>	N.º da <i>grasp</i> [42]
Possível	20	1, 2, 4, 6, 9, 11, 13, 14, 15, 16, 19, 22, 24, 25, 26, 27, 28, 30, 31, 33.
Aproximada	11	3, 7, 8, 10, 12, 17, 18, 20, 21, 29, 32.
Impossível	2	5, 23.

- *Grasp* nº5 Inexequível pois, de momento, não é possível dobrar os dedos 3, 4 e 5 com tanta flexão devido a um *design* mecânico não apropriado e a pouco torque do motor para aguentar puxar os três dedos ao máximo sem aquecer demasiado. No entanto pode ser melhorado através de um diferente desenho mecânico dos respetivos dedos.
- *Grasp* nº 3, 12, 17, 18, 20, 21, 29 e 32 Aproximadas pois o dedo polegar não tem capacidade de uma posição intermédia, apenas pode adquirir a posição de abdução/adução, foi então escolhida para cada *grasp* aquela que lhe daria mais estabilidade.
- *Grasps* nº 7 e 8 Aproximadas pois o objeto não é alinhado/colado na posição idêntica aquela do esquema.
- *Grasp* nº 10 Aproximada pois a *ISR-Softand* não tem flexão a nível das articulações IF.

De referir que é sempre necessário a ajuda de outra mão para o posicionamento correto do objeto na mão robótica antes de efetuar a *grasp*. Tal também acontece com outras mãos próstéticas comerciais, como é o caso da *iLIMB* [49] e da *Vincent hand* [51].

Tabela 4.3: Representação das 33 grasps da lista de Feix et al. [42] e a melhor imitação conseguida pela ISR-Softhand. A cada cinco colunas está representado: Número da *grasp*, esquema da respetiva *grasp* [14], imitação pela *ISR-Softhand*, Número da sinergia utilizada para a realização da *grasp*, nível de imitação conseguido (+/- significa que o objeto foi agarrado mas não com a perfeita imitação; + significa que a imitação foi possível). A circunstância vermelha representa a configuração escolhida quando é possível realizar a *grasp* com mais do que uma sinergia.

Nº	Grasp	ISR Softhand	Sinergia Imitação Nº.	Grasp	ISR Softhand	Sinergia Imitação Nº.	Grasp	ISR Softhand	Sinergia Imitação
1			I.A +			I.A +/-			Impossible -- --
2			I.A +			I.A +			I.A + II +
3			I.A +/-			I.A +			I.A +
4			III +			III +			I.A +
5			--			I.B +			I.A +
6			I.A +			IV +/-			I.B +

<p>IV +/-</p>  <p>29</p> 	<p>III +</p>  <p>30</p> 	<p>II +</p>  <p>31</p> 	<p>IV +/-</p>  <p>32</p> 	<p>I.A. +</p>  <p>I.A. +</p>  <p>33</p> 
<p>I.A. +/-</p>  <p>18</p> 	<p>I.A. +/-</p>  <p>19</p> 	<p>I.A. +/-</p>  <p>20</p> 	<p>I.A. +/-</p>  <p>21</p> 	<p>I.A. +/-</p>  <p>22</p> 
<p>I.A. +/-</p>  <p>7</p> 	<p>I.A. +/-</p>  <p>8</p> 	<p>I.A. +</p>  <p>9</p> 	<p>I.A. +/-</p>  <p>10</p> 	<p>I.A. +</p>  <p>11</p> 

4.2 Categorização das *grasps* e sinergias

Com o intuito de criar as sinergias adaptativas baseadas na posição das pontas dos dedos, das articulações de cada atuador e na posição registada no controlo dos atuadores, as *grasps* foram categorizadas em grupos coerentes onde cada grupo pode realizado por uma sinergia. O objetivo foi sempre minimizar o número total de grupos e maximizar o número de *grasps* por grupo. Seguindo isto, foi criado um grupo principal com o maior número de *grasps* possíveis de realizar pela *ISR-Soft-hand*.

Primeiramente serão categorizadas as *grasps* e na próxima secção (secção 4.3) é explicado o *design* das sinergias para cada um dos grupos. É importante referir que este processo foi interativo, a categorização dos grupos e o *design* das sinergias foram mudando ao longo do projeto por várias vezes de maneira a se conseguir um número máximo de *grasps* no grupo principal e o máximo de estabilidade em cada *grasp*. Os resultados presentes na Tabela 4.3 mostram apenas apenas as *grasps* finais realizadas com as sinergias adaptativas.

Se uma *grasp* era possível de realizar pela Sinergia I e qualquer outra sinergia, esta era colocada no primeiro grupo, desde que a realização da *grasp* por outra sinergia não oferecesse uma vantagem óbvia. Isto pode ser visto, por exemplo, na *grasp* n.º 9, onde a diferença entre as duas categorias é apenas a posição dos três dedos passivos (os três dedos que não tocam no objeto). Visto isto, esta *grasp* foi colocada no Grupo I, o que é mostrado pela circunferência vermelha na Tabela 4.3. Já a *grasp* n.º 18 foi colocada no Grupo V dado a melhor estabilidade conferida pela Sinergia V, como será discutido mais à frente nesta secção.

As *grasps* foram então categorizadas nos seguintes 5 grupos:

1. Grupo I. O grupo principal.

Este é o principal e maior grupo definido. A principal característica das *grasps* deste grupo está no facto da flexão do polegar e dos restantes dedos ser feita simultaneamente. Em alguns casos, por exemplo a *grasp* nr. 9, a simultaneidade de movimento das atuadores não é necessária para o desempenho da *grasp*, mas dado que

o movimento de todos os dedos não afeta a performance da *grasp* esta é inserida no grupo principal para seguir o objetivo de maximizar o número de *grasps* neste grupo.

Este grupo contém 7 das 15 *power grasps* presentes na categorização de Feix et al. [42], 11 das 12 *grasps* de procição e 3 das 6 *grasps* intermédias.

É ainda dividido em duas subcategorias, para os casos em que o polegar está na posição de abdução (Figura 4.4) e adução (Figura 4.5). Em ambos, o padrão de movimento dos atadores é semelhante, diferenciando apenas a posição do polegar em relação à articulação CMC. A sub categorização torna-se possível porque o movimento da articulação CMC é manual na ISR-Soft hand (limitado a duas posições, como já referenciado ao longo dos capítulos 1, 2 e 3).

I.A Polegar em abdução: Nesta subcategoria do grupo principal, o eixo de flexão polegar deve ser paralelo ao eixo de flexão do dedo indicador. Como pode ser visto na Tabela 4.3, existem dezanove *grasps* nesta subcategoria, as *grasps* n.º 1, 2, 3, 6, 7, 8, 9, 10, 11, 12, 13, 14, 20, 21, 24, 25, 26, 27 e 33.

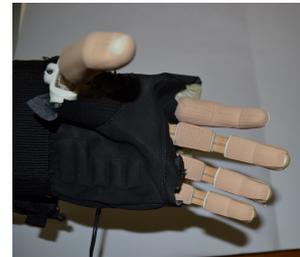


Figura 4.4: *ISR-Soft hand* com o polegar em Abdução.

I.B Polegar em adução: Nesta subcategoria do grupo principal, o eixo de flexão polegar deve ser perpendicular ao eixo de flexão do dedo indicador. Como pode ser visto na Tabela 4.3, existem duas *grasps* nesta subcategoria, as *grasps* n.º 16 e 28.



Figura 4.5: *ISR-Soft hand* com o polegar em Adução.

Em resumo, podemos colocar no grupo principal 21 das 31 *grasps* possíveis de realizar com este protótipo da *ISR-Soft hand*.

2. Grupo II.

Este grupo contém apenas uma *grasp*, a *grasp* n.º 31. As *grasps* n.º 9, 24 e 33 poderiam ser colocadas neste grupo, mas dado que neste caso os restantes 3 dedos não tocam no objeto, e não tem efeito na maneira em que o objeto é agarrado, estas foram colocadas no primeiro grupo.

A necessidade da existência deste grupo e a da sua respetiva sinergia serão discutidas mais à frente. No entanto, dado que o seguimento da categorização de Feix et al. [42] este grupo foi criado e considerado.

3. Grupo III.

Este grupo contém 3 *grasp*, as *grasps* n.º 4, 15 e 30. Neste grupo o polegar não sofre qualquer flexão, podendo tocar ou não o objeto (na *grasp* n.º 15 não existe contacto entre o polegar e o objeto).

4. Grupo IV.

Este grupo contém as *grasps* onde o dedo indicador está esticado para dar estabilidade ao agarrar longos objetos. Como se visualiza na Tabela 4.3, existem 4 *grasps* neste grupo, as *grasps* n.º 17, 19, 29 e 32. Na prática significa que o atuador acoplado ao dedo indicador move-se o mínimo possível para dar a força de contacto necessária para a estabilidade da *grasp*.

5. Grupo V.

Este grupo contém as *grasps* de objetos com forma paralelepípeda. Existem duas *grasps* neste grupo, as *grasps* n.º 18 e 22. Como se pode ver na Tabela 4.3 estes casos também poderiam ser colocados no Grupo I, no entanto quando se agarrava os objetos com o movimento acoplado dos atuadores as *grasps* não eram estáveis. Ou seja, aquando a realização destas *grasps* com a Sinergia I, tornava-se difícil encontrar uma posição onde os objetos estivessem estáveis. A diferença em relação ao primeiro grupo está na flexão do polegar, aqui o polegar flete mais que os restantes dedos,

dando uma maior estabilidade. A Figura 4.6 mostra a diferente área de contacto dos 4 dedos em ambos os casos. Quando se tentava agarrar objetos mais finos com a primeira sinergia, como o caso da *grasp* n.º 18, os dedos necessitavam de estar praticamente no seu máximo de flexão. Isto fazia com que a área de contato entre os dedos e o objetos fosse mínima, dando muita pouca estabilidade. Quando se colocou o polegar a dobrar mais que os restantes dedos, verificou-se que a área de contacto aumentava, dando então a estabilidade pretendida.



(a) Sinergia I. (b) Sinergia V.

Figura 4.6: Fotografias das *grasps* n.º 18 e 22, realizadas com a Sinergia I e V.

A Tabela 4.4 resume o total de *grasps* realizáveis (em cada um dos grupos) e não realizáveis pela *ISR-SoftHand*.

Tabela 4.4: Número e percentagem de *grasps* realizáveis (em cada um dos grupos) e não realizáveis pela *ISR-SoftHand*.

	I	II	III	IV	V	Impossíveis	Total
Nr.	21	1	3	4	2	2	33
%	63,3	3,0	9,1	12,1	6,1	6,1	100

4.3 Sinergias

Com base nos grupos anteriores foram criadas 5 sinergias. Isto é, foram ajustadas as posições e velocidades de cada atuador com a finalidade de realizar todas as *grasps* em cada grupo. Com a exceção da Sinergia I, todas as restantes sinergias são compos-

4.3. Sinergias

tas por apenas um movimento simultâneo que envolve a movimentação dos atuadores de uma posição inicial para uma posição final, como se pode ver na Tabela 4.5. Enquanto uma mão rígida com articulações mecânicas necessitaria de um controlo dos parâmetros de entrada mais preciso, para que os pontos de contacto da mão com o objeto fossem estabelecidos, a *ISR-Softand* permite um controlo dos parâmetros de entrada dos atuadores mais simples. Isto deve-se à complacência integrada no desenho mecânico do protótipo da mão próstética.

Tabela 4.5: Sinergias passo a passo

	Passo					
Sinergia	1	2	3	4	5	6
I.A						
I.B						
II						
III						
IV						
V						

A Sinergia I foi dividida em vários passos desde a posição inicial até à posição final dos atuadores para conseguir realizar o máximo de *grasps* possíveis. Mais uma vez, estes ajustes foram simples de realizar, dado que a *ISR-Softand* não precisa

de um controle preciso do movimento das articulações. Isto clarifica o conceito de sinergia adaptativa. Num sistema mecânico rígido, cada *grasps* necessitaria de um posicionamento preciso de cada atuador, enquanto que com sinergias adaptativas os pontos de contacto podem ser aproximados, dado que podem ser considerados pontos de contacto dentro do objeto e não é necessário um estudo aprofundado de cada objeto para a realização da respetiva *grasp*.

