

DESENVOLVIMENTO DO SISTEMA DE CONTROLO DE UM BRAÇO ROBOTIZADO E RESPECTIVA INTERFACE COM O UTILIZADOR

Fernando Miguel Gomes Pinto



Mestrado em Engenharia Electrotécnica e de Computadores

Área de Especialização de Automação e Sistemas

Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

2012

Este relatório satisfaz os requisitos que constam da Ficha da unidade curricular de Tese/Dissertação, do Mestrado em Engenharia Eletrotécnica e de Computadores na área de especialização de Automação e Sistemas

Candidato: Fernando Miguel Gomes Pinto, N° 1060380, 106038@isep.ipp.pt

Orientação científica: Prof. Dr. Manuel Fernando dos Santos Silva, mss@isep.ipp.pt

Coorientação científica: Prof. Dr. Ramiro de Sousa Barbosa, rsb@isep.ipp.pt



Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

19 de Novembro de 2012

Aos meus pais e irmão

Agradecimentos

Em primeiro lugar gostaria de agradecer a realização deste trabalho ao Eng.º Manuel Silva e ao Eng.º Ramiro Barbosa, pela oportunidade, atenção e dedicação na orientação deste trabalho e principalmente pelo incentivo demonstrado.

Gostaria também de agradecer ao Carlos Paiva e ao Pedro Peixoto pelo apoio e ajuda dedicada à minha pessoa ao longo deste projeto.

Por fim, um enorme agradecimento a toda a minha família, amigos e colegas por todo o apoio e força depositada em mim ao longo da realização deste documento, bem como ao longo do meu percurso escolar.

Resumo

A robótica tem evoluído de forma significativa nos últimos anos e passa a ser indispensável em várias aplicações nas áreas da engenharia, aeronáutica, medicina, entre outras. O estado da arte do presente trabalho está dividido em duas partes, uma que aborda vários aspetos relacionados com a robótica e outra com os aspetos da fundamentação matemática por de trás da robótica, porque para controlar o robô é necessário implementar expressões matemáticas para o poder controlar.

Neste trabalho é apresentado um sistema de controlo do braço robótico MENTOR e o desenvolvimento de uma interface para o utilizador. Para o controlo do braço robótico foi necessário calcular a cinemática direta e inversa, para que se possa obter os ângulos das juntas para uma dada posição ou qual é a posição final do braço robótico para um valor das juntas.

O sistema é bastante flexível e foi desenvolvido para ser utilizado essencialmente para aprendizagem de robótica, podendo no entanto ser utilizado em outras aplicações.

Palavras-Chave

Robôs, Cinemática direta e inversa, Convenção Denavit-Hartenberg, Braço robotizado, MENTOR, Microsoft Visual Studio

Abstract

Robotics has evolved significantly in recent years and is now essential in many applications in the fields of engineering, aeronautics, medicine, etc. The state of the art of the present work is divided into two parts, one that covers several aspects related to robotics and one with the math underneath to the robotics because to control the robot is required to implement mathematical expressions.

This dissertation presents a control system of the robotic arm MENTOR and the development of a user interface. For the control of the robotic arm was necessary to calculate the direct and inverse kinematics, so that one can obtain the angles of the joints to a given position or what is the final position of the robotic arm given the joint values.

The system is very flexible and is designed to be used primarily for learning robotics, however can be used in other applications.

Keywords

Robots, Forward and inverse kinematics, Denavit-Hartenberg convention, Robot arm, MENTOR, Microsoft Visual Studio.

Índice

AGRADECIMENTOS	I
RESUMO	III
ABSTRACT	V
ÍNDICE	VII
ÍNDICE DE FIGURAS	XI
ÍNDICE DE TABELAS	XV
ACRÓNIMOS	XVII
1. INTRODUÇÃO	1
1.1. CONTEXTUALIZAÇÃO	1
1.2. OBJETIVOS	1
1.3. CALENDARIZAÇÃO	2
1.4. ORGANIZAÇÃO DO DOCUMENTO	2
2. ROBÓTICA	5
2.1. HISTÓRIA DA ROBÓTICA	6
2.2. DEFINIÇÃO DE ROBÔ	12
2.3. AS TRÊS LEIS DA ROBÓTICA	12
2.4. CONSTITUIÇÃO DE UM ROBÔ INDUSTRIAL	13
2.4.1. Braço e punho de um braço robótico	13
2.4.2. Tipo de juntas	14
2.4.3. Graus de liberdade	16
2.5. TIPOS DE ROBÔS	17
2.5.1. Cartesianos ou Gantry	17
2.5.2. Articulado Vertical ou Antropomórfico	18
2.5.3. Cilíndricos	18
2.5.4. Esféricos	19
2.5.5. Articulado Horizontal ou SCARA	19
2.5.6. Paralelos	20
2.5.7. Volume de Trabalho	21
2.6. CONTROLO DOS ROBÔS	21
2.7. ATUADORES DE SISTEMAS DE ACIONAMENTO	22

2.7.1.	Acionamento hidráulico.....	22
2.7.2.	Acionamento pneumático	23
2.7.3.	Acionamento elétrico	24
2.7.4.	Comparação dos tipos de atuadores	25
2.8.	SENSORES.....	26
2.8.1.	Sensores de posição	26
2.8.2.	Sensores de velocidade	27
2.9.	RESOLUÇÃO, REPETIBILIDADE E PRECISÃO	28
3.	CINEMÁTICA, DINÂMICA E PLANEAMENTO DE TRAJETÓRIAS	31
3.1.	CINEMÁTICA	32
3.1.1.	Cinemática direta	34
3.1.2.	Cinemática Inversa	35
3.2.	DINÂMICA	37
3.2.1.	Dinâmica Direta.....	38
3.2.2.	Dinâmica Inversa	38
3.3.	FORMALISMO DE DENAVIT-HARTENBERG.....	39
3.4.	PLANEAMENTO DE TRAJETÓRIAS	41
3.4.1.	Interpolação linear	42
3.4.2.	Interpolação cúbica	42
3.4.3.	Interpolação de quinta ordem.....	43
3.5.	MÉTODOS DE CONTROLO DO ROBÔ.....	44
3.6.	CONCLUSÕES.....	44
4.	ARQUITETURA DO SISTEMA	47
4.1.	DEFINIÇÃO DOS REQUISITOS DO SISTEMA	47
4.2.	BRAÇO ROBÓTICO MENTOR	48
4.3.	FRIENDLYARM MICRO2440	53
4.4.	MICROSOFT VISUAL STUDIO	56
4.5.	FONTE DE ALIMENTAÇÃO ATX	56
4.6.	UNIDADE DE PROCESSAMENTO	59
4.6.1.	STM32VLDISCOVERY	60
4.7.	CONTROLADOR DE MOTORES.....	61
4.7.1.	Ponte H	62
4.7.2.	ST VNH5180	64
5.	IMPLEMENTAÇÃO DO SISTEMA	67
5.1.	CÁLCULO NUMÉRICO	67

5.1.1.	Cinemática Direta Segundo Denavit-Hartenberg	72
5.1.2.	Cinemática inversa	75
5.1.3.	Planeamento de trajetórias	85
5.2.	HARDWARE	87
5.2.1.	Placa de <i>driver</i> dos motores	87
5.2.2.	Controlador do robô	89
5.2.3.	<i>Joystick</i> para a consola	91
5.3.	SOFTWARE.....	93
5.3.1.	Microsoft Visual Studio	94
5.3.2.	Controlo do robô	98
5.4.	CONCLUSÕES	102
6.	TESTES E VALIDAÇÃO DO SISTEMA DESENVOLVIDO	103
6.1.	CÁLCULO NUMÉRICO	103
6.1.1.	Cinemática Direta.....	104
6.1.2.	Cinemática inversa	107
6.2.	TRABALHO DESENVOLVIDO	109
6.2.1.	Interface Gráfica - Consola	109
6.3.	CONCLUSÕES	110
7.	CONCLUSÕES.....	111
	REFERÊNCIAS DOCUMENTAIS	115
	APÊNDICE A. RELAÇÕES TRIGONOMÉTRICAS.....	117
	ANEXO A. DOCUMENTAÇÃO DO TRABALHO.....	119
	ANEXO B. CUSTOS DE IMPLEMENTAÇÃO.....	121

Índice de Figuras

Figura 1	Robô da Unimation [2].....	8
Figura 2	Braço de Stanford [3]	8
Figura 3	Braço robótico PUMA [4].....	9
Figura 4	KUKA FAMULUS [5].....	9
Figura 5	Honda P3 [6] (à esquerda) e Sony QRIO [7] (à direita).....	10
Figura 6	Robô da ABB [8] (à esquerda) e robô da KUKA [9] (à direita)	11
Figura 7	Esquema de um manipulador mecânico [13]	13
Figura 8	Elos e juntas de um manipulador	14
Figura 9	Junta prismática ou linear.....	14
Figura 10	Junta rotacional	15
Figura 11	Junta de torção.....	15
Figura 12	Junta de revolução	15
Figura 13	Junta esférica	16
Figura 14	Exemplo de graus de liberdade	17
Figura 15	Robô de configuração cartesiana [15]	17
Figura 16	Robô de configuração articulado vertical [15]	18
Figura 17	Robô de configuração cilíndrica [15]	18
Figura 18	Robô de configuração esférica [15].....	19
Figura 19	Robô de configuração articulado horizontal [15].....	19
Figura 20	Robô paralelo - configuração <i>tripod</i> [8]	20
Figura 21	Robô paralelo – configuração <i>hexapod</i> [16]	20
Figura 22	Robô de acionamento hidráulico [17]	23
Figura 23	Robô de acionamento pneumático [18].....	24
Figura 24	Robô de acionamento elétrico	25
Figura 25	Potenciômetro para indicar posições angulares [19].....	26
Figura 26	Codificador incremental e princípio de funcionamento	27
Figura 27	Codificador absoluto e princípio de funcionamento	27
Figura 28	Tacômetro mecânico	28
Figura 29	Tacômetro ótico.....	28
Figura 30	Ilustração dos conceitos de repetibilidade e precisão.....	29
Figura 31	Relação entre cinemática direta e cinemática inversa	33
Figura 32	Robô SCARA – configuração <i>right elbow</i> e <i>left elbow</i>	34
Figura 33	Representação de um robô com duas juntas rotacionais	35
Figura 34	Representação de um robô com uma junta rotacional e uma junta prismática.....	35

Figura 35	Representação de um robô com duas juntas rotacionais	36
Figura 36	Representação de um robô com uma junta rotacional e uma junta prismática.....	36
Figura 37	Configuração do braço robótico PUMA para uma mesma posição final	37
Figura 38	Braço robótico MENTOR	48
Figura 39	Esquema do braço robótico	49
Figura 40	Estrutura mecânica do eixo 0 do robô	50
Figura 41	Estrutura mecânica do eixo 1 do robô	51
Figura 42	Estrutura mecânica do punho	52
Figura 43	Rotação do punho.....	52
Figura 44	Manipulador do braço robótico MENTOR	53
Figura 45	Unidade de processamento.....	54
Figura 46	Controlador FriendlyArm Micro2440+SDK Board	55
Figura 47	Controlador FriendlyArm Micro2440+SDK Board com display.....	56
Figura 48	Fonte de alimentação ATX.....	57
Figura 49	Placa de desenvolvimento STM32VLDISCOVERY	60
Figura 50	Ponte H.....	62
Figura 51	Ponte H – condução positiva, motor gira no sentido dos ponteiros do relógio	62
Figura 52	Ponte H – condução negativa, motor gira no sentido contrário aos ponteiros do relógio	63
Figura 53	Ponte H – motor travado, à esquerda através da fonte, à direita através da massa...	63
Figura 54	Sistema de coordenadas da origem.....	68
Figura 55	Sistema de coordenadas para a junta 1	69
Figura 56	Sistema de coordenadas para a junta 2	69
Figura 57	Sistema de coordenadas para a junta 3	70
Figura 58	Sistema de coordenadas para a junta 4	70
Figura 59	Sistema de coordenadas para a garra.....	71
Figura 60	Sistema de coordenadas do braço robótico.....	71
Figura 61	Área em que o braço robótico permite a configuração braço direito e braço esquerdo	77
Figura 62	Esquema do braço robótico MENTOR, vista de topo	78
Figura 63	Esquema do braço robótico MENTOR, vista de lado	79
Figura 64	Configuração <i>Upper Elbow</i> à esquerda e <i>Lower Elbow</i> à direita.....	81
Figura 65	Esquemático para o cálculo de θ_2	83
Figura 66	Esquemático para o cálculo de k_1 e k_2	84
Figura 67	<i>Driver</i> da placa de controlo do motor DC	88
Figura 68	Indicador luminoso para o utilizador do sentido da corrente	88
Figura 69	Conectores de alimentação e controlo do motor	89
Figura 70	Módulo STM32VLDISCOVERY	90
Figura 71	Conectores da placa de controlo do robô	91

Figura 72	Microcontrolador da placa do <i>joystick</i>	92
Figura 73	<i>Joystick</i>	92
Figura 74	<i>Switches</i> da placa do <i>joystick</i>	93
Figura 75	Menu da consola.....	95
Figura 76	Opção “Executar programa” da consola	96
Figura 77	Opção “Programação On-Line” da consola	97
Figura 78	Opção “Move braço” da consola.....	97
Figura 79	Opção “Modificar programa” da consola.....	98
Figura 80	Relação entre a consola, controlo e MENTOR	99
Figura 81	Aplicação desenvolvida para o cálculo da cinemática direta e inversa.....	104
Figura 82	Teste da cinemática direta para a primeira iteração	105
Figura 83	Teste da cinemática direta para a segunda iteração.....	105
Figura 84	Teste da cinemática direta para a terceira iteração.....	106
Figura 85	Teste da cinemática direta para a quarta iteração	106
Figura 86	Teste da cinemática direta para a quinta iteração	107
Figura 87	Teste da cinemática inversa para a primeira iteração.....	107
Figura 88	Teste da cinemática inversa para a segunda iteração	108
Figura 89	Teste da cinemática inversa para a terceira iteração	109

Índice de Tabelas

Tabela 1	Calendarização do projeto	2
Tabela 2	Comparação do volume de trabalho para cada configuração	21
Tabela 3	Comparação dos tipos de atuadores	25
Tabela 4	Características dos eixos	48
Tabela 5	Características principais do robô	50
Tabela 6	Conector ATX, cores e significados.....	58
Tabela 7	Ponte H – estado do motor consoante os interruptores	64
Tabela 8	Tabela de verdade do ST VNH5180	65
Tabela 9	Translação e rotação da junta 1, vista pela junta 0	72
Tabela 10	Translação e rotação da junta 2, vista pela junta 1	73
Tabela 11	Translação e rotação da junta 3, vista pela junta 2	73
Tabela 12	Translação e rotação da junta 4, vista pela junta 3	74
Tabela 13	Translação e rotação da junta 4, vista pela junta 5	74
Tabela 14	Orientações e rotações em torno do eixo x e z	75
Tabela 15	Cinemática direta para as juntas 2, 3 e 4 segundo Denavit-Hartenberg	79