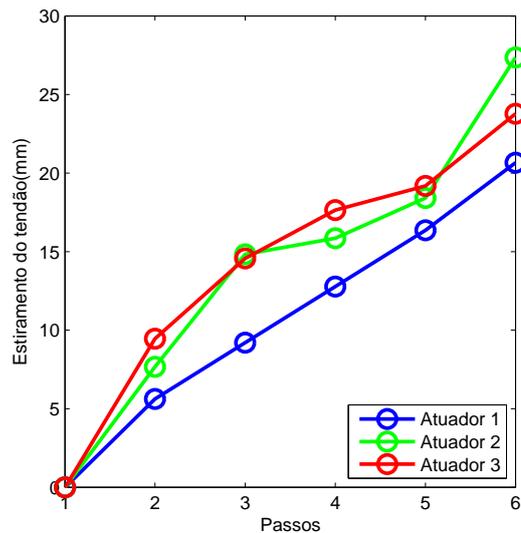
4.3.1 Sinergia I.A & I.B

Para ser possível realizar todas as *grasps* categorizadas no Grupo I, foram definidos 4 pontos intermédios entre a mão completamente aberta e completamente fechada. Os pontos intermédios e o perfil de movimento dos atuadores de cada um dos passos foram cuidadosamente definidos para cobrir todas as *grasps* na primeira categoria. Tem ainda a vantagem do utilizador poder parar o mão em qualquer um dos passos desta sinergia.

A Tabela 4.6 mostra quanto o tendão é puxado por cada atuador. Como se pode ver na Figura 4.7, nos primeiros três passos, os dedos são puxados com a mesma taxa pelos 2 atuadores (Atuador 2 e 3). Estes passos apenas de diferenciam na melhor adaptabilidade da mão a objetos com diferentes tamanhos. No passo 4, o atuador 2 puxa o dedo indicador menos que os restantes, dado que existem cinco *grasps* (*Grasps* n.º 8, 9, 20, 21 e 25) que necessitam, de contacto lateral com o dedo indicador. No passo 5, o dedo indicador volta a estar alinhado com os restantes dedos. No último passo, todos os atuadores puxam o tendão até ao ponto máximo possível. Neste passo o tendão do polegar não apresenta tanto estiramento como os restantes, dado que este tem um menor comprimento quando comparado com os tendões dos outros dois atuadores.

Tabela 4.6: Valores do estiramento do tendão, em milímetros, por cada atuador em cada um dos passos da Sinergia I.

	1(mm)	2(mm)	3(mm)	4(mm)	5(mm)	6(mm)
Actuator 1	0	5,62	9,20	12,78	16,36	20,66
Actuator 2	0	7,67	14,83	15,85	18,41	27,36
Actuator 3	0	9,46	14,57	17,64	19,18	23,78

**Figura 4.7:** Estiramento dos tendões, em mm, em cada passo da Sinergia I

Por outro lado os tendões dos 3 dedos acoplados no atuador 3 também não conseguem ser puxados até ao seu máximo, isto deve-se ao desenho mecânico deste primeiro protótipo, como se visualiza na Figura 4.8 e ao facto de o motor utilizado não conseguir puxar os 3 dedos até ao seu máximo sem que falhe, pela demasiado torque aplicado (isto é visível na *grasp* n.º 2, na Tabela 4.3, no passo 6 da Sinergia I e na passo final da Sinergia 3, na Tabela 4.5).

Em relação à velocidade de rotação dos atuadores (representado na Figura 4.9), nos primeiros dois passos os atuadores estão na máxima velocidade (59 rpm), nos passos 3 e 4, a velocidade baixa para os 10% da velocidade máxima, para que as *grasps* sejam feitas com suavidade, principalmente as *grasps* de precisão que estão inseridas no grupo I. De seguida o polegar baixa a sua velocidade para os 5% e os restantes atuadores continuam 10%, para que o polegar consiga dobrar por cima do dedo indicador e se consiga realizar a *grasp* n.º 2.

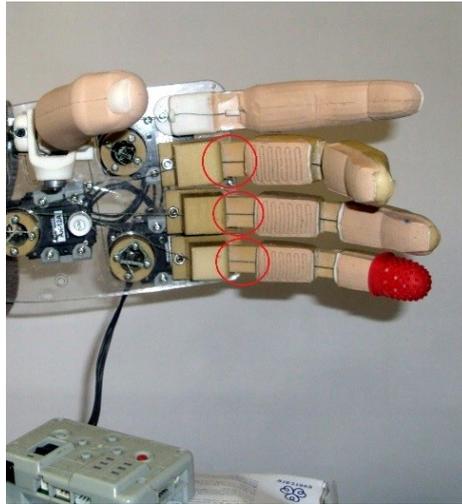


Figura 4.8: Figura onde se visualiza o problema do desenho mecânico (a vermelho) que impede o máximo fecho dos 3 dedos atuados pelo atuador 3.

Para clarificar as vantagens dos passo intermédios, deve ser mencionado que quando se tentou realizar esta sinergia com apenas dois passos (com a posição inicial e a posição de fecho em que o polegar está em contacto com o dedo indicador e os restantes dedos estão alinhados com o último) apenas 13 *grasps* eram possíveis de realizar. As *grasps* n.º 2, 8, 12, 16, 20, 21, 25 e 28 não conseguiram ser realizadas. Mesmo as 13 *grasps* possíveis não eram feitas com grande qualidade, dado que a divisão da sinergia em 4 passos intermédios permite que os movimentos sejam mais suaves, o que permite, por exemplo, agarrar um ovo (Figura 4.10). Uma razão para este melhor desempenho é que, na ausência de fases intermédias, uma força de contacto excessiva é aplicada a objetos de maiores dimensões, uma vez que o ponto de contacto é definida muito longe das fronteiras efetivas do objeto. Além disso, é possível definir diferentes perfis de movimento para os atuadores, permitindo diferentes velocidades de fecho cada atuador em cada passo.

Nos primeiros três passos (passo 1, 2 e 3), o perfil de movimento foi projetado para que o dedo indicador e os restantes 3 dedos estivessem alinhados, para um aplicação de forças de contacto idênticas em objetos longos. No passo 4, como já referido, o dedo indicador move-se mais que os restantes dedos, para que o dedo médio possa aplicar força de contacto lateral nos objetos (*grasp* n.º 20, 21 e 25). Neste passo também se dá o contacto entre o dedo indicador e o polegar, permitindo a realização de *grasps* de

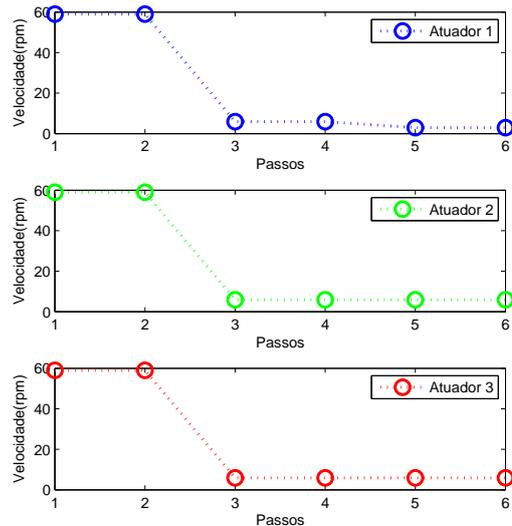


Figura 4.9: Velocidade rotacional (rpm) de cada atuador em cada passo da Sinergia I.



Figura 4.10: *Grasp* de um ovo, realizado pelo passo 3 da Sinergia I.

precisão em objetos muito pequenos (estes objetos não podem ser muito longos, caso contrário não seria possível realizar a sua *grasp* com esta sinergia, dado a interferência dos restantes dedos). Nos últimos dois passos, a mão fecha completamente. Desta maneira é possível realizar as 21 *grasps* categorizadas no Grupo I, como se pode ver na Tabela 4.7. Nesta Tabela, na *grasp* n.º 13 foi substituído o uso da bola de ping pong pelo ovo, dado às suas dimensões semelhantes e para provar a adaptabilidade de *ISR-SoftHand* a objetos frágeis.

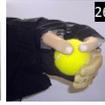
4.3.2 Sinergia II

Com a segunda sinergia apenas é possível realizar uma *grasp*:

- *Grasp* n.º 31.

Nesta sinergia o atuador 3, que controla o movimento dos três dedos, não é ativado. Esta sinergia foi projetada para *grasps* de objetos mais longos, onde os três dedos não devem interferir. Aqui foi definida uma menor velocidade de fecho, para gerar uma *grasp* suave e natural, como seria de esperar numa mão humana (Sinergia II na Figura 4.11).

Tabela 4.7: *Grasps* que conseguem ser efetuadas por cada um dos passos da Sinergia I.

Passo 1	Posição de descanso								
Passo 2									
Passo 3									
Passo 4									
Passo 5									
Passo 6									

4.3.3 Sinergia III

Com a terceira sinergia é possível realizar três *grasps*:

- *Grasp* n.º 4, 15 e 30.

Nesta sinergia o polegar mantém-se estático. Esta sinergia foi projetada de maneira a que o dedo indicador e os três restantes dedos fechem juntos, para a *grasp* de objetos com diferentes tamanhos (Sinergia III na Figura 4.11).

4.3.4 Sinergia IV

Com a quarta sinergia é possível realizar quatro *grasps*:

- *Grasp* n.º 17, 19, 29 e 32.

Ao visualizar os esquemas das *grasps* realizadas por esta sinergia, verifica-se que o dedo indicador está estendido. No entanto no *design* desta sinergia o dedo indicador flete um pouco, para assegurar que realiza a força de contacto suficiente para a estabilidade do objeto. O polegar fecha um pouco mais que os restantes dedos (como é visível na Sinergia IV da Tabela 4.8) para que consiga aplicar força de contacto suficiente para um maior número de objetos com diferentes dimensões. Como se pode ver na Tabela 4.5, a posição dos restantes três foi testada e projetada de maneira a que existissem pontos de contacto laterais da mão com o objeto, tornando a *grasp* mais estável.

4.3.5 Sinergia V

Com a quinta sinergia é possível realizar duas *grasps*:

- *Grasp* n.º 18 e 22.

Esta sinergia foi criada para particularmente para objetos de forma paralelepípeda, como mencionado na secção anterior. No seu *design* foi tido em conta a maneira como uma mão humana a realizaria este movimento. Depois de uma análise meticulosa da posição dos atuadores a agarrar vários objetos do tipo mencionado anteriormente, esta sinergia foi criada e revelou-se vantajosa.

O estiramento dos tendões para cada uma destas sinergias, bem como a velocidade de rotação dos atuadores utilizada, estão representados na Tabela 4.8 e na Figura 4.11.

Tabela 4.8: Valores do estiramento do tendão, em milímetros, por cada atuador em cada uma Sinergias II, III, IV e V. Atuador 1: polegar, Atuador 2: dedo indicador, Atuador 3: Restantes 3 dedos.

→ Atuador	Sinergia II			Sinergia III			Sinergia IV			Sinergia V		
	1 (mm)	2 (mm)	3 (mm)	1 (mm)	2 (mm)	3 (mm)	1 (mm)	2 (mm)	3 (mm)	1 (mm)	2 (mm)	3 (mm)
Aberto (Passo 1)	0	0	0	0	0	0	0	0	0	0	0	0
Fechado (Passo 2)	12,78	13,30	0	0	25,82	26,58	20,66	5,11	16,36	16,57	10,23	10,23

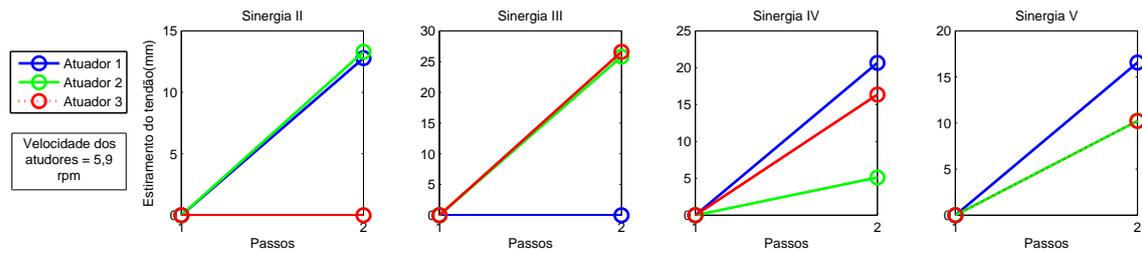


Figura 4.11: Estiramento dos tendões, em milímetros, em cada uma das Sinergias II, III, IV e V. Nestas a velocidade rotacional de todos os atuadores é mantida nas 5,9 rpm.

Capítulo 5

Conclusões e Trabalho Futuro

Nesta secção, discutem-se as principais estratégias para uma maior simplificação futura da *ISR-Softhand* em termos de número de atuadores e número de sinergias.

Em relação à redução do número de atuadores é discutida a sua consequência em relação ao número de possíveis *grasps* realizadas pela *ISR-Softhand*, utilizando a mesma estratégia de controlo (ou seja, as mesmas cinco sinergias adaptativas).

No final conclui-se qual das estratégias é mais apropriada de implementar para um controlo ainda mais simples da *ISR-Softhand* e adequado ao propósito de futuro controlo através do sinal proveniente dos sensores superficiais de EMG.

5.1 Estratégias para reduzir o número de atuadores

5.1.1 2 Atuadores

5.1.2 1 Atuador

5.2 Estratégias para reduzir o número de sinergias

5.3 Conclusões

5.1 Estratégia para reduzir o número de atuadores

Dado que um dos principais objetivos deste projeto é a criação de uma mão protética simples e de baixo custo, um breve estudo foi feito para entender como a redução do número de atuadores poderia afetar a funcionalidade da *ISR-Softhand*.

5.1.1 2 Atuadores

Se fossem aplicados apenas dois atuadores no protótipo, existiam sempre dois grupos de *grasps* que não seriam possíveis de realizar. Isso ocorre porque os Grupos II, III e IV têm a particularidade de ter sempre um atuador que não se move (Grupos II e III) ou que se move com um ritmo diferente em relação aos outros (Grupo IV). Neste caso, existiram três maneiras de acoplar a flexão dos dedos:

1. Acoplar a flexão do dedo indicador com os dedos 3, 4 e 5
2. Acoplar a flexão do polegar com o indicador
3. Acoplar a flexão do polegar com os dedos 3, 4 e 5.

A Tabela 5.1 mostra como é que esta combinação afetaria o desempenho do protótipo.

Um aspeto muito importante que foi notado durante os testes e a criação da primeira sinergia foi que a transição entre o terceira e o quarto passo da Sinergia I é essencial. O facto do dedo indicador fletir mais que os dedos 3, 4 e 5 fornece apoio lateral para a *grasp* de alguns objetos cilíndricos longos, como é o caso da *grasp* n.º 20. Portanto, como também pode ser visto na Tabela 5.1, a flexão do dedo indicador e dos dedos 3,4,5 deve permanecer independente. Isto é, a fim de reduzir a um atuador, a melhor opção é acoplar a flexão do polegar ou com do dedo indicador ou com os dedos 3, 4 e 5.

Tabela 5.1: *Grasps* que seriam impossíveis de realizar com uma versão da *ISR-SoftHand* com 2 atuadores. Na coluna das configurações “e” significa que as articulações estão acopladas e “+” significa que os dedos são atuados pelo mesmo atuador

Configuração	Número de <i>grasps</i> impossíveis					Total	Comentários	
	I.A	I.B	II	III	IV			V
Polegar + Todos os dedos	6	-	1	-	4	-	11	A flexão do dedo indicador e dos restantes dedos está acoplada. O Grupo II e IV não seriam realizáveis. As 6 <i>grasps</i> impossíveis do Grupo I são aquelas realizadas pelo passo 4. No entanto, se a posição inicial dos 3 dedos for colocada um pouco mais à frente que o dedo polegar, estas já seriam possíveis. Mesmo assim não seria possível realizar algumas das <i>grasps</i> de precisão realizadas no passo 3 da Sinergia I (<i>grasps</i> n.º 6 e 7). Incluí duas <i>grasp</i> do top 10 [11].
Polegar e Dedo indicador + Dedos 3, 4 e 5	-	-	-	3	4	-	7	A flexão do polegar e do dedo indicador é acoplada. O Grupo III e IV não seriam realizáveis, o que inclui uma <i>grasp</i> do top 10 [11].
Polegar e Dedos 3,4 and 5 + Dedo indicador	-	-	1	3	-	-	4	A flexão do polegar e dos dedos 3, 4 e 5 é acoplada. O Grupo II e III não seriam realizáveis. Nenhuma destas <i>grasp</i> pertence ao top 10 [11].

Estas duas categorizações podem então ser analisadas em termos de números de *grasps* e eficiência. Um estudo feito por Bullock et al. [11] categorizou o top 10 de *grasps* realizadas em tarefas do dia-a-dia. Considerando essas *grasps* (*Grasps* n.º 3, 5, 7, 8, 11, 12, 14, 16, 17 e 25), e com o objetivo de tornar estas *grasps* possíveis, o caso mais vantajoso é a acoplação da flexão do polegar com os dedos 3, 4 e 5. Com esta configuração 4 *grasps* (n.º 4, 15, 30 e 31) que pertencem aos Grupos II III não seriam possíveis. No entanto nenhuma destas pertence ao top 10 das *grasps* mais utilizadas nas tarefas do dia-a-dia.

Na outra configuração, com a acoplação do polegar com o dedo indicador, 7 *grasps* (n.º 4, 15, 17, 19, 29, 30 e 32) seriam impossíveis, destas, a *graps* número 17 está incluída no top 10.

5.1.2 1 Atuador

Com um único atuador, todos os dedos se deslocariam simultaneamente. Portanto, após o ajuste da posição inicial dos dedos e do polegar e da sua velocidade de flexão, seriam apenas possíveis de executar algumas das *grasps* do grupo principal. Pela razão já mencionada da importância do movimento independente do dedo indicador e dos restantes dedos, algumas *grasps* do Grupo I não seriam possíveis. Em resumo apenas 13 *grasps* seriam possíveis com um único atuador, as *grasps* nr. 1, 3, 6, 7, 10, 11, 12, 13, 14, 16, 26, 27 e 28. Seis destas *grasps* (*Grasps* nr. 3, 7, 11, 12, 14 e 16) pertencem ao topo 10 [11].

5.2 Estratégias para reduzir o número de sinergias

Tendo em conta todas as sinergias adaptativas criadas, é visível que a Sinergia I é uma sinergia bem concebida e abrangente, pois cobre 74,2 % do total de 31 *grasps* possíveis de realizar pela *ISR-Softand*.

Por outro lado, também é visível que o Grupo II e o Grupo V contêm muitas poucas *grasps*. O Grupo II foi projetado para executar as *grasps* que não necessitam

da interação dos dedos 3, 4 e 5. Neste caso, como pode ser visto na Tabela 4.3, as *grasps* número 9, 31 e 33 foram inicialmente colocadas neste grupo. No entanto como os objetos não são muito longos, as *grasps* número 9 e 31 podem ser realizadas pela Sinergia I. Num ponto de vista funcional a *grasp* número 31 não é muito utilizada no dia-a-dia [11], podendo ainda ser substituída pela *grasp* n.º 1 e, conseqüentemente, a Sinergia II poderá deixar de ser considerada.

Em relação ao Grupo V, tendo apenas duas *grasps* seria lógico supor que este grupo poderia também deixar de ser considerado, tal como a sinergia associada. Contudo, durante este projeto foi notória a importância desta sinergia para agarrar este tipo de objetos (finos e largos, tal como um livro), uma vez que nenhuma das restantes sinergias poderia fazer este tipo de *grasps* com a mesma estabilidade e uma das principais funcionalidades de qualquer tipo de prótese robótica é conseguir fazer *grasps* estáveis.

Sendo assim, na versão atual do ISR-SoftHand, só seria plausível excluir a Sinergia II.

Outra questão a considerar é que com uma versão implicada da *ISR-SoftHand* com apenas dois atuadores, em que a flexão do polegar e do dedo indicador são acopladas, as Sinergia III e IV não seriam possíveis e com a versão em que o dedo polegar e os dedos 3, 4 e 5 estão acoplados as Sinergias II e III também não eram possíveis de realizar.

5.3 Conclusões

Neste projeto foram relatados os resultados dos testes da mão contra uma categorização de 33 *grasps*, onde foram criadas cinco sinergias que conseguiram executar 31 das 33 *grasps* possíveis. Destas 31 *grasps* possíveis, 21 foram realizadas com a sinergia principal, a Sinergia I.

Demonstrou-se ainda a possibilidade da *grasp* de um ovo, como representação de um objeto frágil. Considerando que a mão não contém qualquer sensor de força nos

dedos, e todos as *grasps* são realizados com um controle sem qualquer tipo de *feedback* é notório que a complacência integrada da mão permite um alcance de um grande número de *grasps* com uma arquitetura de controlo simples.

As sinergias adaptativas são então possíveis de realizar em mãos robóticas com complacência integrada, dado que os pontos de contacto não têm que ser colocados precisamente nos contornos do objeto e podem ser criados pontos virtuais dentro do objeto. É a diferença entre os pontos colocados exatamente no contorno do objeto e os pontos virtuais e a complacência das articulações da mão robótica que determina a força aplicada no objeto.

É visível que as Sinergias II, III e IV são estruturadas para agarrar principalmente objetos cilíndricos e longos e Sinergia V é capaz de segurar objetos com forma de paralelepípeda. O principal desafio foi projetar uma sinergia que conseguisse fazer diferentes tipos de *grasps* (isto é, *power grasps*, *grasps* de precisão e intermédias). Por exemplo, pode-se observar que no passo 3 da Sinergia I, a mão pode atingir *grasps* de precisão (6, 7, 13, 14, 27, 33) e *power grasps* (3, 11, 26) e no Passo 4 consegue-se executar *grasps* de precisão (8, 9, 20, 24) e *grasps* intermédias (21, 25). Isto prova que a complacência integrada na mão robótica e também o conceito de sinergia adaptativa são bem aplicados neste projeto. Entre outras razões, uma das motivações para a criação de quatro passos intermédios na Sinergia I, era manter a gama de forças de contacto dentro de uma faixa razoável para agarrar objetos com tamanhos diferentes.

Também deve ser mencionado que das 31 *grasps*, podemos imitar 20 *graps* de forma semelhantes à da imagem na categorização utilizada e as restantes *grasps* também são possíveis de realizar, mas a imitação exata não foi possível. Isto acontece porque a *ISR-SoftHand* tem apenas 3 atuadores que limita a destreza da mesma.

Das conclusões e resultados descritas nas estratégias para reduzir os atuadores (Secção 5.1), podemos concluir que uma mão sub atuada com apenas um atuador podia apenas executar um número pequeno de *grasps* comparado com a versão da mão robótica prótica em uso neste projeto. Mas a configuração com dois atuadores, em que a flexão do polegar é acoplada com a flexão do dedo indicador, mostra-se promiss-

sora, tanto em termos de número de *grasps* realizáveis como em concepção mecânica. Portanto esta opção é relevante para ser explorada numa nova versão da *ISR-Softhand*.

Em relação às estratégias para reduzir o número de sinergias é evidente que mesmo com alguma redução na funcionalidade em relação à versão atual do *ISR-Softhand*, uma versão simplificada da mão é muito apelativa, devido à sua estrutura eletromecânica e controle simplificado. Pode ainda ser explorada uma versão inovadora para simplificar as sinergias da atual versão da *ISR-Softhand* sem reduzir a sua funcionalidade, esta abordagem tem em conta a utilização de uma única sinergia (a Sinergia I) com uma função de “desligar” para cada um dos atuadores. Ou seja, assim seria possível uma combinação das Sinergias II, III, IV e V com a sinergia principal. Por exemplo, para a utilização da Sinergia III, o utilizador poderia desligar o atuador que flete o polegar (atuador 1) e de seguida fletir a mão com o uso da de apenas uma sinergia. Logo, a partir destas conclusões, pode-se sugerir uma versão mais simples do *ISR-Softhand* com apenas 2 atuadores e, conseqüentemente, 2 sinergias, o que levaria a um controlo mais simples não necessitando do que mais que dois patamares de divisão de um único sinal de EMG para o simples controlo da mãos robótica antropomórfica próstética adaptativa.