Acrónimos

μ C	-	Microcontrolador
μ P	-	Microprocesador
A.C.	-	Antes de Cristo
ABB	-	<i>Asea Brown Boveri</i>
AC	-	<i>Alternated Current</i>
ADC	-	<i>Analogue-to-Digital Converter</i>
ALU	-	<i>Arithmetic Logic Unit</i>
ARM	-	<i>Advanced RISC Machine</i>
AT	-	<i>Advanced Technology</i>
ATX	-	<i>Advanced Technology Extended</i>
BIOS	-	<i>Basic Input/Output System</i>
BJT	-	<i>Bipolar Junction Transistor</i>
CA	-	Corrente Alternada
CC	-	Corrente Contínua
CD	-	<i>Compact Disk</i>
CE	-	<i>Compact Embedded</i>
CISC	-	<i>Complex Instruction Set Computer</i>
CMOS	-	<i>Complementary Metal-Oxide Semiconductor</i>
CPU	-	<i>Central Processing Unit</i>

DC	-	<i>Direct Current</i>
DH	-	Denavit-Hartenberg
DOF	-	<i>Degrees-of-freedom</i>
E/S	-	Entrada/Saída
EEPROM	-	<i>Electrically-Erasable Programmable Read-Only Memory</i>
GUI	-	<i>Graphical User Interface</i>
HTML	-	<i>Hypertext Markup Language</i>
I/O	-	<i>Input/Output</i>
I2C	-	<i>Inter-Integrated Circuit</i>
IBM	-	<i>International Business Machines</i>
IDE	-	<i>Integrated Development Environment</i>
ISA	-	<i>Industry Standard Architecture</i>
ISEP	-	Instituto Superior de Engenharia do Porto
JTAG	-	<i>Joint Test Action Group</i>
LCD	-	<i>Liquid Crystal Display</i>
LED	-	<i>Light Emitting Diode</i>
MCU	-	<i>Microcontroller Unit</i>
MOSFET	-	<i>Metal-Oxide-Semiconductor Field-Effect Transistor</i>
PCB	-	<i>Printed Circuit Board</i>
PCI	-	Placa de Circuito Impresso
PID	-	Proporcional Integral Diferencial

PNP	-	<i>Positive-Negative-Positive</i>
PUMA	-	<i>Programmable Universal Machine for Assembly</i>
PWM	-	<i>Pulse-Width Modulation</i>
RAM	-	<i>Random Access Memory</i>
RISC	-	<i>Reduced Instruction Set Computer</i>
RP	-	Rotacional, Prismático
RR	-	Rotacional, Rotacional
RTC	-	<i>Real Time Clock</i>
SCARA	-	<i>Selectively Compliant Articulated Robot Arm</i>
SD	-	<i>Secure Digital</i>
SDK	-	<i>Software Development Kit</i>
SMD	-	<i>Surface Mount Device</i>
SPI	-	<i>Serial Peripheral Interface</i>
T3	-	<i>Tool of The fuTure</i>
TTL	-	<i>Transistor-Transistor Logic</i>
ULA	-	Unidade Lógica Aritmética
USA	-	<i>United States of America</i>
USB	-	<i>Universal Serial Bus</i>
VDC	-	<i>Voltage Direct Current</i>
VGA	-	<i>Video Graphics Array</i>
XHTML	-	<i>Extensible Hypertext Markup Language</i>

XML - *Extensible Markup Language*

XSLT - *Extensible Stylesheet Language*

1. INTRODUÇÃO

O presente documento pretende abordar todo o trabalho desenvolvido, bem como apresentar os aspetos importantes para a realização do mesmo, salientando principalmente a robótica. Também pretende apresentar a matemática envolvida no controlo de braços robóticos.

1.1. CONTEXTUALIZAÇÃO

Este projeto surgiu do desejo de realizar o controlo e o desenvolvimento de uma interface para o utilizador de um braço robótico existente no Laboratório de Controlo. O braço robótico existente carecia de uma atualização nas placas de controlo do mesmo, para tal, surgiu o tema “Desenvolvimento do Sistema de Controlo de um Braço Robotizado e Respetiva Interface com o Utilizador”. O controlo do braço robótico MENTOR era muito rudimentar bem como a interface para o utilizar, que não era nada amigável.

1.2. OBJETIVOS

Para a realização do trabalho este foi dividido em objetivos mais pequenos de forma a facilitar a sua prossecução, nomeadamente:

- Desenvolvimento do controlo;
- Possibilidade de efetuar movimentos lineares e de juntas;
- Interpolação de movimentos;
- Desenvolvimento de uma interface para o utilizador;

- Desenvolvimento de uma consola para controlo do braço.

1.3. CALENDARIZAÇÃO

Neste tópico é apresentada a calendarização das tarefas desenvolvidas para a realização deste trabalho. A Tabela 1 mostra a calendarização gráfica do projeto.

Tabela 1 Calendarização do projeto

ID	Nome das etapas	Início	Fim	Duração	2011				2012											
					Set	Out	Nov	Dez	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	
1	Período de aulas do 1º semestre	15-09-2011	28-03-2012	28s																
2	Pesquisa do estado da arte	15-02-2012	30-03-2012	6s 9h																
3	Escrita do estado da arte	15-03-2012	05-04-2012	3s 3h																
4	Pesquisa dos conceitos matemáticos ligados à robótica	02-04-2012	30-04-2012	4s 3h																
5	Escrita do capítulo Cinemática, Dinâmica e Planeamento de trajetórias	20-04-2012	15-05-2012	3s 9h																
6	Arquitetura do sistema a desenvolver	15-05-2012	31-05-2012	2s 9h																
7	Escrita do capítulo Arquitetura do sistema	21-05-2012	29-06-2012	6s																
8	Sinistro e recuperação	02-07-2012	15-08-2012	6s 9h																
9	Implementação do trabalho - Cinemática e Planeamento de trajetórias	16-08-2012	14-09-2012	4s 6h																
10	Implementação do trabalho – Hardware eSoftware	03-09-2012	05-10-2012	5s																
11	Testes e ajustes da implementação	01-10-2012	30-10-2012	4s 6h																
12	Elaboração dos restantes capítulos da tese	16-08-2012	05-11-2012	11s 9h																

1.4. ORGANIZAÇÃO DO DOCUMENTO

Este documento encontra-se estruturado nos seguintes capítulos:

- Robótica – Neste capítulo será feita uma introdução, explicação do que é a robótica, tipo de robôs, controlo e configuração;
- Cinemática, Dinâmica e Planeamento de Trajetórias – Este capítulo destina-se à análise dos fundamentos matemáticas que são necessários para efetuar o controlo do movimento de um robô, bem como o planeamento de trajetórias;
- Arquitetura do Sistema – Aqui serão explicadas todas as tecnologias usadas para a realização deste projeto;
- Implementação – Capítulo que diz respeito ao desenvolvimento do trabalho realizado, onde é explicado todo o processo da realização da interface gráfica e do controlo do robô;
- Testes e Validação do Trabalho Desenvolvido – Neste capítulo são explicados e demonstrados os resultados obtidos na realização do projeto;

- Conclusões - Por fim serão tiradas algumas conclusões de todo o trabalho realizado e serão apontadas algumas possibilidades para desenvolvimentos futuros.

2. ROBÓTICA

Desde os primórdios da sua origem, o ser humano sempre utilizou ferramentas e utensílios que o auxiliaram na realização de diversas atividades quotidianas relacionadas às suas necessidades de sobrevivência.

Para a civilização ocidental o conceito de evolução humana está diretamente associado ao grau de desenvolvimento tecnológico adquirido ao longo do tempo, através do aperfeiçoamento destes objetos. Portanto, a motivação de se criar máquinas que possam substituir o homem na realização de tarefas é uma característica da própria cultura ocidental. A primeira referência explícita a este conceito foi escrita por Aristóteles no século IV A.C.: “se os instrumentos pudessem realizar suas próprias tarefas, obedecendo ou antecipando o desejo de pessoas...”. Ao longo dos séculos, diversas invenções propiciaram a necessária bagagem tecnológica para a gradual substituição do homem pela máquina. Porém, somente quando ocorre de forma sistemática a aplicação da ciência à indústria há uma concreta alteração do cenário, resultando na sociedade industrial. Já no fim do século XVI, Francis Bacon preconizava a ideia “...de que o saber devesse produzir seus frutos na prática, de que a ciência devesse ser aplicável à indústria, de que os homens tivessem o dever sagrado de se organizarem para melhorar e transformar as condições de vida”.

A partir da máquina a vapor desenvolvida por James Watt em 1769, houve um acentuado progresso em termos de automação de processos produtivos. A produção industrial em larga escala e os meios de transporte revolucionaram social e economicamente as relações humanas. Até meados do século XX o processo de produção foi baseado no emprego de máquinas projetadas especificamente para a fabricação em série de produtos de uma mesma característica, visando uma elevada produtividade, volume e qualidade. Este modelo é denominado automação rígida e foi bastante difundido pelo empresário Henry Ford no início do século passado.

O avanço tecnológico das últimas décadas teve reflexo direto na organização das indústrias, as quais buscam minimizar os seus custos industriais através da adoção de diversos modelos de produção. Neste contexto, destacam-se a automação programável, relativa à fabricação em série de pequenos e médios lotes de produtos, e a automação flexível, referida à fabricação de lotes variáveis de produtos diversos.

2.1. HISTÓRIA DA ROBÓTICA

O conceito de robô vem já do início da história, quando mitos faziam referência a mecanismos que ganhavam vida. Na civilização grega, os primeiros modelos de “robôs” encontrados foram figuras de aparência humana e/ou animal, que usavam sistemas de pesos e bombas hidráulicas.

No entanto, para se fazer referência ao aparecimento e crescimento da robótica é quase obrigatório falar de Leonardo Da Vinci. Muito à frente do seu tempo, Da Vinci pensou num robô de forma diferente, estudando primeiro a anatomia humana e animal tentando perceber e arranjar soluções para a implementação de um robô que produzia movimentos e funções humanas (robô antropomórfico). O livro Codex Atlanticus fala dessa ideia, tendo textos, gráficos e esquemas sobre a implementação de um robô.

Passando para a época mais atual, o termo robô vem do checo “*robota*” que significa trabalho forçado e foi usado pela primeira vez em 1921 por Karel Capek na sua peça de teatro “Rossum’s Universal Robots”. Na sua visão, os robôs eram máquinas de trabalho incansáveis, de aspeto humano com capacidades avançadas mesmo para os robôs da atualidade.

Isaac Asimov foi quem inventou a palavra robótica, para denominar a ciência que lida com os robôs, em 1942, numa pequena história intitulada “Runaround”. Asimov propôs a existência de leis aplicáveis à robótica, que serão faladas mais à frente.

O Homem sempre fantasiou com robôs, e essa fantasia está bem assente na ficção científica através de filmes, livros, banda desenhada e até desenhos animados. Exemplos mais conhecidos são o C3PO, o R2-D2, da saga Star Wars, ou até mesmo a trilogia Terminator.

Os primeiros trabalhos em robótica de manipulação foram concebidos alguns anos depois do fim da 2ª guerra mundial. As primeiras máquinas eram do tipo *Master-Slave* e foram introduzidas para manipular materiais perigosos. Entre eles, estão o “gantry-robô” desenvolvido pela General Mills Corporation (USA, 1950), o “Planetbot” (1957) sendo o primeiro robô comercial com coordenadas polares e o robô desenvolvido por Normam Diedrich na Universidade Case Western Reserve (Cleveland, USA), que foi o primeiro manipulador elétrico com juntas de revolução.

Já os primeiros passos da robótica industrial começam a ser dados por volta de 1954, quando George Devol efectuou as primeiras patentes sobre robótica. A primeira empresa a produzir um robô industrial foi a Unimation, fundada por George Devol e Joseph F. Engelberger em 1956, sendo baseada nas patentes originais de Devol.

Vários modelos surgiram com estas novas tecnologias, como o T3 (*Tool of The fuTure*) desenvolvido pela Cincinnati Millacron (1974), o braço de Stanford (fim dos anos 60) que deu origem ao *Programmable Universal Machine for Assembly* (PUMA) da Unimation Inc. (1978). O manipulador da IBM (1975) e o SCARA (*Selective Compliant Articulated Robot Arm*) (1978-79) são também exemplos de robôs que surgiram nesta época.

Os robôs da Unimation (Figura 1) eram apelidados de “máquinas de transferência programadas”, visto que a sua principal função era transferir objetos de um ponto para outro. Para isso os robôs utilizavam atuadores hidráulicos e eram programados através da aprendizagem de um “conjuntos de coordenadas”, ou ângulos de todas as juntas, que durante o processo normal de trabalho eram executados ciclicamente [1].

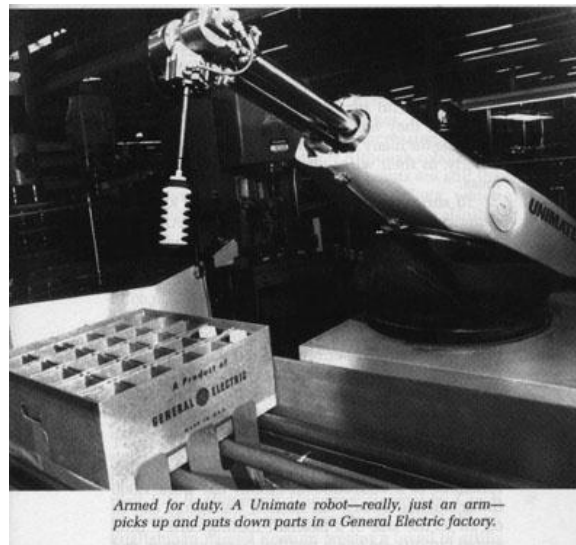


Figura 1 Robô da Unimation [2]

Em 1969 Victor Scheinman desenvolveu o “braço de Stanford” (Figura 2), na Universidade de Stanford, um robô articulado de 6 eixos, totalmente elétrico, projetado para permitir uma solução utilizando a anatomia de um braço. Isto permitiu que o robô fosse capaz de seguir com precisão caminhos arbitrários no espaço e aumentou as possibilidades de utilizar robôs em aplicações mais sofisticadas tais como montagem e soldadura.



Figura 2 Braço de Standford [3]

Scheinman vendeu os seus projetos à Unimation que, em parceria com a General Motors, o comercializou posteriormente como a Máquina Programável Universal para Montagem – PUMA como se pode ver na Figura 3.

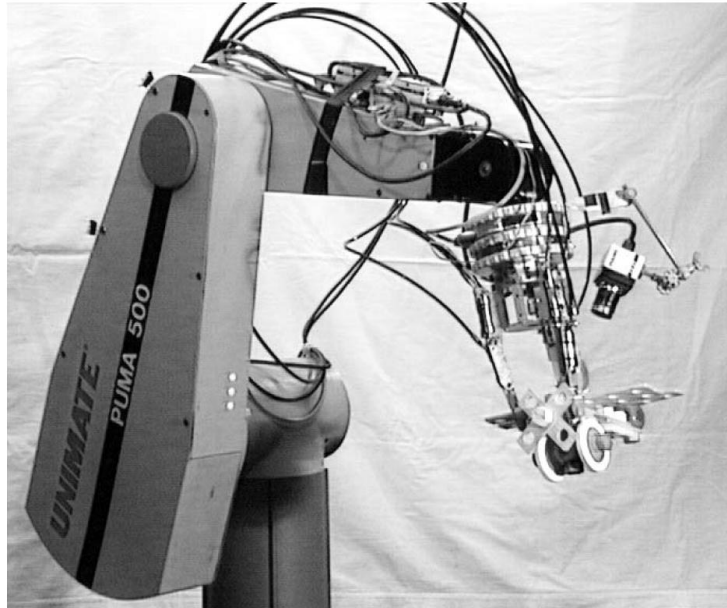


Figura 3 Braço robótico PUMA [4]

Em 1973 a KUKA construiu o seu primeiro robô industrial, conhecido como FAMULUS (Figura 4), sendo este o primeiro robô industrial articulado a possuir seis eixos controlados eletronicamente.

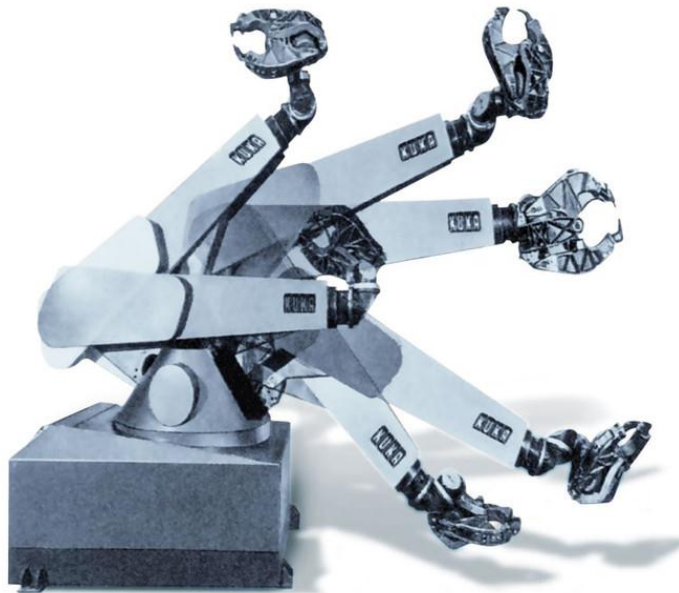


Figura 4 KUKA FAMULUS [5]

Mais tarde, no ano de 1979, começa a ser comercializado pela Sankyo (Japão) e pela IBM, o robô *Selective Compliant Articulated Robot Arm (SCARA)*, sendo bastante utilizado nas indústrias de montagens de pequenas peças, tais como:

- Componentes eletrônicos;
- Computadores;
- Placas de circuito impresso.

Em 1983 surge a primeira linha flexível de montagem automatizada com o uso de robôs. Desde então o desenvolvimento dos robôs foi crescendo, e a grande fantasia do Homem foi sendo possível de criar – o robô antropomórfico ou humanoide. O primeiro robô deste tipo com sucesso foi desenvolvido pela Honda (Figura 5, à esquerda), em 1997, e conseguia subir escadas. Em 2001 a Honda anuncia novos desenvolvimentos, como a capacidade de locomoção, visão, olfato, tacto e voz. Em 2003, foi a vez de a Sony apresentar o seu robô humanoide (Figura 5, à direita), que tinha como principal desenvolvimento a possibilidade de correr.

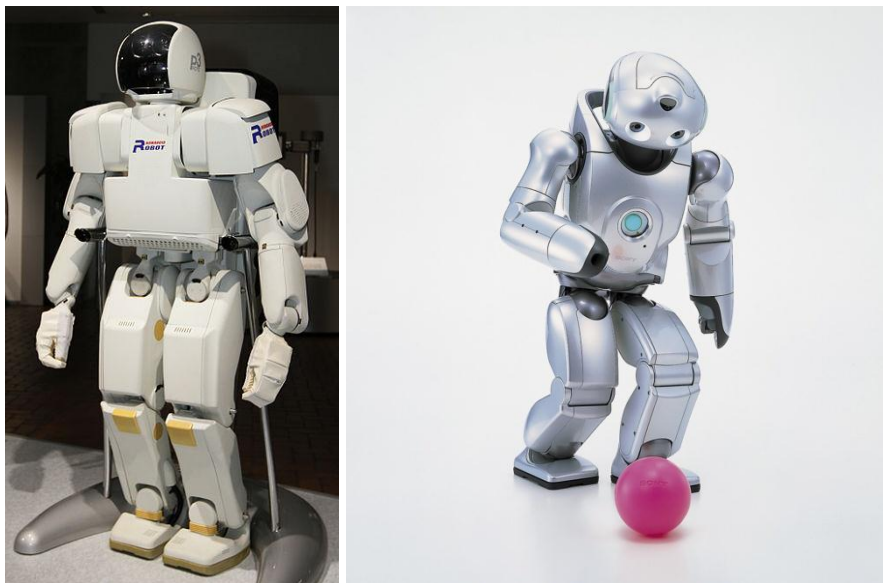


Figura 5 Honda P3 [6] (à esquerda) e Sony QRIO [7] (à direita)

O conseqüente desenvolvimento da eletrônica e da informática leva hoje a robótica para novas áreas, como o setor doméstico, a exploração de locais de difícil acesso para o Homem e os brinquedos robotizados.

Desde o simples robô-aspirador até à sonda Mars Pathfinder, a robótica cada vez está mais presente no dia-a-dia, sendo o aspecto humanoide cada vez mais aperfeiçoado nos robôs.

Em relação à atualidade da robótica industrial, as principais marcas de robôs e manipuladores são, entre outras:

- ABB (ASEA Brown-Boveri);
- Adept Technology;
- Fanuc;
- Genmark;
- Honda;
- Innovative Robotics;
- Kawasaki;
- KUKA Robotic;
- Motoman;
- Staubli.

Os robôs evoluíram muito ao longo das últimas décadas, tendo sido desenvolvidas várias técnicas de controlo. No entanto, ainda há muito trabalho nesta área, nomeadamente a programação destes, o controlo de força, a retroação visual, a integração sensorial, passando também por novas estruturas mecânicas e novos materiais de forma a criar robôs mais leves, flexíveis e baratos. Na Figura 6 podem-se observar exemplos deste tipo de robôs.

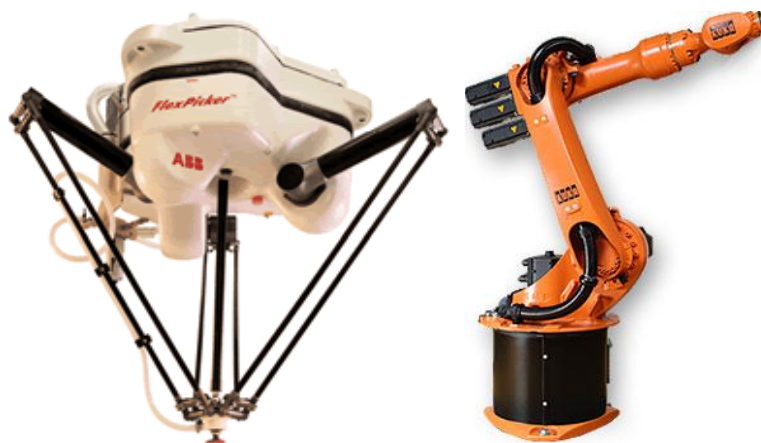


Figura 6 Robô da ABB [8] (à esquerda) e robô da KUKA [9] (à direita)

2.2. DEFINIÇÃO DE ROBÔ

A origem do termo “Robot”, em português o termo alternativo é robô, provém do termo eslavo¹ *Robota*, que significa trabalho forçado ou escravo e teve a sua introdução numa peça de 1921 de Karel Capek.

No dicionário português da Porto Editora, robô é um mecanismo automático, por vezes com a configuração de um ser humano, capaz de fazer movimentos e executar certos trabalhos em substituição do Homem [10].

2.3. AS TRÊS LEIS DA ROBÓTICA

Em 1942, o escritor e cientista russo Isaac Asimov, numa história sua (“Runaround”), enunciou 3 leis para a robótica. São elas:

- Lei 1: Um robô não pode ferir um ser humano ou, por omissão, permitir que este sofra algum mal.
- Lei 2: Um robô deve obedecer às ordens que lhe sejam dadas por seres humanos, exceto nos casos em que tais ordens contrariem a Primeira Lei.
- Lei 3: Um robô deve proteger sua própria existência desde que tal proteção não entre em conflito com a Primeira e Segunda Leis.

Isaac Asimov concebeu estas leis de forma a ser possível a existência de robôs inteligentes, ou seja, que conseguissem fazer a distinção entre o bem e o mal, em convívio com os seres Humanos, sem que os robôs se revoltassem contra estes.

Mais tarde surgiu uma quarta lei:

- “Lei Zero”: Um robô não pode fazer mal à humanidade e nem, por inação, permitir que ela sofra algum mal.

Desta forma, o bem da humanidade encontra-se em primeiro lugar, sendo esta mais importante do que um indivíduo isolado [11].

¹ Relativo aos Eslavos, povos que se estabeleceram no centro, leste e sudeste da Europa

2.4. CONSTITUIÇÃO DE UM ROBÔ INDUSTRIAL

O sistema robótico mais adotado na indústria é um manipulador mecânico, que pretende assemelhar-se ao braço humano. Em analogia com o corpo humano, é constituído por base, corpo, braço e punho, como mostra a Figura 7 [12].

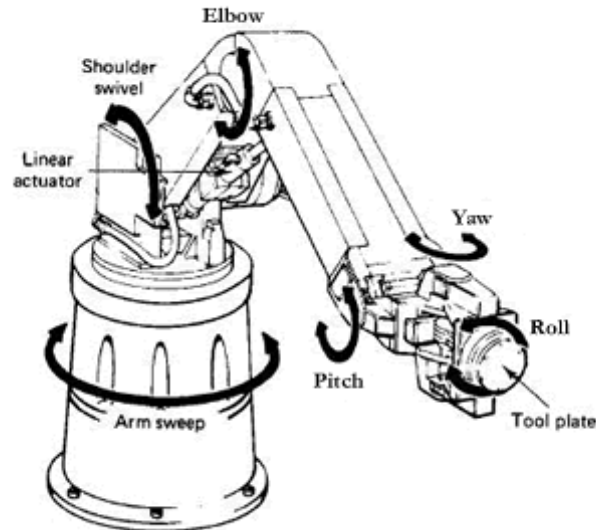


Figura 7 Esquema de um manipulador mecânico [13]

2.4.1. BRAÇO E PUNHO DE UM BRAÇO ROBÓTICO

O braço (*arm*) é a parte do manipulador que normalmente está associada ao posicionamento (x,y,z) no espaço físico cartesiano ou operacional. Já o punho (*wrist*) tem como função orientar a garra, pinça ou outros tipos de atuadores finais para executar uma determinada função. Estes componentes são constituídos por partes rígidas, os elos (*links*), sendo ligadas entre si por juntas (*joints*), como mostra a Figura 8 [12].

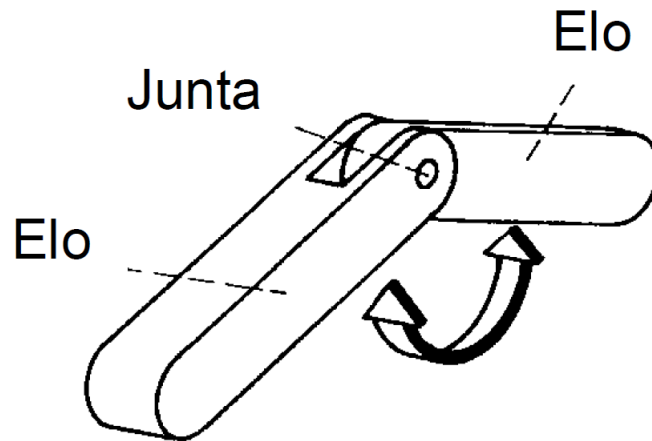


Figura 8 Elos e juntas de um manipulador

2.4.2. TIPO DE JUNTAS

Junta é o mecanismo que faz a união entre dois elos; estas podem ser:

- Prismática ou Linear (P) – São juntas onde os elos se movem linearmente, como demonstra a Figura 9.

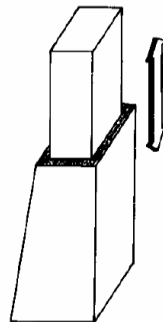


Figura 9 Junta prismática ou linear

- Rotacionais (R) – As juntas rotacionais são caracterizadas por o elo de entrada e o elo de saída estarem ligados por uma junta (dobradiça), e com um elo podendo se mover num movimento cadenciado em relação ao outro elo, como se mostra na Figura 10.

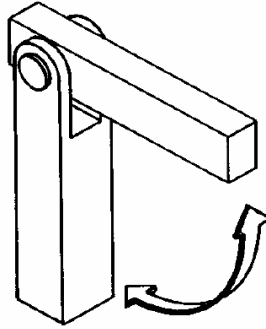


Figura 10 Junta rotacional

- Torção (T) – São juntas onde o elo de entrada e o elo de saída são paralelos e a rotação da junta é perpendicular aos dois eixos, como mostra a Figura 11.

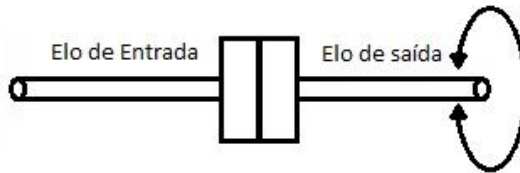


Figura 11 Junta de torção

- Revolução (V) – A junta de revolução é caracterizada pelo elo de entrada ser perpendicular ao elo de saída e a rotação da junta ser perpendicular ao elo de entrada e paralela ao elo de saída, como ilustra a Figura 12.