Ao longo desta dissertação foi óbvia a necessidade de desenvolver mãos próstéticas mais simples, leves e eficazes. Os resultados deste projeto demonstraram como a complacência integrada e um *design* apropriado das sinergias adaptativas pode resultar em:

- Simplificação da estrutura mecânica (redução do número de atuadores, peso e custo)
- Simplificação do sistema mecatrónico (controlo sem *feedback* de força e menos atuadores)
- Simplificação do esquema de controlo (número reduzido de sinergias e de sistemas de parâmetros entrada)

Bibliografía

- [1] M. Luchetti, O. Montebanocci, N. Rossi, A. G. Cutti, and A. R. Sutin, “Autobiographical memory and psychological distress in a sample of upper-limb amputees,” *PloS one*, vol. 9, no. 6, 2014.
- [2] M. T. Mason and J. K. Salisbury Jr, *Robot hands and the mechanics of manipulation*. Cambridge, MA: The MIT Press, 1985.
- [3] T. Mouri, T. Endo, and H. Kawasaki, “Review of gifu hand and its application,” *Mechanics based design of structures and machines*, vol. 39, no. 2, pp. 210–228, 2011.
- [4] M. A. Diftler, J. Mehling, M. E. Abdallah, N. A. Radford, L. B. Bridgwater, A. M. Sanders, R. S. Askew, D. M. Linn, J. D. Yamokoski, F. Permenter *et al.*, “Robonaut 2-the first humanoid robot in space,” pp. 2178–2183, 2011.
- [5] M. Grebenstein, M. Chalon, G. Hirzinger, and R. Siegwart, “Antagonistically driven finger design for the anthropomorphic dlr hand arm system,” pp. 609–616, 2010.
- [6] L. Birglen, C. M. Gosselin, and T. Laliberté, *Underactuated robotic hands*. Springer, 2008, vol. 40.
- [7] J. Pons, E. Rocon, R. Ceres, D. Reynaerts, B. Saro, S. Levin, and W. Van Moorleghem, “The manus-hand dextrous robotics upper limb prosthesis: mechanical and manipulation aspects,” *Autonomous Robots*, vol. 16, no. 2, pp. 143–163, 2004.

- [8] R. Cabas, L. M. Cabas, and C. Balaguer, “Optimized design of the underactuated robotic hand,” pp. 982–987, 2006.
- [9] J. T. Belter and A. M. Dollar, “Performance characteristics of anthropomorphic prosthetic hands,” pp. 1–7, 2011.
- [10] C. Pylatiuk, S. Schulz, and L. Döderlein, “Results of an internet survey of myoelectric prosthetic hand users,” *Prosthetics and orthotics international*, vol. 31, no. 4, pp. 362–370, 2007.
- [11] I. Bullock, J. Zheng, S. Rosa, C. Guertler, and A. Dollar, “Grasp frequency and usage in daily household and machine shop tasks,” *Haptics, IEEE Transactions*, vol. 6, no. 3, pp. 296–308, 2013.
- [12] A. D. Roche, H. Rehbaum, D. Farina, and O. C. Aszmann, “Prosthetic myoelectric control strategies: A clinical perspective,” *Current Surgery Reports*, vol. 2, no. 3, pp. 1–11, 2014.
- [13] M. Tavakoli and de Almeida A. T., “Adaptive under-actuated anthropomorphic hand: Isr-softhand,” *2014 IEEE International Conference on Robotics and Automation, ICRA*, 2014.
- [14] T. Feix, “Human grasping database @ONLINE,” <http://grasp.xief.net/>, [Acedido: Outubro-2013].
- [15] t. f. e. Wikipedia, “Hand @ONLINE:,” http://en.wikipedia.org/wiki/File:Scheme_human_hand_bones-en.svg, [Acedido: Maio-2014].
- [16] T. Feix, “Anthropomorphic hand optimization based on a latent space analysis,” Master’s thesis, Technische Universität Wien, October 2011.
- [17] I. A. Kapandji and L. H. Honoré, *The physiology of the joints: annotated diagrams of the mechanics of the human joints*. E. & S. Livingstone London, 1970, vol. 1, no. 2.

-
- [18] J. A. Katarincic, “Thumb kinematics and their relevance to function.” *Hand clinics*, vol. 17, no. 2, pp. 169–174, 2001.
- [19] Morphopedics, “Scaphoid fractures @ONLINE:,” <http://morphopedics.wikidot.com/scaphoid-fractures>, [Acedido: Junho-2014].
- [20] C. Melchiorri and M. Kaneko, “Robot hands,” in *Springer Handbook of Robotics*, 2008, pp. 345–359.
- [21] K. J. Waldron and J. P. Schmiedeler, “Kinematics,” in *Springer Handbook of Robotics*, 2008, pp. 9–33.
- [22] R. Featherstone and D. E. Orin, “Dynamics,” in *Springer Handbook of Robotics*, 2008, pp. 35–65.
- [23] L. Ross, S. Fardo, J. Masterson, and R. Towers, *Unit 1 - Principles of Robotics*. Goodheart-Willcox Publisher, 2010.
- [24] S. Harangue, “Designing for degrees of freedom @ONLINE,” <http://wertel.blogspot.pt/2009/11/designing-for-degrees-of-freedom.html>, [Acedido: Junho-2014].
- [25] M. Catalano, G. Grioli, E. Farnioli, A. Serio, C. Piazza, and A. Bicchi, “Adaptive synergies for the design and control of the pisa/iit soft hand,” *The International Journal of Robotics Research*, vol. 33, no. 5, pp. 768–782, 2014.
- [26] V. Scheinman and J. M. McCarthy, “Mechanisms and actuation,” in *Springer Handbook of Robotics*, 2008, pp. 67–86.
- [27] M. G. Catalano, G. Grioli, A. Serio, E. Farnioli, C. Piazza, and A. Bicchi, “Adaptive synergies for a humanoid robot hand.” pp. 7–14, 2012.
- [28] G. Grioli, M. Catalano, E. Silvestro, S. Tono, and A. Bicchi, “Adaptive synergies: an approach to the design of under-actuated robotic hands,” pp. 1251–1256, 2012.
-

- [29] L. Villani and J. D. Schutter, “Force control,” in *Springer Handbook of Robotics*, 2008, pp. 161–185.
- [30] W. Wang, R. N. Loh, and E. Y. Gu, “Passive compliance versus active compliance in robot-based automated assembly systems,” *Industrial Robot: An International Journal*, vol. 25, no. 1, pp. 48–57, 1998.
- [31] M. Grebenstein, A. Albu-Schäffer, T. Bahls, M. Chalon, O. Eiberger, W. Friedl, R. Gruber, S. Haddadin, U. Hagn, R. Haslinger *et al.*, “The dlr hand arm system,” pp. 3175–3182, 2011.
- [32] C. Y. Brown and H. H. Asada, “Inter-finger coordination and postural synergies in robot hands via mechanical implementation of principal components analysis,” pp. 2877–2882, 2007.
- [33] R. R. Ma, L. U. Odhner, and A. M. Dollar, “A modular, open-source 3d printed underactuated hand,” pp. 2737–2743, 2013.
- [34] A. M. Dollar and R. D. Howe, “The sdm hand as a prosthetic terminal device: a feasibility study,” pp. 978–983, 2007.
- [35] D. Kortenkamp and R. G. Simmons, “Robotic systems architectures and programming,” in *Springer Handbook of Robotics*, 2008, pp. 187–206.
- [36] J. S. Kelso, D. L. Southard, and D. Goodman, “On the coordination of two-handed movements.” *Journal of Experimental Psychology: Human Perception and Performance*, vol. 5, no. 2, p. 229, 1979.
- [37] A. Bicchi, M. Gabbicini, and M. Santello, “Modelling natural and artificial hands with synergies,” *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 366, no. 1581, pp. 3153–3161, 2011.
- [38] J. R. Napier, “The prehensile movements of the human hand,” *The Journal of bone and joint surgery*, vol. 38 B, pp. 902–913, 1956.

- [39] T. Iberall, G. Bingham, and M. Arbib, “Opposition space as a structuring concept for the analysis of skilled hand movements,” *Experimental brain research series*, vol. 15, pp. 158–173, 1986.
- [40] J. Landsmeer, “Power grip and precision handling,” *Annals of the rheumatic diseases*, vol. 21, no. 2, p. 164, 1962.
- [41] N. Kamakura, M. Matsuo, H. Ishii, F. Mitsuboshi, and Y. Miura, “Patterns of static prehension in normal hands.” *The American journal of occupational therapy: official publication of the American Occupational Therapy Association*, vol. 34, no. 7, pp. 437–445, 1980.
- [42] T. Feix, R. Pawlik, H. Schmiedmayer, J. Romero, and D. Kragic, “A comprehensive grasp taxonomy,” pp. 2–3, 2009.
- [43] I. G. Schlesinger, “Der mechanische aufbau der künstlichen glieder,” in *Ersatzglieder und Arbeitshilfen*. Springer, 1919, pp. 321–661.
- [44] I. A. Kapandji, *Physiology of the Joints (Upper Extremities)*, 5th ed. Churchill Livingstone, 1982, vol. 1.
- [45] J. M. Elliott and K. Connolly, “A classification of manipulative hand movements,” *Developmental Medicine & Child Neurology*, vol. 26, no. 3, pp. 283–296, 1984.
- [46] M. R. Cutkosky, “On grasp choice, grasp models, and the design of hands for manufacturing tasks,” *Robotics and Automation, IEEE Transactions*, vol. 5, no. 3, pp. 269–279, 1989.
- [47] O. Bock, “Sensorhand speed @ONLINE,” http://www.ottobock.com/cps/rde/xchg/ob_com_en/hs.xsl/3652.html, [Acedido: Outubro-2013].
- [48] —, “Michel angelo hand @ONLINE,” <http://www.living-with-michelangelo.com/gb/home/>, [Acedido: Outubro-2013].

- [49] TouchBionics, “iLIMB hand @ONLINE,” <http://www.touchbionics.com/>, [Acedido: Outubro-2013].
- [50] RSLsteeper, “Bebionic hand @ONLINE,” <http://www.rslsteeper.com/>, [Acedido: Outubro-2013].
- [51] V. Systems, “Vincent hand @ONLINE,” <http://handprothese.de/vincent-hand/>, [Acedido: Outubro-2013].
- [52] C. Cipriani, M. Controzzi, and M. C. Carrozza, “Objectives, criteria and methods for the design of the smarthand transradial prosthesis,” *Robotica*, vol. 28, no. 6, pp. 919–927, 2010.
- [53] —, “The smarthand transradial prosthesis,” *Journal of neuroengineering and rehabilitation*, vol. 8, no. 1, p. 29, 2011.
- [54] MekaBot, *Meka H2 Compliant Hand Datasheet*, MekaBot, 2009.
- [55] P. s.l.r. grasping innovation, “Th2 azzurra hand @ONLINE:,” <http://www.prensilia.com/index.php?q=en/node/40>, [Acedido: Junho-2014].
- [56] *Michel Angelo operation manual*, Otto Bock, 2012.
- [57] S. Schulz, “First experiences with the vincent hand,” 2011.
- [58] R. J. Schwarz and C. L. Taylor, “The anatomy and mechanics of the human hand,” *Artificial Limbs*, vol. 2, pp. 22–35, 1955.
- [59] M. Santello, M. Flanders, and J. F. Soechting, “Postural hand synergies for tool use,” *The Journal of Neuroscience*, vol. 18, no. 23, pp. 10 105–10 115, 1998.
- [60] E. J. Weiss and M. Flanders, “Muscular and postural synergies of the human hand,” *Journal of Neurophysiology*, vol. 92, no. 1, pp. 523–535, 2004.
- [61] M. L. Latash, J. K. Shim, A. V. Smilga, and V. M. Zatsiorsky, “A central back-coupling hypothesis on the organization of motor synergies: a physical metaphor and a neural model,” *Biological cybernetics*, vol. 92, no. 3, pp. 186–191, 2005.