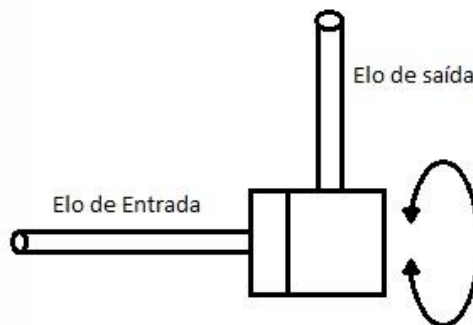


Figura 12 Junta de revolução

- Esféricas (S) – As juntas esféricas são juntas que apresentam três graus de liberdade, nomeadamente, mover nos eixos x e y e em torno do próprio elo. A Figura 13 é um exemplo de uma junta esférica.

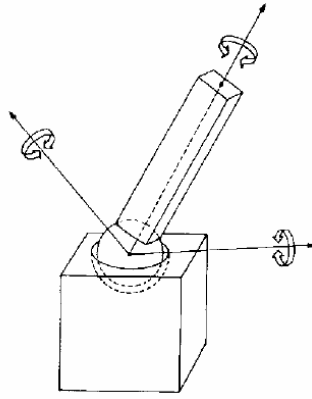


Figura 13 Junta esférica

2.4.3. GRAUS DE LIBERDADE

Os graus de liberdade (*degrees-of-freedom* - DOF) estão associados ao número total de movimentos independentes que um objeto pode efetuar. Por exemplo, se se pegar num cubo, e se se considerar um sistema de 3 eixos (x,y,z), pode-se rodá-lo em torno de cada um dos eixos e ainda movê-lo ao longo desses mesmos 3 eixos, tendo assim 6 graus de liberdade.

Um conceito diferente é o de graus de mobilidade, associado ao número de juntas existentes. Um exemplo comum desta diferença são os tripés: na verdade em cada pé existem várias juntas prismáticas que afetam o mesmo movimento, isto é, ao longo daquele eixo em particular. Se em cada pé houver 3 juntas, tem-se um tripé com 3 graus de liberdade mas 9 graus de mobilidade.

A Figura 14 representa um exemplo gráfico do que são graus de liberdade, onde o objetivo é colocar a peça no encaixe. Para efetuar o movimento para a posição desejada (x,y,z) são precisos três graus de liberdade, mais três para efetuar a orientação do encaixe (o manipulador da presente figura não permite efetuar a orientação), o que perfaz um total de seis graus de liberdade [12].

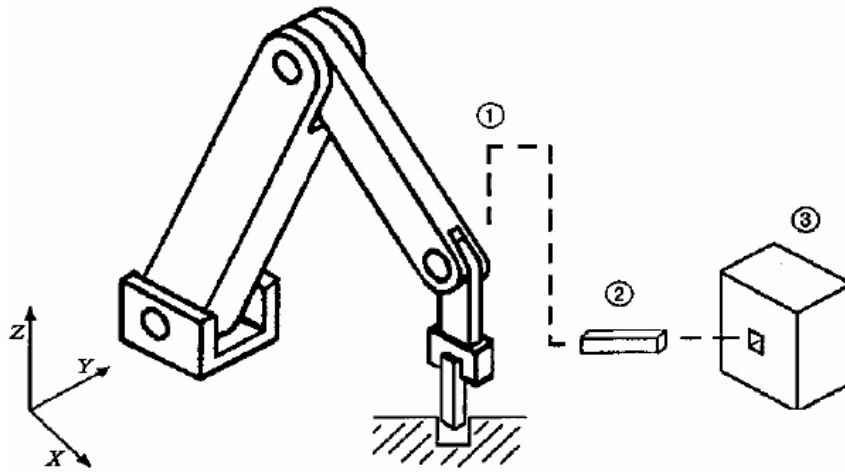


Figura 14 Exemplo de graus de liberdade

2.5. TIPOS DE ROBÔS

Atualmente existem no mercado vários tipos de robôs, sendo que estes diferem entre si, entre outras características, na área de trabalho e configuração. Na indústria uma operação específica precisa de um tipo de mecanismo específico para a elaboração operação, de maneira que precisa de uma configuração do robô específica, e como tal optou-se por desenvolver configurações de robôs que conseguissem cobrir as exigências que cada tarefa precisa.

2.5.1. CARTESIANOS OU GANTRY

Os robôs de configuração cartesiana ou de Gantry são compostos por três juntas prismáticas e com uma área de trabalho semelhante a um paralelepípedo ou cubo, como mostra a Figura 15.

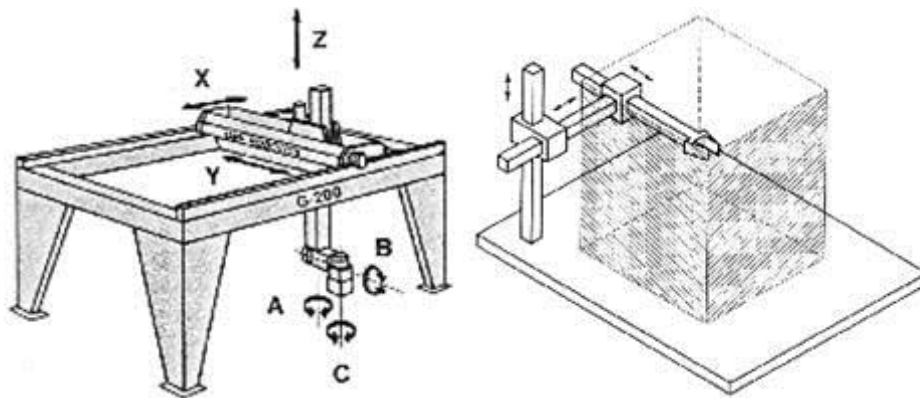


Figura 15 Robô de configuração cartesiana [15]

2.5.2. ARTICULADO VERTICAL OU ANTROPOMÓRFICO

A configuração articulada vertical é a configuração de robôs mais usada na indústria. Estes robôs normalmente são constituídos por três ou mais juntas rotativas, resultando uma área de trabalho normalmente parecida com uma esfera. Na Figura 16 pode-se observar esse tipo de robô. De todos as configurações está é aquela que mais se assemelha a um braço humano.

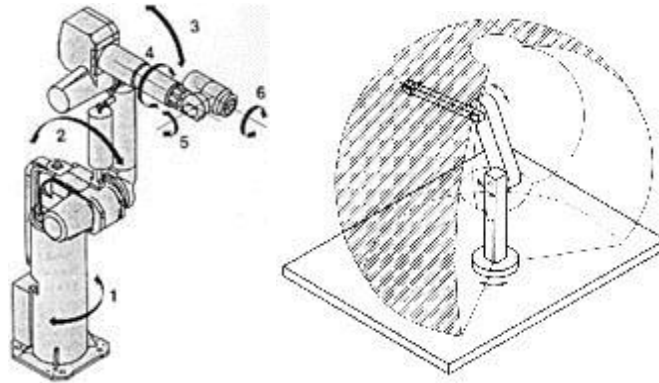


Figura 16 Robô de configuração articulada vertical [15]

2.5.3. CILÍNDRICOS

Os robôs de configuração cilíndrica são caracterizados por possuírem duas juntas prismáticas e uma junta rotacional. O espaço de trabalho deste robô assemelha-se a um cilindro, como se pode ver na Figura 17, e daí provém o nome desta configuração.

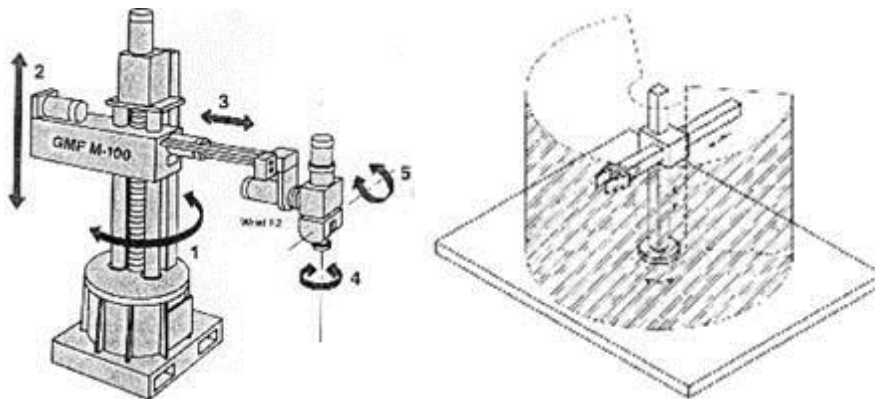


Figura 17 Robô de configuração cilíndrica [15]

2.5.4. ESFÉRICOS

Ao contrário da configuração articulada que só possui juntas rotacionais, este robô possui juntas rotacionais e prismáticas, sendo normalmente composto por duas juntas rotacionais e uma prismática. O espaço de trabalho deste robô assemelha-se a uma esfera. Na Figura 18 pode-se observar um exemplo de robô com configuração esférica.

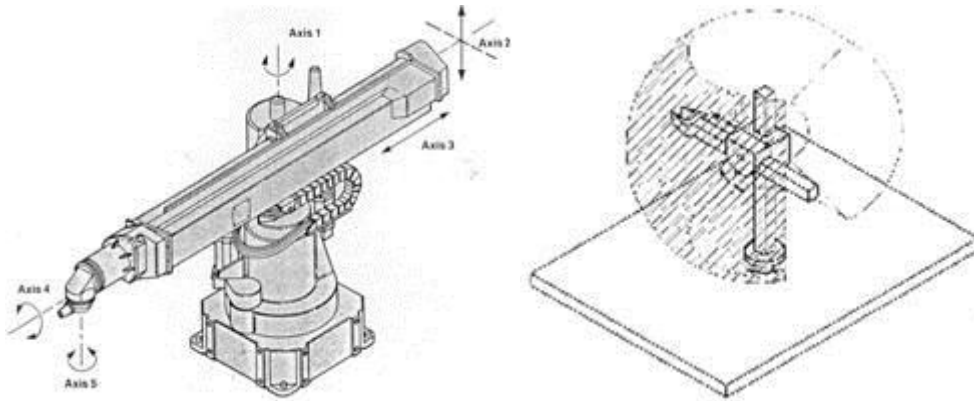


Figura 18 Robô de configuração esférica [15]

2.5.5. ARTICULADO HORIZONTAL OU SCARA

Os robôs *Selectively Compliant Articulated Robot Arm* (SCARA) são normalmente utilizados em aplicações de montagem, isto porque a sua configuração permite que se atinja elevadas velocidades para trabalhos leves. A configuração deste robô pode ser observada na Figura 19.

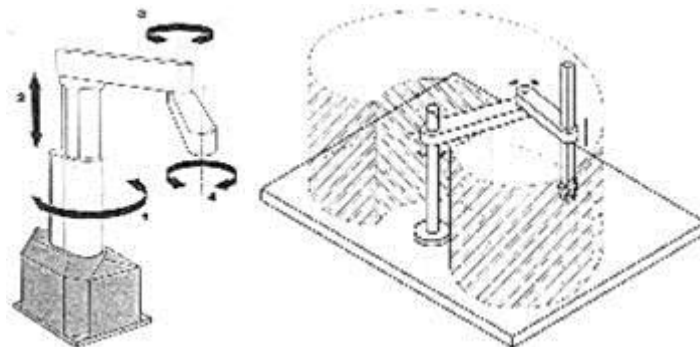


Figura 19 Robô de configuração articulado horizontal [15]

2.5.6. PARALELOS

De todas as configurações esta é a mais distinta, derivado à sua configuração. Estes robôs adotam geralmente duas configurações como se lista a seguir.

- *Tripod* – A configuração *tripod* é constituído por três braços que interligam a base, placa e o *end-effector* (dispositivo da extremidade). São usados em aplicações de *pick and place* devido à sua configuração permitir grandes velocidades e o seu volume de trabalho assemelhar-se a um meia-esfera. A Figura 20 mostra um exemplar deste tipo de configuração.



Figura 20 Robô paralelo - configuração *tripod* [8]

- *Hexapod* – A configuração *hexapod* ou *Stewart Platform* é constituída por seis braços, usa-se em aplicações em que é necessário efetuar orientações em *roll*, *pitch* e *yaw*, como por exemplo, simulação de voo e posicionamento de antenas. Pode-se ver este tipo de configuração na Figura 21.



Figura 21 Robô paralelo – configuração *hexapod* [16]

2.5.7. VOLUME DE TRABALHO

Na Tabela 2 tem-se uma comparação do volume de trabalho para cada tipo de configuração que um robô pode ter, considerando a situação ideal, onde cada articulação R é capaz de rodar 360°, cada articulação P pode realizar uma translação de L (igual ao comprimento de cada elo) e cada configuração tem três juntas.

Tabela 2 Comparação do volume de trabalho para cada configuração

Configuração	Volume de trabalho
Cartesiana	L^3
Cilíndrica	$3\pi L^3$
Esférica	$\frac{28}{3}\pi L^3$
Articulado horizontal	$4\pi L^3$
Articulado Vertical	$\frac{32}{3}\pi L^3$

Como se mostra, a configuração com maior volume de trabalho é a configuração articulada vertical, sendo que esta se assemelha a um braço humano [15].

2.6. CONTROLO DOS ROBÔS

O controlador dos robôs é normalmente uma unidade capaz de gerar informação de ativação de um ou mais atuadores com base num algoritmo de controlo. Esse algoritmo pode levar em linha de conta o comando desejado, o estado corrente do atuador, e o próprio ambiente. Os algoritmos mais comuns recorrem ao chamado controlo Proporcional-Integral-Diferencial (PID), podendo em certos casos específicos ter-se o controlo de lógica difusa (*Fuzzy Logic*).

Quando não existente em separado, o controlador chega a incluir a unidade de potência, ou seja, o elemento que liga diretamente ao atuador, fornecendo-lhe a energia que necessita com base numa informação de baixo teor energético, como são os sinais elétricos à saída de muitos controladores. Por vezes o controlador faz parte de um sistema maior de interligação com o utilizador exterior, como é o caso de um computador pessoal. Nestes casos o controlador pode assumir a forma de uma carta de expansão do próprio computador, ou ser um dispositivo exterior que comunica com o computador de uma forma padrão, como por exemplo uma ligação série RS-232.

Esta integração num computador, devido ao abundante número de ferramentas de interface disponíveis, permite especificar mais facilmente os comandos desejados para o controlador, ou até de os gerar de forma automática, como fazem muitos programas (*software*).

O controlo de robôs está subdividido em duas categorias:

- Robôs não inteligentes: são robôs controlados por unidades de processamento e o algoritmo usado segue as ordens programadas para o robô se mover.
- Robôs inteligentes: são robôs que também são compostos por unidades de processamento, mas o *software* que está no controlador, é um sistema capaz de interagir com seu ambiente através de sensores e tomar decisões em tempo real.

2.7. ATUADORES DE SISTEMAS DE ACIONAMENTO

Entende-se por atuador o elemento capaz de converter um determinado tipo de energia numa ação, em resposta a comandos que podem ser manuais ou automáticos. Por exemplo, se o funcionamento do sistema estiver baseado em algum movimento de uma de suas partes, serão necessários atuadores para fornecer energia mecânica para o movimento, ou se o sistema for térmico, será necessário um atuador que forneça energia térmica para atingir uma dada temperatura desejada. O mesmo se passa com os outros tipos de atuadores.

Existe três tipos de sistemas de acionamento nos robôs, sendo eles:

- Acionamento hidráulico;
 - Acionamento pneumático;
- Acionamento elétrico.

2.7.1. ACIONAMENTO HIDRÁULICO

Os principais componentes deste sistema são: motor(es), cilindro(s), bomba(s) de óleo, válvula(s) e tanque(s) de óleo. A bomba é responsável pelo fluxo de óleo no cilindro em direção ao pistão que movimenta a junta.

Este tipo de acionamento permite valores elevados de velocidade e força, mas tem como grande desvantagem um custo muito elevado. São preferíveis em ambientes nos quais os de acionamento elétrico possam causar problemas, como arcos elétricos.

Assim, este tipo de acionador é geralmente associado a robôs de grande porte, quando comparados aos acionadores pneumáticos e elétricos, embora a sua precisão em relação aos acionadores elétricos seja menor. Na Figura 22 pode-se observar um robô com este tipo de acionamento.



Figura 22 Robô de acionamento hidráulico [17]

2.7.2. ACIONAMENTO PNEUMÁTICO

Os acionadores pneumáticos são semelhantes aos acionadores hidráulicos, a diferença está na utilização de ar em vez de óleo.

É de levar em consideração que a utilização de um robô com acionamento pneumático (Figura 23) deverá dispor de uma instalação de ar comprimido, incluindo compressor, sistema de distribuição, filtros, secadores, etc. Geralmente, devido à compressibilidade do ar, os atuadores pneumáticos não conseguem uma boa precisão de posicionamento, e as cargas máximas suportadas (forças e momentos de torção) são menores relativamente aos sistemas hidráulicos.

A vantagem deste tipo de sistemas reside principalmente no seu baixo custo, segurança e facilidade de operação. De salientar que a maioria dos espaços industriais já tem uma instalação de ar comprimido que pode ser utilizada.

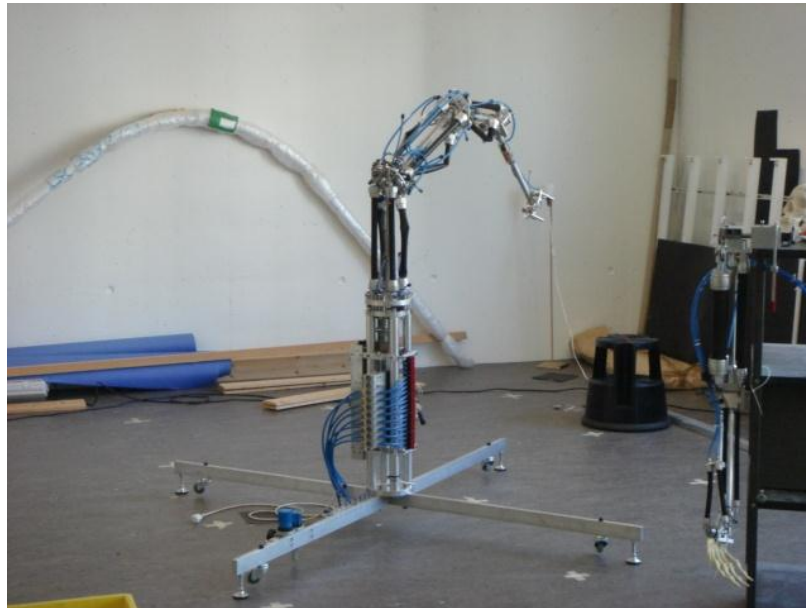


Figura 23 Robô de acionamento pneumático [18]

2.7.3. ACIONAMENTO ELÉTRICO

De entre os tipos de atuadores existentes, os atuadores elétricos são aqueles cada vez mais utilizados devido às suas características de facilidade de controle, sensibilidade e precisão. No entanto oferecem menor força e velocidade.

Existem três grandes tipos de atuadores elétricos:

- Motores de corrente contínua e alternada;
- Servomotores de corrente contínua e alternada;
- Motores passo a passo.

No que diz respeito aos motores de corrente alternada, estes são os que têm tido mais aplicações na área da robótica devido aos custos de manutenção serem baixos e à facilidade de controle. Os servomotores de corrente alternada são motores de pequena dimensão, de baixa inércia e que podem produzir elevadas acelerações e desacelerações. São controlados mediante referências de velocidade e são os mais utilizados atualmente devido ao seu fácil controle. Os motores passo a passo são motores muito leves, fiáveis e fáceis de controlar. A sua principal vantagem em relação aos tradicionais servomotores é a

capacidade para assegurar um posicionamento simples e exato, podendo rodar de forma contínua e com várias velocidades. Na Figura 24 pode-se observar um exemplar de robô de acionamento elétrico.



Figura 24 Robô de acionamento elétrico

2.7.4. COMPARAÇÃO DOS TIPOS DE ATUADORES

Na Tabela 3 pode-se encontrar de forma resumida o que foi referido nas secções 2.7.1, 2.7.2 e 2.7.3.

Tabela 3 Comparação dos tipos de atuadores

Características	Tipo de atuador		
	Hidráulico	Pneumático	Elétrico
Controlo	Fácil com válvulas e servo-válvulas	Muito difícil devido a questões de compressibilidade do ar	Fácil, possibilidade de ser elaborado
Velocidade	Média/grande	Muito grande	Grande
Binário	Grande	Pequeno	Pequeno/médio
Precisão	Boa	Má, exceto em operações a posições fixas	Boa, limitada pelo uso de transmissão
Funcionamento em situação estática	Excelente, trata-se de funcionamento normal	Bom, não há risco de danificação do sistema	Mau, requer travões
Questões ambientais	Perigoso, fugas de óleo	Sistemas limpos, risco de poluição sonora de componentes, compressores e das fugas	A presença de arcos elétricos pode ser indesejada
Custos	Elevados	Baixos	Baixos

2.8. SENSORES

Os sensores são elementos destinados à medição do estado interno do manipulador bem como à percepção do ambiente exterior. Os principais tipos de sensores usados num manipulador são: sensores de posição, sensores de velocidade, fins-de-curso, sensores de força, detetores de proximidade capacitivos e indutivos.

Estes fornecem informação ao sistema de controlo, nomeadamente em que posições se encontram as diversas juntas do manipulador, sensores que delimitam as deslocações externas das juntas e os que fornecem informação do meio ambiente envolvente.

2.8.1. SENSORES DE POSIÇÃO

2.8.1.1. Potenciómetro

O sensor indicado na Figura 25 é um potenciómetro angular, onde a posição é obtida através de uma queda de tensão proveniente da resistência dada pelo cursor do potenciómetro, por exemplo, se tivermos um potenciómetro, cujo cursor pode girar entre zero a trezentos e sessenta graus, o valor da resistência dada pelo cursor varie entre zero e mil Ohm. Se a resistência dada pelo cursor for de quinhentos Ohm (180°) e o potenciómetro for alimentado a cinco Volt, têm-se à saída 2,5 Volt, o que significa que a posição obtida é metade da posição máxima.

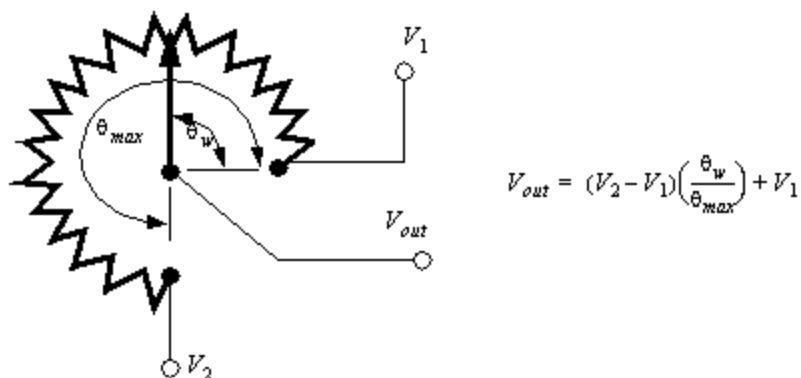


Figura 25 Potenciómetro para indicar posições angulares [19]

2.8.1.2. Codificadores angulares

Os codificadores angulares apresentam a vantagem de não possuírem contactos sujeitos a desgaste como o cursor de um potenciómetro. Na Figura 26 apresenta-se um codificador do tipo incremental. Neste codificador tem-se um bit para sinalizar uma posição de referência (posição de *Home*) e outro que vai alternar entre 0 e 1 sempre que exista movimento.

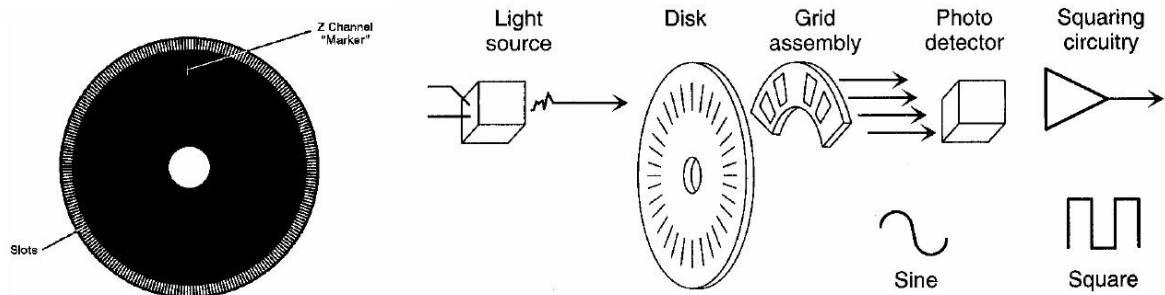


Figura 26 Codificador incremental e princípio de funcionamento

Na Figura 27 mostra-se um codificador absoluto. Neste codificador é usado normalmente o código de Gray, ou código binário inventado por Frank Gray, sendo um sistema de numeração binária em que dois valores sucessivos diferem em apenas um bit.

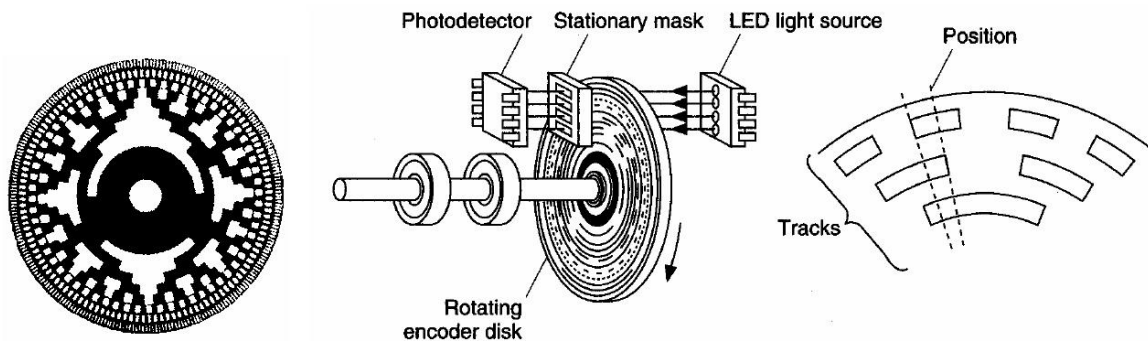


Figura 27 Codificador absoluto e princípio de funcionamento

2.8.2. SENSORES DE VELOCIDADE

Para indicador de velocidade é normalmente usado um tacómetro. Os tacómetros podem ser mecânicos ou óticos. Na Figura 28 apresenta-se um tacómetro mecânico. O sensor deteta movimento através dos dentes da roda, sendo a velocidade dada pela expressão:

$$v \left(\frac{\text{voltas}}{\text{unidade de tempo}} \right) = \frac{\text{impulso}}{\text{n}^{\circ} \text{dedentes} \cdot \text{tempo}} \quad (1)$$



Figura 28 Tacómetro mecânico

A Figura 29 representa um tacómetro ótico, onde o funcionamento é semelhante ao tacómetro mecânico, só que neste tipo de sensor existe um emissor de luz, um recetor de luz e uma roda com perfurações. Sempre que o recetor de luz deteta luz, significa que existe movimento, sendo a velocidade dada também pela equação (1).

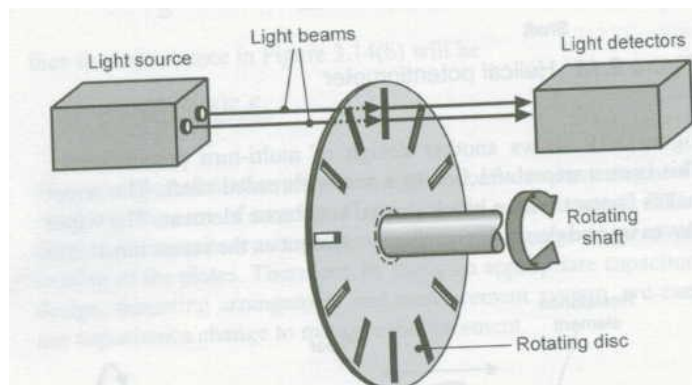


Figura 29 Tacómetro ótico

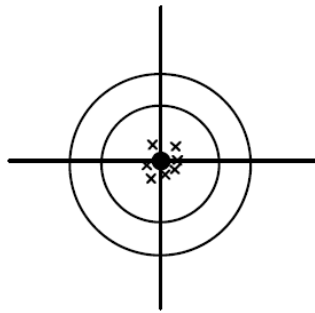
2.9. RESOLUÇÃO, REPETIBILIDADE E PRECISÃO

Existem alguns conceitos que são utilizados para a escolha de um robô. Entre estes estão, a resolução, a repetibilidade e a precisão. A resolução refere-se ao menor movimento incremental da junta. Por exemplo, se o dispositivo de medição detetar 720 incrementos em 360° , ou seja $360^\circ/720=0,5^\circ$, isto significa que se tem uma resolução de $0,5^\circ$ por incremento de uma junta.