-
- [62] M. Santello, G. Baud-Bovy, and H. Jörntell, “Neural bases of hand synergies,” *Frontiers in computational neuroscience*, vol. 7, 2013.
- [63] T. A. Easton, “On the normal use of reflexes: The hypothesis that reflexes form the basic language of the motor program permits simple, flexible specifications of voluntary movements and allows fruitful speculation,” *American Scientist*, pp. 591–599, 1972.
- [64] M. Ciocarlie, C. Goldfeder, and P. K. Allen, “Dimensionality reduction for hand-independent dexterous robotic grasping,” pp. 3270–3275, 2007.
- [65] M. Gabiccini, A. Bicchi, D. Prattichizzo, and M. Malvezzi, “On the role of hand synergies in the optimal choice of grasping forces,” *Autonomous Robots*, vol. 31, no. 2-3, pp. 235–252, 2011.
- [66] M. Carrozza, G. Cappiello, S. Micera, B. Edin, L. Beccai, and C. Cipriani, “Design of a cybernetic hand for perception and action,” *Biological Cybernetics*, vol. 95, no. 6, pp. 629–644, 2006.
- [67] B. Popov, “The bio-electrically controlled prosthesis,” *Journal of Bone and Joint Surgery, British Volume*, vol. 47-B, no. 3, pp. 421–424, 1965.
- [68] L. Hargrove, K. Englehart, and B. Hudgins, “A comparison of surface and intramuscular myoelectric signal classification,” *Biomedical Engineering, IEEE Transactions*, vol. 54, no. 5, pp. 847–853, 2007.
- [69] O. bock, “Dmc plus @ONLINE,” http://www.ottobock.com/cps/rde/xchg/ob-com_en/hs.xsl/3722.html.
- [70] TouchBionics, “iLIMB ultra revolution hand @ONLINE,” <http://www.touchbionics.com/products/active-prostheses/i-limb-ultra-revolution>, [Acedido: Julho-2014].
- [71] A. Fougner, O. Stavdahl, P. Kyberd, Y. Losier, and P. Parker, “Control of upper limb prostheses: Terminology and proportional myoelectric control, a review,”
-

- Neural Systems and Rehabilitation Engineering, IEEE Transactions*, vol. 20, no. 5, pp. 663–677, 2012.
- [72] B. Kent, N. Karnati, and E. Engeberg, “Electromyogram synergy control of a dexterous artificial hand to unscrew and screw objects,” *Journal of NeuroEngineering and Rehabilitation*, vol. 11, no. 1, p. 41, 2014.
- [73] B. Hudgins, P. Parker, and R. Scott, “A new strategy for multifunction myoelectric control,” *Biomedical Engineering, IEEE Transactions*, vol. 40, no. 1, pp. 82–94, 1993.
- [74] P. O. Int, “Deka arm @ONLINE:,” <http://www.dekaresearch.com/index.shtml>, [Acedido: Outubro-2013].
- [75] B. Lee, F. J. Attenello, C. Y. Liu, M. P. McLoughlin, and M. L. Apuzzo, “Recapitulating flesh with silicon and steel: Advancements in upper extremity robotic prosthetics,” *World neurosurgery*, vol. 81, no. 5, pp. 730–741, 2014.
- [76] ROBOTIS, “Robotis Productes - Dynamixel AX-12A @ONLINE:,” http://www.robotis-shop-en.com/shop/step1.php?number=873&b_code=B20070914050547&c_code=C20111128024236, [Acedido: Novembro-2013].
- [77] —, “Dynamixel AX-12A @ONLINE,” http://www.robotis.com/xen/dynamixel_en, [Acedido: Novembro-2013].

Anexos

Anexo A

Publicações

Submetidas

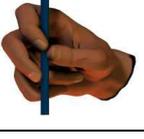
“Adaptive Synergies of the ISR-Soft Hand” **submetido a:** International Journal of Robotics Research

“Underactuated Anthropomorphic Hands: Actuation Strategies for a Better Functionality ” **submetido a:** Journal of Robotics and Autonomous Systems

Anexo B

Categorização de Feix et al.

Categorização desenvolvida por Feix et al. [16]. Disponível online em [14]

Nr.	Name	Picture	Type	Opp. Type	Thumb Pos.	VF1	VF2	VF3	#
1	Large Diameter		Power	Palm	Abd	P	2-5		10
2	Small Diameter		Power	Palm	Abd	P	2-5		3
3	Medium Wrap		Power	Palm	Abd	P	2-5		6
4	Adducted Thumb		Power	Palm	Add	P	2-5	1	2
5	Light Tool		Power	Palm	Add	P	2-5	(1)	2
6	Prismatic 4 Finger		Precision	Pad	Abd	1	2-5		4
7	Prismatic 3 Finger		Precision	Pad	Abd	1	2-4		4
8	Prismatic 2 Finger		Precision	Pad	Abd	1	2-3		2
9	Palmar Pinch		Precision	Pad	Abd	1	2		12

Nr.	Name	Picture	Type	Opp. Type	Thumb Pos.	VF1	VF2	VF3	#
10	Power Disk		Power	Palm	Abd	P	2-5		3
11	Power Sphere		Power	Palm	Abd	P	2-5		8
12	Precision Disk		Precision	Pad	Abd	1	2-5		2+ 2+ 1
13	Precision Sphere		Precision	Pad	Abd	1	2-5		6
14	Tripod		Precision	Pad	Abd	1	2-3		8
15	Fixed Hook		Power	Palm	Add	P	2-5		2
16	Lateral		Inter- mediate	Side	Add	1	2		12
17	Index Finger Extension		Power	Palm	Add	P	3-5	2	4
18	Extension Type		Power	Pad	Abd	1	2-4		1

Nr.	Name	Picture	Type	Opp. Type	Thumb Pos.	VF1	VF2	VF3	#
19	Distal Type		Power	Pad	Abd	1	2-5		2
20	Writing Tripod		Precision	Side	Abd	1	3		6
21	Tripod Variation		Inter-mediate	Side	Abd	1	3-4		1
22	Parallel Extension		Precision	Pad	Add	1	2-5		5
23	Adduction Grip		Inter-mediate	Side	Abd	1	2		3
24	Tip Pinch		Precision	Pad	Abd	1	2		9
25	Lateral Tripod		Inter-mediate	Side	Add	1	3		1
26	Sphere 4 Finger		Power	Pad	Abd	1	2-4		1
27	Quadpod		Precision	Pad	Abd	1	2-4		2

Nr.	Name	Picture	Type	Opp. Type	Thumb Pos.	VF1	VF2	VF3	#
28	Sphere 3 Finger		Power	Pad	Abd	1	2-3		1
29	Stick		Intermediate	Side	Add	1	2		1
30	Palmar		Power	Palm	Add	1	2-5		1
31	Ring		Power	Pad	Abd	1	2		1
32	Ventral		Intermediate	Side	Add	1	2		1
33	Inferior Pincer		Precision	Pad	Abd	1	2		1

	Power	Palm	
	Intermediate	Side	
	Precision	Pad	
			Abducted
			Adducted

Anexo C

Códigos Matlab[®]

Sinergia I:

```
function [] = Sinergia.1()  
clc;  
clear all;  
  
%iniciar_global();  
global ERRBIT_VOLTAGE  
ERRBIT_VOLTAGE = 1;  
global ERRBIT_ANGLE  
ERRBIT_ANGLE = 2;  
global ERRBIT_OVERHEAT  
ERRBIT_OVERHEAT = 4;  
global ERRBIT_RANGE  
ERRBIT_RANGE = 8;  
global ERRBIT_CHECKSUM  
ERRBIT_CHECKSUM = 16;  
global ERRBIT_OVERLOAD  
ERRBIT_OVERLOAD = 32;  
global ERRBIT_INSTRUCTION  
ERRBIT_INSTRUCTION = 64;  
  
global COMM_TXSUCCESS
```

```
COMM_TXSUCCESS      = 0;
global COMM_RXSUCCESS
COMM_RXSUCCESS      = 1;
global COMM_TXFAIL
COMM_TXFAIL         = 2;
global COMM_RXFAIL
COMM_RXFAIL         = 3;
global COMM_TXERROR
COMM_TXERROR        = 4;
global COMM_RXWAITING
COMM_RXWAITING      = 5;
global COMM_RXTIMEOUT
COMM_RXTIMEOUT      = 6;
global COMM_RXCORRUPT
COMM_RXCORRUPT      = 7;
```

```
%libreria que permite controlar o Dynamixel
```

```
loadlibrary('dynamixel', 'dynamixel.h');
```

```
libfunctions('dynamixel');
```

```
DEFAULT_PORTNUM = 5;  %COM5
```

```
DEFAULT_BAUDNUM = 1;  %1Mbps
```

```
%la sinergia
```

```
%iniciar variaveis
```

```
P_PRESENT_POSITION = 36;
```

```
P_Moving = 46;
```

```
P_GOAL_POSITION = 30;
```

```
T_LIMIT=34;
```

```
P_GOAL_SPEED      = 32;
```

```
%Defenir ID's para cada actuador
```

```
id_9 = 9; % id do actuador que controla os 3 dedos
```

```
id_4= 4; %actuador que controa o dedo indicador
```

```
id_13= 13; %actuador que conrola o polegar
```

```

%posicoes do software Roboplus

int32 GoalPos9;
int32 GoalPos4;
int32 GoalPos13;

GoalPos9 = [135 320 420 480 510 600];
GoalPos4 = [370 520 660 680 730 905];
GoalPos13 = [620 730 800 820 940 1023];
int32 index;

%calibrar
calllib('dynamixel','dxl_write_word',id_9,T_LIMIT,1023);
calllib('dynamixel','dxl_write_word',id_4,T_LIMIT,1023);
calllib('dynamixel','dxl_write_word',id_13,T_LIMIT,1023);

while (1)
    reply = input('Iniciar Sinergia I [S/N]: ', 's');
    if isempty(reply)
        continue;
    elseif (reply == 'S' || reply == 's')
        reply2 = input('Escolha o passo pretendido, de 1 a 6: ', 's')...
            ;
        if (reply2 == '1' || reply2 == '2')
            %definir velocidade
            calllib('dynamixel','dxl_write_word',id_13,P_GOAL_SPEED...
                ,0); % velocidade ao maximo
            calllib('dynamixel','dxl_write_word',id_9,P_GOAL_SPEED,0)...
                ;
            calllib('dynamixel','dxl_write_word',id_4,P_GOAL_SPEED,0)...
                ;

            if (reply2 == '1')

```

```

    index=1;
    %definir posicao
    calllib('dynamixel', 'dxl_initialize', ...
            DEFAULT_PORTNUM, DEFAULT_BAUDNUM);
    calllib('dynamixel', 'dxl_write_word', id_4, ...
            P_GOAL_POSITION, GoalPos4(index));
    calllib('dynamixel', 'dxl_write_word', id_9, ...
            P_GOAL_POSITION, GoalPos9(index));
    calllib('dynamixel', 'dxl_write_word', id_13, ...
            P_GOAL_POSITION, GoalPos13(index));
end

if (reply2 == '2')
    index=2;
    %definir posicao
    calllib('dynamixel', 'dxl_initialize', ...
            DEFAULT_PORTNUM, DEFAULT_BAUDNUM);
    calllib('dynamixel', 'dxl_write_word', id_4, ...
            P_GOAL_POSITION, GoalPos4(index));
    calllib('dynamixel', 'dxl_write_word', id_9, ...
            P_GOAL_POSITION, GoalPos9(index));
    calllib('dynamixel', 'dxl_write_word', id_13, ...
            P_GOAL_POSITION, GoalPos13(index));
end

end

if (reply2 == '3' || reply2 == '4')
    %definir velocidade
    calllib('dynamixel', 'dxl_write_word', id_13, P_GOAL_SPEED...
            ,100);
    calllib('dynamixel', 'dxl_write_word', id_9, P_GOAL_SPEED...
            ,100);
    calllib('dynamixel', 'dxl_write_word', id_4, P_GOAL_SPEED...
            ,100);
end

```

```

if (reply2 == '3')
    index=3;
    %definir posicao
    calllib('dynamixel','dxl_write_word',id_4,...
        P_GOAL_POSITION,GoalPos4(index));
    calllib('dynamixel','dxl_write_word',id_9,...
        P_GOAL_POSITION,GoalPos9(index));
    calllib('dynamixel','dxl_write_word',id_13,...
        P_GOAL_POSITION,GoalPos13(index));
end

if (reply2 == '4')
    index=4;
    %definir posicao
    calllib('dynamixel','dxl_write_word',id_4,...
        P_GOAL_POSITION,GoalPos4(index));
    calllib('dynamixel','dxl_write_word',id_9,...
        P_GOAL_POSITION,GoalPos9(index));
    calllib('dynamixel','dxl_write_word',id_13,...
        P_GOAL_POSITION,GoalPos13(index));
end

end

if (reply2 == '5' || reply2 == '6')
    %definir velocidade
    calllib('dynamixel','dxl_write_word',id_13,P_GOAL_SPEED...
        ,50); % 5% da velocidade maxima
    calllib('dynamixel','dxl_write_word',id_9,P_GOAL_SPEED...
        ,100); % 10% da velocidade maxima
    calllib('dynamixel','dxl_write_word',id_4,P_GOAL_SPEED...
        ,100);

    if (reply2 == '5')
        index=5;
        %definir posicao

```

```

    calllib('dynamixel', 'dxl_initialize', ...
           DEFAULT_PORTNUM, DEFAULT_BAUDNUM);
    calllib('dynamixel', 'dxl_write_word', id_4, ...
           P_GOAL_POSITION, GoalPos4(index2));
    calllib('dynamixel', 'dxl_write_word', id_9, ...
           P_GOAL_POSITION, GoalPos9(index2));
    calllib('dynamixel', 'dxl_write_word', id_13, ...
           P_GOAL_POSITION, GoalPos13(index2));
end

if (reply2 == '6')
    index=5;
    %definir posicao
    calllib('dynamixel', 'dxl_initialize', ...
           DEFAULT_PORTNUM, DEFAULT_BAUDNUM);
    calllib('dynamixel', 'dxl_write_word', id_4, ...
           P_GOAL_POSITION, GoalPos4(index2));
    calllib('dynamixel', 'dxl_write_word', id_9, ...
           P_GOAL_POSITION, GoalPos9(index2));
    calllib('dynamixel', 'dxl_write_word', id_13, ...
           P_GOAL_POSITION, GoalPos13(index2));
end
end

Moving4 = 1;
Moving9 = 1;
Moving13 = 1;
while Moving4 == 1 && Moving9 == 1 && Moving13 == 1
    %Ler a posicao atual
    PresentPos4 = int32(calllib('dynamixel', 'dxl_read_word', ...
                               id_4, P_PRESENT_POSITION));
    PresentPos9 = int32(calllib('dynamixel', 'dxl_read_word', ...
                               id_9, P_PRESENT_POSITION));
    PresentPos13 = int32(calllib('dynamixel', 'dxl_read_word', ...
                                 id_13, P_PRESENT_POSITION));

```

```

CommStatus = int32(calllib('dynamixel','dxl_get_result'))...
;
if CommStatus == COMM_RXSUCCESS
    Position4 =[GoalPos4(index) PresentPos4];
    Position9 =[GoalPos9(index) PresentPos9];
    Position13 =[GoalPos13(index) PresentPos13];
    %disp(Position4);
    %disp(Position9);
    %disp(Position13);
    PrintErrorCode();
else
    PrintCommStatus(CommStatus);
    break;
end

%Verificar se o moviemnto foi realizado
Moving4 = int32(calllib('dynamixel','dxl_read_byte',id_4,...
    P_Moving));
Moving9 = int32(calllib('dynamixel','dxl_read_byte',id_9,...
    P_Moving));
Moving13 = int32(calllib('dynamixel','dxl_read_byte',...
    id_13,P_Moving));
CommStatus = int32(calllib('dynamixel','dxl_get_result'))...
;
if CommStatus == COMM_RXSUCCESS
    if Moving4 == 0 && Moving9 == 0 && Moving13 == 0
        if index == 1
            index=2;

        else
            index=1;

        end
    end
end
PrintErrorCode();

```

```

        else
            PrintCommStatus (CommStatus);
            break;

        end

    end

elseif (reply == 'n' || reply == 'N')
    break;
end

end

end

% Verificar se o valor das variaveis 'e correto
function [] = PrintErrorCode()
global ERBIT_VOLTAGE
global ERBIT_ANGLE
global ERBIT_OVERHEAT
global ERBIT_RANGE
global ERBIT_CHECKSUM
global ERBIT_OVERLOAD
global ERBIT_INSTRUCTION

if int32 (calllib ('dynamixel', 'dxl_get_rxpacket_error', ERBIT_VOLTAGE...
)) == 1
    disp ('Input Voltage Error!');
elseif int32 (calllib ('dynamixel', 'dxl_get_rxpacket_error', ...
ERBIT_ANGLE)) == 1
    disp ('Angle limit error!');
elseif int32 (calllib ('dynamixel', 'dxl_get_rxpacket_error', ...
ERBIT_OVERHEAT)) == 1
    disp ('Overheat error!');

```

```

elseif int32(calllib('dynamixel','dxl_get_rxpacket_error',...
    ERBBIT_RANGE)) == 1
    disp('Out of range error!');
elseif int32(calllib('dynamixel','dxl_get_rxpacket_error',...
    ERBBIT_CHECKSUM)) == 1
    disp('Checksum error!');
elseif int32(calllib('dynamixel','dxl_get_rxpacket_error',...
    ERBBIT_OVERLOAD)) == 1
    disp('Overload error!');
elseif int32(calllib('dynamixel','dxl_get_rxpacket_error',...
    ERBBIT_INSTRUCTION)) == 1
    disp('Instruction code error!');
end
end

% Verificar o estado dos atuadores
function [] = PrintCommStatus(CommStatus)

global COMM_TXFAIL
global COMM_RXFAIL
global COMM_TXERROR
global COMM_RXWAITING
global COMM_RXTIMEOUT
global COMM_RXCORRUPT

switch(CommStatus)
    case COMM_TXFAIL
        disp('COMM_TXFAIL : Failed transmit instruction packet!');
    case COMM_TXERROR
        disp('COMM_TXERROR: Incorrect instruction packet!');
    case COMM_RXFAIL
        disp('COMM_RXFAIL: Failed get status packet from device!');
    case COMM_RXWAITING
        disp('COMM_RXWAITING: Now recieving status packet!');
    case COMM_RXTIMEOUT

```

```
        disp('COMM_RXTIMEOUT: There is no status packet!');
    case COMM_RXCORRUPT
        disp('COMM_RXCORRUPT: Incorrect status packet!');
    otherwise
        disp('This is unknown error code!');
end
end
```

Sinergia II:

```
function [] = Sinergia_2 ()

%iniciar_global();
global ERRBIT_VOLTAGE
ERRBIT_VOLTAGE = 1;
global ERRBIT_ANGLE
ERRBIT_ANGLE = 2;
global ERRBIT_OVERHEAT
ERRBIT_OVERHEAT = 4;
global ERRBIT_RANGE
ERRBIT_RANGE = 8;
global ERRBIT_CHECKSUM
ERRBIT_CHECKSUM = 16;
global ERRBIT_OVERLOAD
ERRBIT_OVERLOAD = 32;
global ERRBIT_INSTRUCTION
ERRBIT_INSTRUCTION = 64;

global COMM_TXSUCCESS
COMM_TXSUCCESS = 0;
global COMM_RXSUCCESS
COMM_RXSUCCESS = 1;
global COMM_TXFAIL
COMM_TXFAIL = 2;
global COMM_RXFAIL
```

```

COMM_RXFAIL          = 3;
global COMM_TXERROR
COMM_TXERROR         = 4;
global COMM_RXWAITING
COMM_RXWAITING       = 5;
global COMM_RXTIMEOUT
COMM_RXTIMEOUT       = 6;
global COMM_RXCORRUPT
COMM_RXCORRUPT       = 7;

%libreria que permite controlar o Dynamixel
loadlibrary('dynamixel', 'dynamixel.h');
libfunctions('dynamixel');

DEFAULT_PORTNUM = 5;  %COM5
DEFAULT_BAUDNUM = 1;  %1Mbps

%2a sinergia
%iniciar variaveis
P_GOAL_POSITION = 30;
P_PRESENT_POSITION = 36;
P_Moving = 46;
P_GOAL_SPEED     = 32;

%Defenir ID's para cada actuador
id_9 = 9; % id do actuador que controla os 3 dedos
id_4 = 4; %actuador que controa o dedo indicador
id_13 = 13; %actuador que conrola o polegar

%posicoes do software Roboplus
int32 GoalPos9;
int32 GoalPos4;
int32 GoalPos13;
GoalPos9 = [80 80];
GoalPos4 = [400 660];

```

```

GoalPos13 = [620 870];
int32 index;

%iniciar
calllib('dynamixel', 'dxl_initialize', DEFAULT_PORTNUM, ...
        DEFAULT_BAUDNUM);

while (1)
    reply = input('Iniciar a Sinergia II [S/N]: ', 's');
    if isempty(reply)
        continue;
    elseif (reply == 'S' || reply == 's')
        reply2 = input('Abrir [A], Fechar [F]: ', 's');
        %definir velocidade
        calllib('dynamixel', 'dxl_write_word', id_13, P_GOAL_SPEED, 100);
        calllib('dynamixel', 'dxl_write_word', id_9, P_GOAL_SPEED, 100);
        calllib('dynamixel', 'dxl_write_word', id_4, P_GOAL_SPEED, 100);
        if (reply2 == 'A' || reply2 == 'a')
            index = 1;
            %definir posicao
            calllib('dynamixel', 'dxl_write_word', id_4, P_GOAL_POSITION...
                    , GoalPos4(index));
            calllib('dynamixel', 'dxl_write_word', id_9, P_GOAL_POSITION...
                    , GoalPos9(index));
            calllib('dynamixel', 'dxl_write_word', id_13, ...
                    P_GOAL_POSITION, GoalPos13(index));
        end

        if (reply2 == 'F' || reply2 == 'f')
            index = 2;
            %definir posicao
            calllib('dynamixel', 'dxl_write_word', id_4, P_GOAL_POSITION...
                    , GoalPos4(index));
            calllib('dynamixel', 'dxl_write_word', id_9, P_GOAL_POSITION...
                    , GoalPos9(index));

```

```

        calllib('dynamixel', 'dxl_write_word', id_13, ...
            P_GOAL_POSITION, GoalPos13(index));
    end

Moving4 = 1;
Moving9 = 1;
Moving13 = 1;
while Moving4 == 1 && Moving9 == 1 && Moving13 == 1
    %Ler a posicao atual
    PresentPos4 = int32(calllib('dynamixel', 'dxl_read_word', ...
        id_4, P_PRESENT_POSITION));
    PresentPos9 = int32(calllib('dynamixel', 'dxl_read_word', ...
        id_9, P_PRESENT_POSITION));
    PresentPos13 = int32(calllib('dynamixel', 'dxl_read_word', ...
        id_13, P_PRESENT_POSITION));
    CommStatus = int32(calllib('dynamixel', 'dxl_get_result')) ...
        ;
    if CommStatus == COMM_RXSUCCESS
        Position4 =[GoalPos4(index) PresentPos4];
        Position9 =[GoalPos9(index) PresentPos9];
        Position13 =[GoalPos13(index) PresentPos13];
        %disp(Position4);
        %disp(Position9);
        %disp(Position13);
        PrintErrorCode();
    else
        PrintCommStatus(CommStatus);
        break;
    end

    %Verificar se o moviemnto foi realizado
    Moving4 = int32(calllib('dynamixel', 'dxl_read_byte', id_4, ...
        P_Moving));
    Moving9 = int32(calllib('dynamixel', 'dxl_read_byte', id_9, ...
        P_Moving));

```

```

Moving13 = int32(calllib('dynamixel','dxl_read_byte',...
    id_13,P.Moving));
CommStatus = int32(calllib('dynamixel','dxl_get_result'))...
    ;
if CommStatus == COMM_RXSUCCESS
    if Moving4 == 0 && Moving9 == 0 && Moving13 == 0
        if index == 1
            index=2;

        else
            index=1;

        end

    end

    PrintErrorCode();

else
    PrintCommStatus(CommStatus);
    break;

end

end

elseif(reply == 'n' || reply=='N')
    break;
end

end

end

% Verificar se o valor das variaveis 'e correto
function [] = PrintErrorCode()
global ERBIT_VOLTAGE
global ERBIT_ANGLE
global ERBIT_OVERHEAT
global ERBIT_RANGE

```

```

global ERRBIT_CHECKSUM
global ERRBIT_OVERLOAD
global ERRBIT_INSTRUCTION

if int32(calllib('dynamixel','dxl_get_rpacket_error', ERRBIT_VOLTAGE...
    ))==1
    disp('Input Voltage Error!');
elseif int32(calllib('dynamixel','dxl_get_rpacket_error',...
    ERRBIT_ANGLE))==1
    disp('Angle limit error!');
elseif int32(calllib('dynamixel','dxl_get_rpacket_error',...
    ERRBIT_OVERHEAT))==1
    disp('Overheat error!');
elseif int32(calllib('dynamixel','dxl_get_rpacket_error',...
    ERRBIT_RANGE))==1
    disp('Out of range error!');
elseif int32(calllib('dynamixel','dxl_get_rpacket_error',...
    ERRBIT_CHECKSUM))==1
    disp('Checksum error!');
elseif int32(calllib('dynamixel','dxl_get_rpacket_error',...
    ERRBIT_OVERLOAD))==1
    disp('Overload error!');
elseif int32(calllib('dynamixel','dxl_get_rpacket_error',...
    ERRBIT_INSTRUCTION))==1
    disp('Instruction code error!');
end
end

% Verificar o estado dos atuadores
function [] = PrintCommStatus(CommStatus)

global COMM_TXFAIL
global COMM_RXFAIL
global COMM_TXERROR

```

```

global COMM_RXWAITING
global COMM_RXTIMEOUT
global COMM_RXCORRUPT

switch(CommStatus)
    case COMM_TXFAIL
        disp('COMM_TXFAIL : Failed transmit instruction packet!');
    case COMM_TXERROR
        disp('COMM_TXERROR: Incorrect instruction packet!');
    case COMM_RXFAIL
        disp('COMM_RXFAIL: Failed get status packet from device!');
    case COMM_RXWAITING
        disp('COMM_RXWAITING: Now recieving status packet!');
    case COMM_RXTIMEOUT
        disp('COMM_RXTIMEOUT: There is no status packet!');
    case COMM_RXCORRUPT
        disp('COMM_RXCORRUPT: Incorrect status packet!');
    otherwise
        disp('This is unknown error code!');
end
end