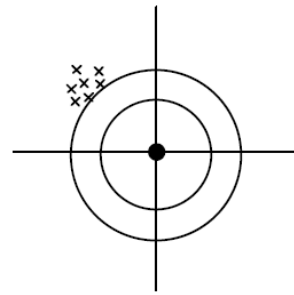
Repetibilidade traduz a diferença de posição com que o robô volta a recolocar-se num ponto visitado anteriormente.

Precisão é dada pela diferença entre uma posição realmente atingida e a posição desejada pela programação. Esta está relacionada com a resolução, sendo que é afetada pela carga transportada e é variável conforme a zona do espaço de trabalho.

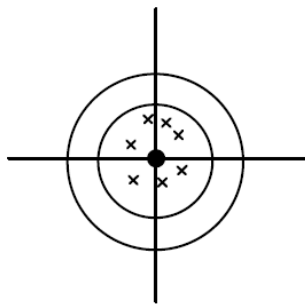
A Figura 30 ilustra a relação entre repetibilidade e precisão.



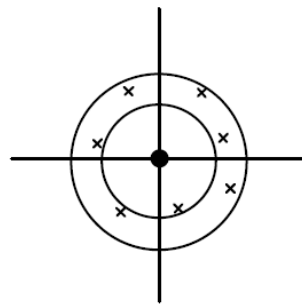
Boa precisão
Boa repetibilidade



Má precisão
Boa repetibilidade



Boa precisão
Má repetibilidade



Má precisão
Má repetibilidade

Figura 30 Ilustração dos conceitos de repetibilidade e precisão

3. CINEMÁTICA, DINÂMICA E PLANEAMENTO DE TRAJETÓRIAS

Neste capítulo serão abordados todos os conceitos matemáticos utilizados no controlo dos robôs e como é elaborado o planeamento de trajetórias.

As primeiras duas secções apresentadas dizem respeito à explicação e cálculo da cinemática direta e inversa.

O documento também apresenta uma secção com uma técnica matemática alternativa para o cálculo da cinemática direta, para robôs com mais de três graus de liberdade, sendo esta conhecida como a convenção/formalismo de Denavit-Hartenberg.

Por fim tem-se uma secção que diz respeito a como se procede para o cálculo de trajetórias, sendo estas no espaço das juntas ou no espaço operacional.

3.1. CINEMÁTICA

Cinemática é o ramo da mecânica que estuda o movimento de corpos independentemente das causas e efeitos que o produzem ou modificam (as causas e efeitos dos movimentos denomina-se por Dinâmica). Em robótica, a cinemática é dada pelas relações geométricas entre as coordenadas no espaço operacional e no espaço das juntas. Para o cálculo da cinemática não se consideram as massas, forças e/ou binários que provocam o movimento do robô.

A título de exemplo, as variáveis para efetuar o cálculo da cinemática, consoante a configuração dos robôs, são:

- Robôs RR
 - Coordenadas no espaço operacional – (x, y)
 - Coordenadas no espaço das juntas – (θ_1, θ_2)
- Robôs RP
 - Coordenadas no espaço operacional – (x, y)
 - Coordenadas no espaço das juntas – (θ, L)

A cinemática de um braço robótico só diz respeito aos parâmetros que definem a estrutura geométrica dos elos e aos ângulos das juntas. A modelação cinemática é efetuada em relação a um sistema fixo de referência de coordenadas, geralmente denominado de sistema de coordenadas da base. Em relação à cinemática direta diferencial, esta estuda as velocidades/acelerações descrevendo o deslocamento espacial do atuador final do robô em relação ao tempo.

A cinemática direta calcula, para um conjunto de valores dos ângulos das juntas e/ou comprimentos dos eixos prismáticos e de parâmetros dos elos, a posição/orientação do atuador final, enquanto, a cinemática inversa fornece os ângulos das juntas e/ou comprimentos dos eixos prismáticos, para um dado conjunto de parâmetros dos elos e para uma posição/orientação do atuador final.

No caso dos robôs com dois graus de liberdade, e a título de exemplo, têm-se as seguintes relações cinemáticas.

Cinemática

- Direta
 - Robôs RR – $(\theta_1, \theta_2) \rightarrow (x, y)$
 - Robôs RP – $(\theta, L) \rightarrow (x, y)$
- Inversa
 - Robôs RR - $(x, y) \rightarrow (\theta_1, \theta_2)$
 - Robôs RP - $(x, y) \rightarrow (\theta, L)$

A relação entre a cinemática direta e inversa pode ser observada no esquema da Figura 31.

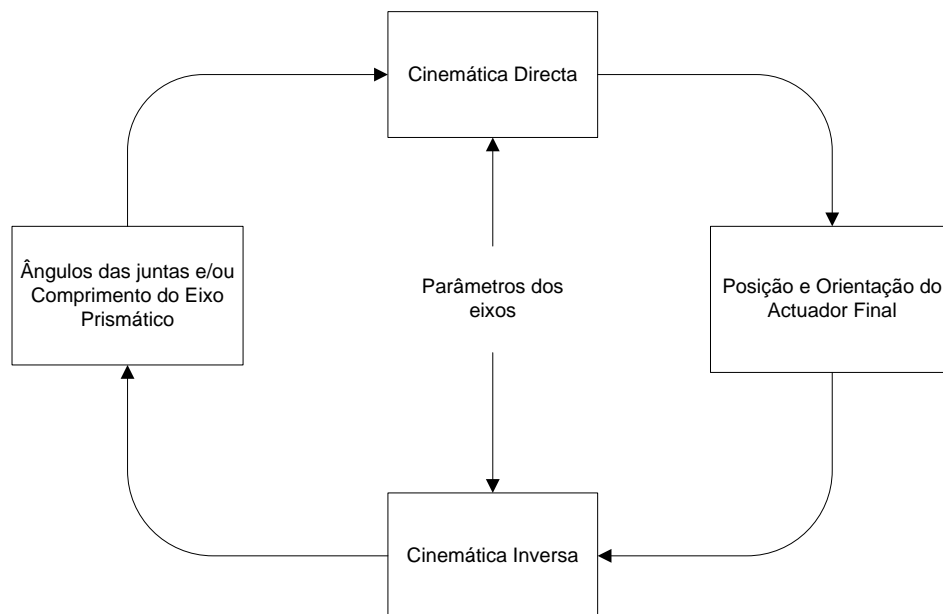


Figura 31 Relação entre cinemática direta e cinemática inversa

Usualmente, a definição da posição e orientação do punho do robô é mais conveniente para um programador se for especificada relativamente ao sistema de coordenadas cartesiano da base do robô ou relativamente a um sistema definido pelo utilizador. Se a programação é orientada ao espaço cartesiano, é necessário implementar a cinemática inversa.

As técnicas geralmente utilizadas para determinar a cinemática de um robô são, em casos mais simples, relações trigonométricas e noutros casos o formalismo de Denavit-Hartenberg (secção 3.3).

A resolução simbólica da cinemática direta, pode-se encontrar facilmente usando o formalismo de Denavit-Hartenberg. A resolução simbólica de cinemática inversa é muito mais complexa levando muitas vezes a obter várias soluções.

Através do seguinte exemplo pode-se observar que se obtêm duas soluções da cinemática inversa para uma mesma localização no espaço cartesiano de um robô SCARA (Figura 32).

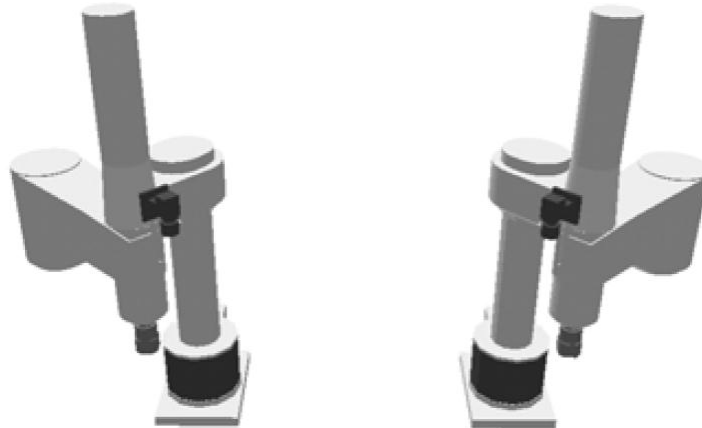


Figura 32 Robô SCARA – configuração *right elbow* e *left elbow*

Na Figura 32 tem-se à esquerda a configuração *Right Elbow* (cotovelo direito) e à direita a configuração *Left Elbow* (cotovelo esquerdo), sendo possível observar que se têm duas configurações diferentes das juntas para a mesma posição final.

Como se pode verificar, para uma dada posição no espaço cartesiano, através da cinemática inversa obtêm-se diversas soluções. É importante salientar que no mundo real, quando uma dada posição do atuador final não pode ser alcançada devido a uma colisão do braço robótico com o meio envolvente, pode ser necessário recorrer a uma das diversas soluções da cinemática inversa, para que a configuração esteja isenta de colisões [20][21][22].

3.1.1. CINEMÁTICA DIRETA

Como já foi referido, a cinemática direta traduz, para um conjunto de valores dos ângulos das juntas, comprimentos dos eixos prismáticos e de parâmetros dos elos, a posição/orientação do atuador final.

Para um robô com dois graus de liberdade rotacionais (RR), como mostra a Figura 33, a cinemática direta traduz-se na equação (2).

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} L_1 \cdot \cos(\theta_1) + L_2 \cdot \cos(\theta_1 + \theta_2) \\ L_1 \cdot \sin(\theta_1) + L_2 \cdot \sin(\theta_1 + \theta_2) \end{bmatrix} \quad (2)$$

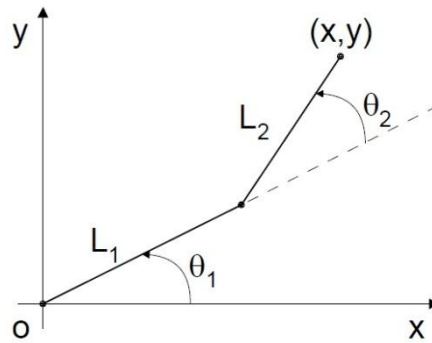


Figura 33 Representação de um robô com duas juntas rotacionais

Na cinemática direta só existe uma solução (x, y) para cada par de valores (θ_1, θ_2) .

Para um robô que contém uma junta rotacional e uma junta prismática, como mostra a Figura 34, a cinemática direta traduz-se na equação (3).

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} (L_1 + L_2) \cdot \cos(\theta_1) \\ (L_1 + L_2) \cdot \sin(\theta_1) \end{bmatrix} \quad (3)$$

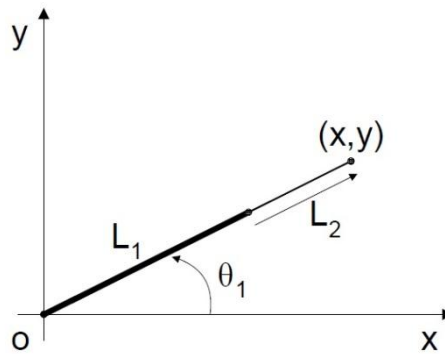


Figura 34 Representação de um robô com uma junta rotacional e uma junta prismática

3.1.2. CINEMÁTICA INVERSA

A cinemática inversa traduz qual é o valor das coordenadas das juntas para uma posição no espaço operacional.

Por exemplo, a cinemática inversa para um robô com configuração RR (Figura 35), é dada pelas equações (4)

$$\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} \text{atan2}(y, x) - \text{atan2}(L_2 \cdot \sin(\theta_2), L_1 + L_2 \cdot \cos(\theta_2)) \\ \arccos\left(\frac{x^2 + y^2 + L_1^2 + L_2^2}{2 \cdot L_1 \cdot L_2}\right) \end{bmatrix} \quad (4)$$

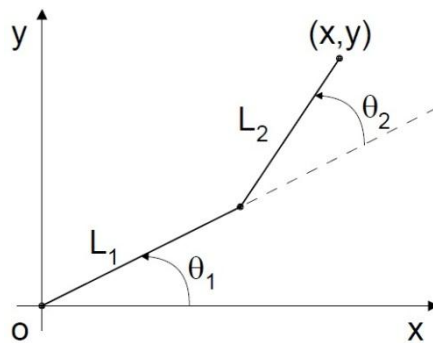


Figura 35 Representação de um robô com duas juntas rotacionais

No caso da configuração do robô RP (Figura 36), a cinemática inversa traduz-se na equação (5) e (6).

$$\begin{bmatrix} \theta_1 \\ L_2 \end{bmatrix} = \begin{bmatrix} \text{atan2}(y, x) \\ \sqrt{x^2 + y^2} - L_1 \end{bmatrix} \quad (5)$$

$$\begin{bmatrix} \theta_1 \\ L \end{bmatrix} = \begin{bmatrix} \text{atan2}(y, x) \\ \sqrt{x^2 + y^2} \end{bmatrix}, L = L_1 + L_2 \quad (6)$$

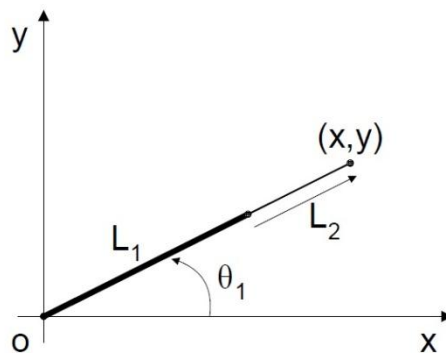


Figura 36 Representação de um robô com uma junta rotacional e uma junta prismática

Na cinemática direta só existe uma solução $(x; y)$ para cada par de valores $(\theta_1; \theta_2)$, enquanto na cinemática inversa para uma posição $(x; y)$ obtêm-se múltiplas soluções para cada par de valores $(\theta_1; \theta_2)$ (só se obtêm múltiplas soluções se a configuração do robô tiver mais de uma junta rotacional e estas possuírem uma amplitude de movimento superior a 180°), isto porque, vai depender da configuração que braço do robô tem para a dada posição. Na Figura 37 pode-se observar quatro tipos de configuração que o braço robótico adota para a mesma posição final, que são: braço esquerdo para cima (*left and above arm*), braço direito

para cima (*right and above arm*), braço esquerdo para baixo (*left and bellow arm*) e braço direito para baixo (*right and bellow arm*).