```

Sinergia III:

```

function [] = Sinergia_3 ()

%iniciar_global();
global ERBIT_VOLTAGE
ERBIT_VOLTAGE      = 1;
global ERBIT_ANGLE
ERBIT_ANGLE        = 2;
global ERBIT_OVERHEAT
ERBIT_OVERHEAT     = 4;
global ERBIT_RANGE
ERBIT_RANGE        = 8;

```

```
global ERRBIT_CHECKSUM
ERRBIT_CHECKSUM    = 16;

global ERRBIT_OVERLOAD
ERRBIT_OVERLOAD    = 32;

global ERRBIT_INSTRUCTION
ERRBIT_INSTRUCTION = 64;
```

```
global COMM_TXSUCCESS
COMM_TXSUCCESS     = 0;
```

```
global COMM_RXSUCCESS
COMM_RXSUCCESS     = 1;
```

```
global COMM_TXFAIL
COMM_TXFAIL        = 2;
```

```
global COMM_RXFAIL
COMM_RXFAIL        = 3;
```

```
global COMM_TXERROR
COMM_TXERROR       = 4;
```

```
global COMM_RXWAITING
COMM_RXWAITING     = 5;
```

```
global COMM_RXTIMEOUT
COMM_RXTIMEOUT     = 6;
```

```
global COMM_RXCORRUPT
COMM_RXCORRUPT     = 7;
```

```
%libreria que permite controlar o Dynamixel
```

```
loadlibrary('dynamixel', 'dynamixel.h');
```

```
libfunctions('dynamixel');
```

```
DEFAULT_PORTNUM = 5;  %COM5
```

```
DEFAULT_BAUDNUM = 1;  %1Mbps
```

```
%3a sinergia
```

```
%iniciar variaveis
```

```
P_GOAL_POSITION = 30;
```

```
P_PRESENT_POSITION = 36;
```

```

P_Moving = 46;
P_GOAL_SPEED = 32;

%Defenir ID's para cada actuador
id_9 = 9; % id do actuador q controla os 3 dedos
id_4 = 4; %actuador q controa o dedo indicador
id_13 = 13; %actuador do polegar

%posicoes do software Roboplus
int32 GoalPos9;
int32 GoalPos4;
int32 GoalPos13;
GoalPos9 = [80 600];
GoalPos4 = [400 1000];
GoalPos13 = [620 620];
int32 index;

%iniciar
calllib('dynamixel', 'dxl_initialize', DEFAULT_PORTNUM, ...
        DEFAULT_BAUDNUM);

while (1)
    reply = input('Iniciar a Sinergia III [S/N]: ', 's');
    if isempty(reply)
        continue;
    elseif (reply == 'S' || reply == 's')
        reply2 = input('Abrir [A], Fechar [F]: ', 's');
        %definir velocidade
        calllib('dynamixel', 'dxl_write_word', id_13, P_GOAL_SPEED, 100);
        calllib('dynamixel', 'dxl_write_word', id_9, P_GOAL_SPEED, 100);
        calllib('dynamixel', 'dxl_write_word', id_4, P_GOAL_SPEED, 100);
        if (reply2 == 'A' || reply2 == 'a')
            index = 1;
            %definir posicao

```

```

    calllib('dynamixel', 'dxl_write_word', id_4, P_GOAL_POSITION...
        ,GoalPos4(index));
    calllib('dynamixel', 'dxl_write_word', id_9, P_GOAL_POSITION...
        ,GoalPos9(index));
    calllib('dynamixel', 'dxl_write_word', id_13, ...
        P_GOAL_POSITION, GoalPos13(index));
end

if (reply2 == 'F' || reply2 == 'f')
    index = 2;
    %definir posicao
    calllib('dynamixel', 'dxl_write_word', id_4, P_GOAL_POSITION...
        ,GoalPos4(index));
    calllib('dynamixel', 'dxl_write_word', id_9, P_GOAL_POSITION...
        ,GoalPos9(index));
    calllib('dynamixel', 'dxl_write_word', id_13, ...
        P_GOAL_POSITION, GoalPos13(index));
end

Moving4 = 1;
Moving9 = 1;
Moving13 = 1;
while Moving4 == 1 && Moving9 == 1 && Moving13 == 1
    %Ler a posicao atual
    PresentPos4 = int32(calllib('dynamixel', 'dxl_read_word', ...
        id_4, P_PRESENT_POSITION));
    PresentPos9 = int32(calllib('dynamixel', 'dxl_read_word', ...
        id_9, P_PRESENT_POSITION));
    PresentPos13 = int32(calllib('dynamixel', 'dxl_read_word', ...
        id_13, P_PRESENT_POSITION));
    CommStatus = int32(calllib('dynamixel', 'dxl_get_result'))...
        ;
    if CommStatus == COMM_RXSUCCESS
        Position4 =[GoalPos4(index) PresentPos4];
        Position9 =[GoalPos9(index) PresentPos9];

```

```

        Position13 =[GoalPos13(index) PresentPos13];
        %disp(Position4);
        %disp(Position9);
        %disp(Position13);
        PrintErrorCode();
    else
        PrintCommStatus(CommStatus);
        break;
    end

    %Verificar se o moviemnto foi realizado
    Moving4 = int32(calllib('dynamixel','dxl_read_byte',id_4,...
        P_Moving));
    Moving9 = int32(calllib('dynamixel','dxl_read_byte',id_9,...
        P_Moving));
    Moving13 = int32(calllib('dynamixel','dxl_read_byte',...
        id_13,P_Moving));
    CommStatus = int32(calllib('dynamixel','dxl_get_result'))...
        ;
    if CommStatus == COMM_RXSUCCESS
        if Moving4 == 0 && Moving9 == 0 && Moving13 == 0
            if index == 1
                index=2;

            else
                index=1;

            end
        end
        PrintErrorCode();
    else
        PrintCommStatus(CommStatus);
        break;
    end

```

```

        end

        elseif(reply == 'n' || reply=='N')
            break;
        end
    end
end

end

% Verificar se o valor das variaveis 'e correto
function [] = PrintErrorCode()
global ERRBIT_VOLTAGE
global ERRBIT_ANGLE
global ERRBIT_OVERHEAT
global ERRBIT_RANGE
global ERRBIT_CHECKSUM
global ERRBIT_OVERLOAD
global ERRBIT_INSTRUCTION

if int32(calllib('dynamixel','dxl_get_rpacket_error', ERRBIT_VOLTAGE...
)) == 1
    disp('Input Voltage Error!');
elseif int32(calllib('dynamixel','dxl_get_rpacket_error',...
ERRBIT_ANGLE)) == 1
    disp('Angle limit error!');
elseif int32(calllib('dynamixel','dxl_get_rpacket_error',...
ERRBIT_OVERHEAT)) == 1
    disp('Overheat error!');
elseif int32(calllib('dynamixel','dxl_get_rpacket_error',...
ERRBIT_RANGE)) == 1
    disp('Out of range error!');
elseif int32(calllib('dynamixel','dxl_get_rpacket_error',...
ERRBIT_CHECKSUM)) == 1
    disp('Checksum error!');

```

```

elseif int32(calllib('dynamixel','dxl_get_rxpaket_error',...
    ERBIT.OVERLOAD))==1
    disp('Overload error!');
elseif int32(calllib('dynamixel','dxl_get_rxpaket_error',...
    ERBIT.INSTRUCTION))==1
    disp('Instruction code error!');
end
end

% Verificar o estado dos atuadores
function [] = PrintCommStatus(CommStatus)

global COMM_TXFAIL
global COMM_RXFAIL
global COMM_TXERROR
global COMM_RXWAITING
global COMM_RXTIMEOUT
global COMM_RXCORRUPT

switch(CommStatus)
    case COMM_TXFAIL
        disp('COMM_TXFAIL : Failed transmit instruction packet!');
    case COMM_TXERROR
        disp('COMM_TXERROR: Incorrect instruction packet!');
    case COMM_RXFAIL
        disp('COMM_RXFAIL: Failed get status packet from device!');
    case COMM_RXWAITING
        disp('COMM_RXWAITING: Now recieving status packet!');
    case COMM_RXTIMEOUT
        disp('COMM_RXTIMEOUT: There is no status packet!');
    case COMM_RXCORRUPT
        disp('COMM_RXCORRUPT: Incorrect status packet!');
    otherwise
        disp('This is unknown error code!');
end

```

end

Sinergia IV:

```
function [] = Sinergia_4 ()

%iniciar_global();
global ERRBIT_VOLTAGE
ERRBIT_VOLTAGE      = 1;
global ERRBIT_ANGLE
ERRBIT_ANGLE        = 2;
global ERRBIT_OVERHEAT
ERRBIT_OVERHEAT     = 4;
global ERRBIT_RANGE
ERRBIT_RANGE        = 8;
global ERRBIT_CHECKSUM
ERRBIT_CHECKSUM     = 16;
global ERRBIT_OVERLOAD
ERRBIT_OVERLOAD     = 32;
global ERRBIT_INSTRUCTION
ERRBIT_INSTRUCTION = 64;

global COMM_TXSUCCESS
COMM_TXSUCCESS      = 0;
global COMM_RXSUCCESS
COMM_RXSUCCESS      = 1;
global COMM_TXFAIL
COMM_TXFAIL         = 2;
global COMM_RXFAIL
COMM_RXFAIL         = 3;
global COMM_TXERROR
COMM_TXERROR        = 4;
global COMM_RXWAITING
COMM_RXWAITING      = 5;
global COMM_RXTIMEOUT
```

```
COMM_RXTIMEOUT      = 6;
global COMM_RXCORRUPT
COMM_RXCORRUPT      = 7;

%libreria que permite controlar o Dynamixel
loadlibrary('dynamixel', 'dynamixel.h');
libfunctions('dynamixel');

DEFAULT_PORTNUM = 5;  %COM5
DEFAULT_BAUDNUM = 1;  %1Mbps

%4a sinergia
%iniciar variaveis
P_GOAL_POSITION = 30;
P_PRESENT_POSITION = 36;
P_Moving = 46;
P_GOAL_SPEED = 32;

%Defenir ID's para cada actuador
id_9 = 9; % id do actuador q controla os 3 dedos
id_4 = 4; %actuador q controa o dedo indicador
id_13 = 13; %actuador do polegar

%posicoes do software Roboplus
int32 GoalPos9;
int32 GoalPos4;
int32 GoalPos13;
GoalPos9 = [80 400];
GoalPos4 = [400 500];
GoalPos13 = [500 1000];
int32 index;

%iniciar
calllib('dynamixel', 'dxl_initialize', DEFAULT_PORTNUM, ...
        DEFAULT_BAUDNUM);
```

```

while (1)
    reply = input('Iniciar a Sinergia IV [S/N]: ', 's');
    if isempty(reply)
        continue;
    elseif (reply == 'S' || reply == 's')
        reply2 = input('Abrir [A], Fechar [F]: ', 's');
        %definir velocidade
        calllib('dynamixel', 'dxl_write_word', id_13, P_GOAL_SPEED, 100);
        calllib('dynamixel', 'dxl_write_word', id_9, P_GOAL_SPEED, 100);
        calllib('dynamixel', 'dxl_write_word', id_4, P_GOAL_SPEED, 100);
        if (reply2 == 'A' || reply2 == 'a')
            index = 1;
            %definir posicao
            calllib('dynamixel', 'dxl_write_word', id_4, P_GOAL_POSITION...
                , GoalPos4(index));
            calllib('dynamixel', 'dxl_write_word', id_9, P_GOAL_POSITION...
                , GoalPos9(index));
            calllib('dynamixel', 'dxl_write_word', id_13, ...
                P_GOAL_POSITION, GoalPos13(index));
        end

        if (reply2 == 'F' || reply2 == 'f')
            index = 2;
            %definir posicao
            calllib('dynamixel', 'dxl_write_word', id_4, P_GOAL_POSITION...
                , GoalPos4(index));
            calllib('dynamixel', 'dxl_write_word', id_9, P_GOAL_POSITION...
                , GoalPos9(index));
            calllib('dynamixel', 'dxl_write_word', id_13, ...
                P_GOAL_POSITION, GoalPos13(index));
        end

        Moving4 = 1;
        Moving9 = 1;