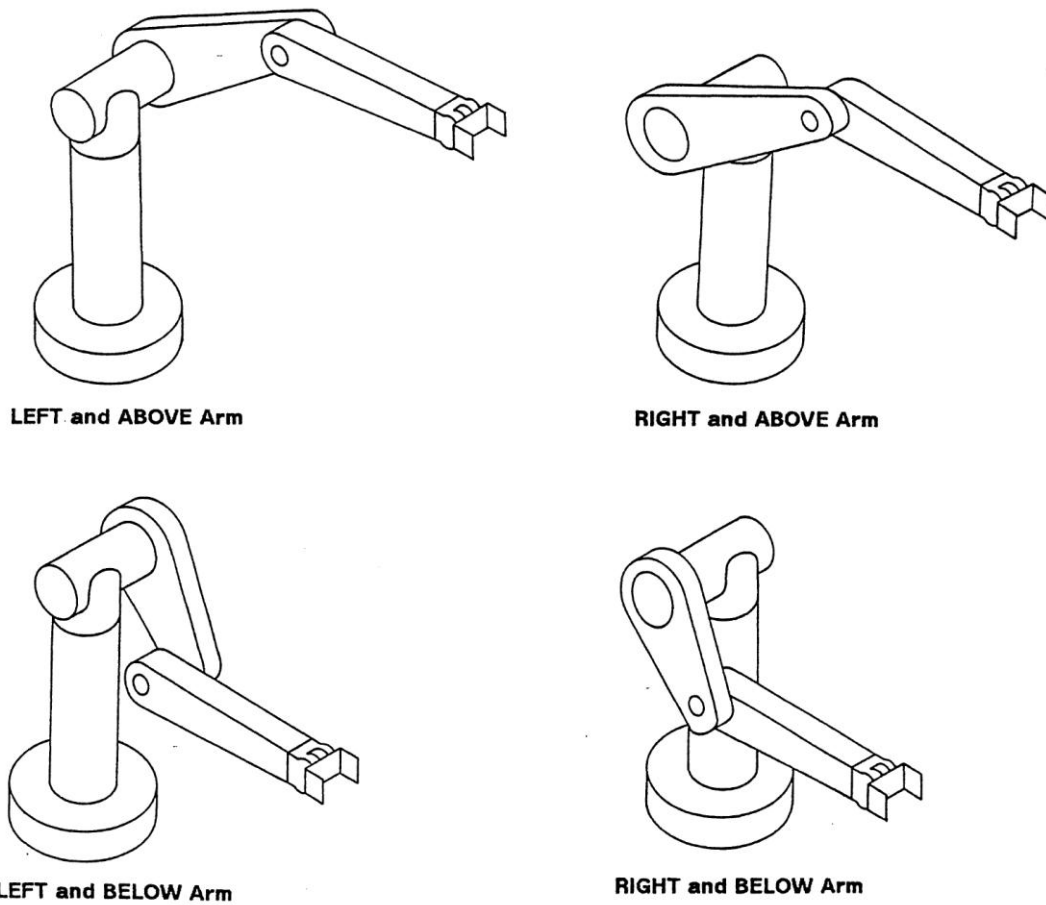


Figura 37 Configuração do braço robótico PUMA para uma mesma posição final

3.2. DINÂMICA

A dinâmica é um ramo da mecânica que estuda as relações entre as forças e os movimentos por elas produzidos, sendo que a esta estão associados as três Leis de Newton. Na robótica, a dinâmica são as equações que estabelecem a relação entre forças/binários (dos atuadores) com as posições, velocidades e acelerações, que ocorrem numa dada trajetória do robô. Para um manipulador de n graus de liberdade, a dinâmica segue as leis da física clássica e pode ser expressa através de um conjunto de equações diferenciais.

Como na cinemática, também aqui se tem a dinâmica direta e a dinâmica inversa. A expressão (7) corresponde à dinâmica inversa.

$$T = [J(\theta)].\ddot{\theta} + [N(\theta)].\dot{\theta}^2 + [P(\theta)].\dot{\theta}_i\dot{\theta}_j + G(\theta) \quad (7)$$

T – vetor dos binários dos atuadores ($n \times 1$)

θ – Vetor das coordenadas no espaço das juntas

$J(\theta)$ – matriz simétrica das inércias ($n \times n$)

$N(\theta)$ – matriz dos binários centrífugos ($n \times n$)

$P(\theta)$ – matriz dos binários Coriolis ($n \times C_n^2$)

$G(\theta)$ – vetor dos binários gravitacionais ($n \times 1$)

3.2.1. DINÂMICA DIRETA

A dinâmica direta permite calcular as acelerações, velocidades e posições das juntas do robô a partir das forças/binários desenvolvidos pelos atuadores das juntas [22][23][24].

A dinâmica direta é calculada segundo as expressões (8), (9), (10) e (11).

$$\{T(t), \theta(t=0), \dot{\theta}(t=0)\} \rightarrow \{\theta(t), \dot{\theta}(t), \ddot{\theta}(t)\} \quad (8)$$

$$\ddot{\theta}(t) = J(\theta)^{-1} \cdot [T - C(\theta, \dot{\theta}) - G(\theta)] \quad (9)$$

$$\dot{\theta}(t) = \int_0^t \ddot{\theta} \cdot dt + \dot{\theta}(0) \quad (10)$$

$$\theta(t) = \int_0^t \dot{\theta} \cdot dt + \theta(0) \quad (11)$$

3.2.2. DINÂMICA INVERSA

A dinâmica inversa permite calcular os binários/forças que surgem a partir das posições, velocidades e acelerações nas juntas do robô.

$$\{\theta(t), \dot{\theta}(t), \ddot{\theta}(t)\} \rightarrow \{T(t)\} \quad (12)$$

$$T = J(\theta) \cdot \ddot{\theta} + C(\theta, \dot{\theta}) + G(\theta) \quad (13)$$

O principal problema com a aplicação de controlo dinâmico em tempo real é a complexidade de muitos termos na equação da dinâmica inversa. Em robôs com seis graus

de liberdade as expressões resultantes são muito complexas, sendo difícil a sua obtenção manualmente. Então, para resolver este problema, faz-se uso do método de Lagrange, onde a energia cinética da ligação i , k_i , de um manipulador é dada por uma expressão do tipo:

$$k_i = \frac{1}{2} \cdot m_i \cdot v_{ci}^T \cdot v_{ci} + \left(\frac{1}{2}\right)^i \omega_i^T \cdot C_i \cdot I_i^i \cdot \omega_i \quad (14)$$

Onde o primeiro termo é a energia cinética devido à velocidade linear do centro de massa do elo e o segundo termo é a energia cinética devido à velocidade angular do elo.

3.3. FORMALISMO DE DENAVIT-HARTENBERG

O formalismo de Denavit-Hartenberg (DH) veio simplificar o cálculo da cinemática direta. Na cinemática direta para robôs cujo volume de trabalho se encontra nos eixos cartesianos x , y e z , seria necessário proceder a três rotações (uma por cada eixo), mais três translações (uma para cada eixo – Jacques Denavit e Richard Hartenberg simplificaram as seis variáveis para quatro, mas para se obter esta simplificação é necessário obedecer às regras da convenção de Denavit-Hartenberg.

As regras da convenção de Denavit-Hartenberg são as seguintes:

- Um robô tem n sistemas de coordenadas, sendo n o número total de juntas;
- O eixo de rotação da junta é sempre paralelo ao eixo z_i ;
- Se o eixo for primático, o eixo z_i é obtido pelo o sentido em que se desloca a junta;
- O eixo y_{n-1} é sempre paralelo ao eixo de abertura e fecho da garra e o eixo z_{n-1} é sempre paralelo ao eixo de orientação da garra;
- Depois de se definir o eixo z_i , obtêm-se os eixos x_i e y_i seguindo a regra da mão direita².

² A regra da mão direita é uma regra e recurso mnemônico geralmente utilizada quando se necessita diferenciar e/ou estabelecer como padrão uma entre duas orientações espaciais. Para se obter um sistema de coordenadas espaciais (x ; y ; z), segundo a regra da mão direita o eixo x é dado pelo indicador direito, eixo y pelo dedo médio direito e o eixo z pelo polegar direito. Depois tentamos criar um ângulo de 90° entre o indicador e o dedo do meio e por fim um ângulo de 90° entre o polegar e o indicador e assim obtemos um sistema de eixos espaciais segundo a regra da mão direita. O sentido positivo do ângulo é obtido através do fecho da mão direita e o polegar apontar no sentido positivo do eixo pelo qual irá ser feita a rotação.

Como já referido, a convenção de Denavit-Hartenberg reduziu de seis para quatro o número de parâmetros necessários para definir a localização relativa entre o sistema de coordenadas da base com o sistema de coordenadas da garra. Depois de se colocar os sistemas de coordenadas em cada junta, procede-se à criação da matriz A_i , que resume a translação e rotação da junta i vista pela junta $i-1$. A matriz A_i é do tipo da expressão (15).

$${}^{n-1}A_n = \begin{bmatrix} \cos\theta_n & -\cos\alpha_n \cdot \sin\theta_n & \sin\alpha_n \cdot \sin\theta_n & a_n \cdot \cos\theta_n \\ \sin\theta_n & \cos\alpha_n \cdot \cos\theta_n & -\sin\alpha_n \cdot \cos\theta_n & a_n \cdot \sin\theta_n \\ 0 & \sin\alpha_n & \cos\alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15)$$

Esta matriz A_i é construída com base noutras matrizes, como demonstra a expressão 16.

$$A_n = Z_n \cdot X_n = Rot_z \cdot Trans_z \cdot Rot_x \cdot Trans_x \quad (16)$$

A matriz é desenvolvida utilizando as expressões (17), (18), (19) e (20).

$$Rot_{xn}(\alpha_n) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_n) & -\sin(\alpha_n) & 0 \\ 0 & \sin(\alpha_n) & \cos(\alpha_n) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (17)$$

$$Trans_{xn}(a_n) = \begin{bmatrix} 1 & 0 & 0 & a_n \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (18)$$

$$Rot_{zn-1}(\theta_n) = \begin{bmatrix} \cos(\theta_n) & -\sin(\theta_n) & 0 & 0 \\ \sin(\theta_n) & \cos(\theta_n) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (19)$$

$$Trans_{zn-1}(d_n) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (20)$$

As expressões (17) – (20) podem-se resumir nas equações (21) e (22), sendo a primeira relacionada com as translações e rotações sobre o eixo x e a outra sobre as rotações e translações sobre o eixo z .

$$[X_i] = Trans_{X_i}(\alpha_{i,i+1}) \cdot Rot_{X_i}(\alpha_{i,i+1}) \quad (21)$$

$$[Z_i] = Trans_{Z_i}(d_i) \cdot Rot_{Z_i}(\theta_i) \quad (22)$$

Da matriz A_i têm-se quatro variáveis, que vão traduzir a translação e rotação de uma junta $i-1$ para uma junta i . As variáveis são as seguintes e têm o seguinte significado:

- θ_i - ângulo formado entre x_{i-1} e x_i medido em z_{i-1}

- α_i – ângulo formado entre z_{i-1} e z_i , e medido em x_i
- a_i - distância ao longo de x_i desde a intersecção de x_i com z_{i-1}
- d_i – distância ao longo de z_{i-1} desde a intersecção de x_i com z_{i-1}

São criadas n matrizes sendo que n corresponde ao número total de juntas. A seguir procede-se ao desenvolvimento da matriz T, que corresponde à relação entre a origem e a extremidade do robô. A matriz T obtém-se através da multiplicação das n matrizes, como demonstra a expressão (23).

$$T = A_0 A_1 \dots A_{n-1} \quad (23)$$

A matriz T é então dada por:

$$T = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (24)$$

Sendo que p_x , p_y e p_z correspondem à posição da extremidade do robô em relação ao sistema de coordenadas da origem.

3.4. PLANEAMENTO DE TRAJETÓRIAS

O planeamento de trajetórias consiste na determinação de um trajeto, fornecendo ao robô uma sequência de pontos, para este se mover de um ponto para outro (Cinemática) com velocidades e acelerações apropriadas (Dinâmica).

Quando se pretende deslocar um manipulador de uma posição inicial P_i para uma posição final P_f , é necessário o controlo decidir como deve executar o deslocamento. O planeamento de trajetórias pode ser efetuado no espaço das juntas ou no espaço operacional e é definido assim para um robô RR:

- No espaço das juntas: $\begin{cases} \bar{\theta}_{1,k} \\ \bar{\theta}_{2,k} \end{cases}, k = 1, 2, \dots$
- No espaço operacional: $\begin{cases} \bar{x}_k \\ \bar{y}_k \end{cases}, k = 1, 2, \dots$

Se for necessário ter um controlo pormenorizado poderá ser necessário definir a primeira e segunda derivadas temporais.

3.4.1. INTERPOLAÇÃO LINEAR

A interpolação linear, como o próprio nome indica, serve para planejar trajetórias lineares ou retilíneas entre dois pontos, P_i e P_f . Considerando um robô RR temos a seguinte interpolação linear, onde i é o valor inicial e f o valor final.

$$\text{No espaço das juntas: } \begin{cases} \overline{\theta}_1(t) = (1-t) \cdot \overline{\theta}_{1,i} + t \cdot \overline{\theta}_{1,f} \\ \overline{\theta}_2(t) = (1-t) \cdot \overline{\theta}_{2,i} + t \cdot \overline{\theta}_{2,f} \end{cases}, t \in [i, f]$$

$$\text{No espaço operacional: } \begin{cases} \overline{x}(t) = (1-t) \cdot \overline{x}_i + t \cdot \overline{x}_f \\ \overline{y}(t) = (1-t) \cdot \overline{y}_i + t \cdot \overline{y}_f \end{cases}, t \in [i, f]$$

No espaço das juntas, para se determinar a posição do robô ao longo da trajetória recorre-se à cinemática direta, pois esta fornece a posição consoante o valor da posição das juntas. No espaço operacional é necessário determinar as posições das juntas do robô ao longo da trajetória, recorrendo para isso à cinemática inversa.

3.4.2. INTERPOLAÇÃO CÚBICA

A equação do planeamento de trajetórias segundo a interpolação cúbica é dada segundo a expressão (25).

$$\theta(t) = a_0 + a_1 \cdot t + a_2 \cdot t^2 + a_3 \cdot t^3 \quad (25)$$

Para a obtenção da velocidade e aceleração tem-se as expressões (26) e (27).

$$\dot{\theta}(t) = a_1 + 2a_2 \cdot t + 3a_3 \cdot t^2 \quad (26)$$

$$\ddot{\theta}(t) = 2a_2 + 6a_3 \cdot t \quad (27)$$

Na interpolação cúbica para se efetuar um movimento suave é necessário implementar pelo menos quatro restrições. As quatro restrições são as seguintes:

- Posição inicial igual a θ_i ;
- Posição final igual a θ_f ;
- Velocidade inicial igual a zero, $\dot{\theta}_i = 0$;
- Velocidade final igual a zero, $\dot{\theta}_f = 0$.