```

```

Moving13 = 1;
while Moving4 == 1 && Moving9 == 1 && Moving13 == 1
    %Ler a posicao atual
    PresentPos4 = int32(calllib('dynamixel','dxl_read_word',...
        id_4,P_PRESENT_POSITION));
    PresentPos9 = int32(calllib('dynamixel','dxl_read_word',...
        id_9,P_PRESENT_POSITION));
    PresentPos13 = int32(calllib('dynamixel','dxl_read_word',...
        id_13,P_PRESENT_POSITION));
    CommStatus = int32(calllib('dynamixel','dxl_get_result'))...
        ;
    if CommStatus == COMM_RXSUCCESS
        Position4 =[GoalPos4(index) PresentPos4];
        Position9 =[GoalPos9(index) PresentPos9];
        Position13 =[GoalPos13(index) PresentPos13];
        %disp(Position4);
        %disp(Position9);
        %disp(Position13);
        PrintErrorCode();
    else
        PrintCommStatus(CommStatus);
        break;
    end

    %Verificar se o moviemnto foi realizado
    Moving4 = int32(calllib('dynamixel','dxl_read_byte',id_4,...
        P_Moving));
    Moving9 = int32(calllib('dynamixel','dxl_read_byte',id_9,...
        P_Moving));
    Moving13 = int32(calllib('dynamixel','dxl_read_byte',...
        id_13,P_Moving));
    CommStatus = int32(calllib('dynamixel','dxl_get_result'))...
        ;
    if CommStatus == COMM_RXSUCCESS
        if Moving4 == 0 && Moving9 == 0 && Moving13 == 0

```

```
        if index == 1
            index=2;

        else
            index=1;

        end

    end

    PrintErrorCode();

else
    PrintCommStatus (CommStatus);
    break;

end

end

elseif(reply =='n' || reply=='N')
    break;
end

end

end

% Verificar se o valor das variaveis 'e correto
function [] = PrintErrorCode()
global ERRBIT_VOLTAGE
global ERRBIT_ANGLE
global ERRBIT_OVERHEAT
global ERRBIT_RANGE
global ERRBIT_CHECKSUM
global ERRBIT_OVERLOAD
global ERRBIT_INSTRUCTION
```

```

if int32(calllib('dynamixel','dxl_get_rpacket_error', ERRBIT_VOLTAGE...
    ))==1
    disp('Input Voltage Error!');
elseif int32(calllib('dynamixel','dxl_get_rpacket_error',...
    ERRBIT_ANGLE))==1
    disp('Angle limit error!');
elseif int32(calllib('dynamixel','dxl_get_rpacket_error',...
    ERRBIT_OVERHEAT))==1
    disp('Overheat error!');
elseif int32(calllib('dynamixel','dxl_get_rpacket_error',...
    ERRBIT_RANGE))==1
    disp('Out of range error!');
elseif int32(calllib('dynamixel','dxl_get_rpacket_error',...
    ERRBIT_CHECKSUM))==1
    disp('Checksum error!');
elseif int32(calllib('dynamixel','dxl_get_rpacket_error',...
    ERRBIT_OVERLOAD))==1
    disp('Overload error!');
elseif int32(calllib('dynamixel','dxl_get_rpacket_error',...
    ERRBIT_INSTRUCTION))==1
    disp('Instruction code error!');
end
end

% Verificar o estado dos atuadores
function [] = PrintCommStatus(CommStatus)

global COMM_TXFAIL
global COMM_RXFAIL
global COMM_TXERROR
global COMM_RXWAITING
global COMM_RXTIMEOUT
global COMM_RXCORRUPT

switch(CommStatus)

```

```

case COMM_TXFAIL
    disp('COMM_TXFAIL : Failed transmit instruction packet!');
case COMM_TXERROR
    disp('COMM_TXERROR: Incorrect instruction packet!');
case COMM_RXFAIL
    disp('COMM_RXFAIL: Failed get status packet from device!');
case COMM_RXWAITING
    disp('COMM_RXWAITING: Now recieving status packet!');
case COMM_RXTIMEOUT
    disp('COMM_RXTIMEOUT: There is no status packet!');
case COMM_RXCORRUPT
    disp('COMM_RXCORRUPT: Incorrect status packet!');
otherwise
    disp('This is unknown error code!');
end
end

```

Sinergia V:

```

function [] = Sinergia_5 ()

%iniciar_global();
global ERRBIT_VOLTAGE
ERRBIT_VOLTAGE = 1;
global ERRBIT_ANGLE
ERRBIT_ANGLE = 2;
global ERRBIT_OVERHEAT
ERRBIT_OVERHEAT = 4;
global ERRBIT_RANGE
ERRBIT_RANGE = 8;
global ERRBIT_CHECKSUM
ERRBIT_CHECKSUM = 16;
global ERRBIT_OVERLOAD
ERRBIT_OVERLOAD = 32;
global ERRBIT_INSTRUCTION

```

```
ERRBIT_INSTRUCTION = 64;
```

```
global COMM_TXSUCCESS
```

```
COMM_TXSUCCESS      = 0;
```

```
global COMM_RXSUCCESS
```

```
COMM_RXSUCCESS      = 1;
```

```
global COMM_TXFAIL
```

```
COMM_TXFAIL         = 2;
```

```
global COMM_RXFAIL
```

```
COMM_RXFAIL         = 3;
```

```
global COMM_TXERROR
```

```
COMM_TXERROR        = 4;
```

```
global COMM_RXWAITING
```

```
COMM_RXWAITING      = 5;
```

```
global COMM_RXTIMEOUT
```

```
COMM_RXTIMEOUT      = 6;
```

```
global COMM_RXCORRUPT
```

```
COMM_RXCORRUPT      = 7;
```

```
%libreria que permite controlar o Dynamixel
```

```
loadlibrary('dynamixel', 'dynamixel.h');
```

```
libfunctions('dynamixel');
```

```
DEFAULT_PORTNUM = 5;  %COM5
```

```
DEFAULT_BAUDNUM = 1;  %1Mbps
```

```
%5a sinergia
```

```
%iniciar variaveis
```

```
P_GOAL_POSITION = 30;
```

```
P_PRESENT_POSITION = 36;
```

```
P_Moving = 46;
```

```
P_GOAL_SPEED      = 32;
```

```
%Defenir ID's para cada actuador
```

```
id_9 = 9; % id do actuador q controla os 3 dedos
```

```

id_4= 4; %actuador q controa o dedo indicador
id_13= 13; %actuador do polegar

%posicoes do software Roboplus
int32 GoalPos9;
int32 GoalPos4;
int32 GoalPos13;
GoalPos9 = [150 350];
GoalPos4 = [400 600];
GoalPos13 = [700 1023];
int32 index;

%iniciar
calllib('dynamixel', 'dxl_initialize', DEFAULT_PORTNUM, ...
        DEFAULT_BAUDNUM);

while (1)
    reply = input('Iniciar a Sinergia V [S/N]: ', 's');
    if isempty(reply)
        continue;
    elseif (reply == 'S' || reply == 's')
        reply2 = input('Abrir [A], Fechar [F]: ', 's');
        %definir velocidade
        calllib('dynamixel', 'dxl_write_word', id_13, P_GOAL_SPEED, 100);
        calllib('dynamixel', 'dxl_write_word', id_9, P_GOAL_SPEED, 100);
        calllib('dynamixel', 'dxl_write_word', id_4, P_GOAL_SPEED, 100);
        if (reply2 == 'A' || reply2 == 'a')
            index = 1;
            %definir posicao
            calllib('dynamixel', 'dxl_write_word', id_4, P_GOAL_POSITION...
                , GoalPos4(index));
            calllib('dynamixel', 'dxl_write_word', id_9, P_GOAL_POSITION...
                , GoalPos9(index));
            calllib('dynamixel', 'dxl_write_word', id_13, ...
                P_GOAL_POSITION, GoalPos13(index));

```

```

end

if (reply2 == 'F' || reply2 == 'f')
    index = 2;
    %definir posicao
    calllib('dynamixel', 'dxl_write_word', id_4, P_GOAL_POSITION...
        ,GoalPos4(index));
    calllib('dynamixel', 'dxl_write_word', id_9, P_GOAL_POSITION...
        ,GoalPos9(index));
    calllib('dynamixel', 'dxl_write_word', id_13, ...
        P_GOAL_POSITION,GoalPos13(index));
end

Moving4 = 1;
Moving9 = 1;
Moving13 = 1;
while Moving4 == 1 && Moving9 == 1 && Moving13 == 1
    %Ler a posicao atual
    PresentPos4 = int32(calllib('dynamixel', 'dxl_read_word', ...
        id_4, P_PRESENT_POSITION));
    PresentPos9 = int32(calllib('dynamixel', 'dxl_read_word', ...
        id_9, P_PRESENT_POSITION));
    PresentPos13 = int32(calllib('dynamixel', 'dxl_read_word', ...
        id_13, P_PRESENT_POSITION));
    CommStatus = int32(calllib('dynamixel', 'dxl_get_result'))...
        ;
    if CommStatus == COMM_RXSUCCESS
        Position4 =[GoalPos4(index) PresentPos4];
        Position9 =[GoalPos9(index) PresentPos9];
        Position13 =[GoalPos13(index) PresentPos13];
        %disp(Position4);
        %disp(Position9);
        %disp(Position13);
        PrintErrorCode();
    else

```

```

        PrintCommStatus (CommStatus);
        break;
    end

    %Verificar se o moviemnto foi realizado
    Moving4 = int32(calllib('dynamixel','dxl_read_byte',id_4,...
        P.Moving));
    Moving9 = int32(calllib('dynamixel','dxl_read_byte',id_9,...
        P.Moving));
    Moving13 = int32(calllib('dynamixel','dxl_read_byte',...
        id_13,P.Moving));
    CommStatus = int32(calllib('dynamixel','dxl_get_result'))...
        ;
    if CommStatus == COMM_RXSUCCESS
        if Moving4 == 0 && Moving9 == 0 && Moving13 == 0
            if index == 1
                index=2;

            else
                index=1;

            end
        end
        PrintErrorCode();
    else
        PrintCommStatus (CommStatus);
        break;

    end
end

elseif(reply =='n' || reply=='N')
    break;
end
end
end

```

```
end

% Verificar se o valor das variaveis 'e correto
function [] = PrintErrorCode()
global ERBBIT_VOLTAGE
global ERBBIT_ANGLE
global ERBBIT_OVERHEAT
global ERBBIT_RANGE
global ERBBIT_CHECKSUM
global ERBBIT_OVERLOAD
global ERBBIT_INSTRUCTION

if int32(calllib('dynamixel','dxl_get_rpacket_error', ERBBIT_VOLTAGE...
    ))==1
    disp('Input Voltage Error!');
elseif int32(calllib('dynamixel','dxl_get_rpacket_error',...
    ERBBIT_ANGLE))==1
    disp('Angle limit error!');
elseif int32(calllib('dynamixel','dxl_get_rpacket_error',...
    ERBBIT_OVERHEAT))==1
    disp('Overheat error!');
elseif int32(calllib('dynamixel','dxl_get_rpacket_error',...
    ERBBIT_RANGE))==1
    disp('Out of range error!');
elseif int32(calllib('dynamixel','dxl_get_rpacket_error',...
    ERBBIT_CHECKSUM))==1
    disp('Checksum error!');
elseif int32(calllib('dynamixel','dxl_get_rpacket_error',...
    ERBBIT_OVERLOAD))==1
    disp('Overload error!');
elseif int32(calllib('dynamixel','dxl_get_rpacket_error',...
    ERBBIT_INSTRUCTION))==1
    disp('Instruction code error!');
```

```
end
end

% Verificar o estado dos atuadores
function [] = PrintCommStatus(CommStatus)

global COMM_TXFAIL
global COMM_RXFAIL
global COMM_TXERROR
global COMM_RXWAITING
global COMM_RXTIMEOUT
global COMM_RXCORRUPT

switch(CommStatus)
    case COMM_TXFAIL
        disp('COMM_TXFAIL : Failed transmit instruction packet!');
    case COMM_TXERROR
        disp('COMM_TXERROR: Incorrect instruction packet!');
    case COMM_RXFAIL
        disp('COMM_RXFAIL: Failed get status packet from device!');
    case COMM_RXWAITING
        disp('COMM_RXWAITING: Now recieving status packet!');
    case COMM_RXTIMEOUT
        disp('COMM_RXTIMEOUT: There is no status packet!');
    case COMM_RXCORRUPT
        disp('COMM_RXCORRUPT: Incorrect status packet!');
    otherwise
        disp('This is unknown error code!');
end
end
```