Através das restrições obtém-se as expressões (28), (29), (30) e (31).

$$\theta_i = a_0 \quad (28)$$

$$\theta_f = a_0 + a_1 \cdot t + a_2 \cdot t^2 + a_3 \cdot t^3 \quad (29)$$

$$0 = a_1 \quad (30)$$

$$0 = a_1 + 2a_2 \cdot t + 3a_3 \cdot t^2 \quad (31)$$

onde a_0 , a_1 , a_2 e a_3 correspondem às expressões:

$$a_0 = \theta_i \quad (32)$$

$$a_1 = 0 \quad (33)$$

$$a_2 = \frac{3}{t_f^2}(\theta_f - \theta_i) \quad (34)$$

$$a_3 = -\frac{2}{t_f^3}(\theta_f - \theta_i) \quad (35)$$

Com as restrições tem-se uma evolução da posição que se assemelha a uma função cúbica, a velocidade a uma parábola e uma aceleração linear.

3.4.3. INTERPOLAÇÃO DE QUINTA ORDEM

A interpolação de quinta ordem é adequada quando se pretende especificar a posição, velocidade e aceleração da posição inicial e final. A interpolação de quinta ordem é dada por [23][24][25]:

$$\theta(t) = a_0 + a_1 \cdot t + a_2 \cdot t^2 + a_3 \cdot t^3 + a_4 \cdot t^4 + a_5 \cdot t^5 \quad (36)$$

Para a obtenção da velocidade e aceleração tem-se as seguintes expressões:

$$\dot{\theta}(t) = a_1 + 2a_2 \cdot t + 3a_3 \cdot t^2 + 4a_4 \cdot t^3 + 5a_5 \cdot t^4 \quad (37)$$

$$\ddot{\theta}(t) = 2a_2 + 6a_3 \cdot t + 12a_4 \cdot t^2 + 20a_5 \cdot t^3 \quad (38)$$

As restrições da interpolação de quinta ordem são então dadas pelas expressões:

$$\theta_i = a_0 \quad (39)$$

$$\theta_f = a_0 + a_1 \cdot t + a_2 \cdot t^2 + a_3 \cdot t^3 + a_4 \cdot t^4 + a_5 \cdot t^5 \quad (40)$$

$$\dot{\theta}_i = a_1 \quad (41)$$

$$\dot{\theta}_f = a_1 + 2a_2 \cdot t + 3a_3 \cdot t^2 + 4a_4 \cdot t^3 + 5a_5 \cdot t^4 \quad (42)$$

$$\ddot{\theta}_i = a_2 \quad (43)$$

$$\ddot{\theta}_f = 2a_2 + 6a_3 \cdot t + 12a_4 \cdot t^2 + 20a_5 \cdot t^3 \quad (44)$$

A solução leva às equações:

$$a_0 = \theta_i \quad (45)$$

$$a_1 = \dot{\theta}_i \quad (46)$$

$$a_2 = \frac{\ddot{\theta}_i}{2} \quad (47)$$

$$a_3 = \frac{20\theta_f - 20\theta_i - (8\dot{\theta}_f + 12\dot{\theta}_i)t_f - (3\ddot{\theta}_i - \ddot{\theta}_f)t_f^2}{2t_f^3} \quad (48)$$

$$a_4 = \frac{30\theta_i - 30\theta_f + (14\dot{\theta}_f + 16\dot{\theta}_i)t_f + (3\ddot{\theta}_i - 2\ddot{\theta}_f)t_f^2}{2t_f^4} \quad (49)$$

$$a_5 = \frac{12\theta_f - 12\theta_i - (6\dot{\theta}_f + 6\dot{\theta}_i)t_f - (\ddot{\theta}_i - \ddot{\theta}_f)t_f^2}{2t_f^5} \quad (50)$$

3.5. MÉTODOS DE CONTROLO DO ROBÔ

Os robôs existentes no mercado têm como controlo, o controlo Proporcional-Integral-Diferencial (PID). Em casos mais específicos pode-se ter o controlo de lógica difusa ou *Fuzzy Logic*, um controlo inteligente, adotando para isso as redes neuronais e algoritmos evolutivos.

3.6. CONCLUSÕES

Neste capítulo apresentaram-se os aspetos matemáticos que estão relacionados com o controlo e movimento dos robôs. Pode-se sugerir que existe dois métodos para o cálculo da cinemática direta, que são, através de conceitos trigonométricos ou através da convenção de Denavit-Hartenberg. A convenção de Denavit-Hartenberg é o método mais prático e simples para calcular a cinemática direta para robôs que contenham mais de três juntas.

Para o cálculo da cinemática inversa, verificou-se que só é possível calcular através de desenvolvimento trigonométrico, como tal é necessário possuir algum conhecimento matemático.

Na secção referente ao planeamento de trajetórias, pode-se concluir que existe diversos métodos de planear o movimento de um robô, podendo este ser simples ou complexo e através de restrições controlar a sua velocidade e aceleração.

A conclusão deste capítulo resume-se numa perspetiva de trabalho futuro de investigação para um doutoramento, que consiste no desenvolvimento de uma fórmula matemática do género da convenção de Denavit-Hartenberg, para o cálculo da cinemática inversa e que nesta seja possível obter as múltiplas soluções que são sinónimo da cinemática inversa convencional. Se a realização do trabalho futuro tiver sucesso, a robótica é melhor manipulada e entendida por qualquer pessoa, não criando tabus para trabalhos realizados com a robótica.

4. ARQUITETURA DO SISTEMA

Ao longo deste capítulo são abordados todos os aspetos técnicos e científicos envolvidos na realização do projeto.

4.1. DEFINIÇÃO DOS REQUISITOS DO SISTEMA

Com a realização deste projeto pretende-se controlar um braço robótico, cujo sistema de controlo está muito descontinuado e como tal é necessário efetuar um *upgrade*. Para tal convém utilizar tecnologias recentes na implementação do controlo. O controlo do robô deverá conter uma unidade de processamento e *drivers* de motores. Mas para controlar o robô é necessário calcular as equações da cinemática direta e inversa, visto que, o braço robótico MENTOR não é muito conhecido e não existe nenhuma publicação das equações referentes à sua cinemática.

Deve-se desenvolver uma interface gráfica, de modo que seja “amigável” para o utilizador poder controlar o braço robótico.

Para concluir, o controlador do robô deverá planear as trajetórias de modo a permitir efetuar movimento no espaço das juntas e no espaço operacional.

4.2. BRAÇO ROBÓTICO MENTOR

O robô MENTOR, que se pode observar na Figura 38, é um braço robótico da Cybernetic Applications com o objetivo de ser prático e económico para aprender a conhecer o funcionamento de robôs. O braço robótico é constituído por 5 eixos, mais a garra. Os eixos e a garra são acionados por motores de corrente contínua acoplados a uma caixa redutora, sendo a posição de cada eixo dada por um potenciômetro de 10 k Ω que está acoplado à saída da caixa redutora [26].



Figura 38 Braço robótico MENTOR

Na Tabela 4 estão apresentadas as principais características de cada eixo.

Tabela 4 Características dos eixos

Eixo 0 – Coluna central	Movimento angular de 210° Comprimento do elo é de 185mm
Eixo 1 – Ombro	Movimento angular de 180° Comprimento do braço entre os centros dos eixos é de 165 mm
Eixo 2 - Braço	Movimento angular de 230° Comprimento desde o eixo central de 150 mm
Eixo 3* – Punho Esquerdo	Movimento angular de 320°
Eixo 4* – Punho Direito	Movimento angular de 320°
Rotação do punho (<i>roll</i>)	Movimento angular de 320°

Inclinação longitudinal do punho (<i>pitch</i>)	Movimento angular de 120°
Garra	Abertura de 30 mm Força de 10 Newton Distância final aos eixos 3 e 4 de 105 mm

*Na Figura 43 entende-se como é que se obtém a rotação (*roll*) e inclinação longitudinal (*pitch*) do punho, através da combinação entre o movimento do eixo 3 e do eixo 4.

Na Figura 39 encontra-se um esquema do robô com as respetivas medidas e gamas de movimento das juntas.

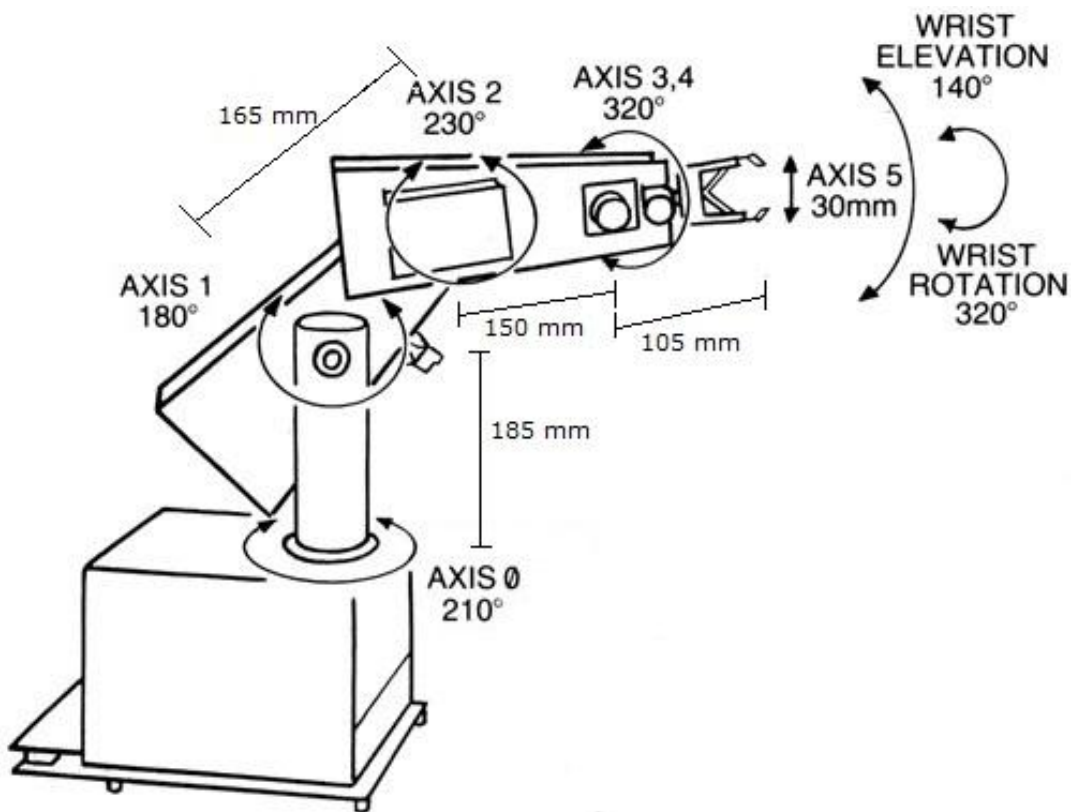


Figura 39 Esquema do braço robótico

Na Tabela 5, estão referidas algumas características do robô.

Tabela 5 Características principais do robô

Repetibilidade	2 mm
Capacidade de carga	1000 g
Alcance (desde o eixo central)	420 mm
Dimensões da base	Comprimento 320 mm Largura 270 mm Altura 189 mm

Na Figura 40 está representado um esquema referente à estrutura mecânica do eixo 0 do braço robótico. Este eixo corresponde à coluna central, que roda sobre uma engrenagem de *nylon* situada no topo da base do robô, sendo atuada pelo conjunto do motor-caixa redutora (caixa redutora de 500:1) com um par de engrenagens. A posição é determinada por um potenciômetro acoplado à parte inferior do eixo, como ilustra a Figura 40.

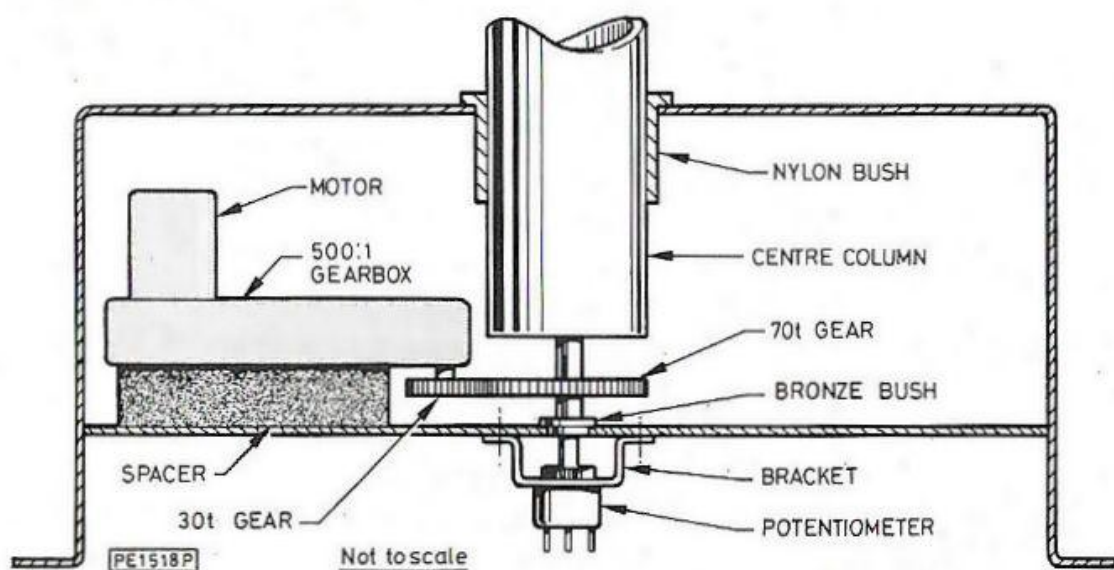


Figura 40 Estrutura mecânica do eixo 0 do robô

A Figura 41 corresponde à secção inferior do braço, que roda apoiada na coluna central. De igual forma, é usado um par de engrenagens para a transferência do movimento do conjunto motor-caixa redutora para o eixo 1. A presente figura também é válida para demonstrar o funcionamento do eixo 2, que corresponde à secção superior do braço, e que é atuado da mesma forma que o eixo 1. Na parte final deste braço encontra-se uma peça em aço, que funciona como contrabalanço, mantendo o braço equilibrado na posição quando

este não se encontra alimentado, evitando assim manter sempre o motor ativado para o braço permanecer na mesma posição. Da mesma forma existem contrapesos no braço anterior com a mesma finalidade.

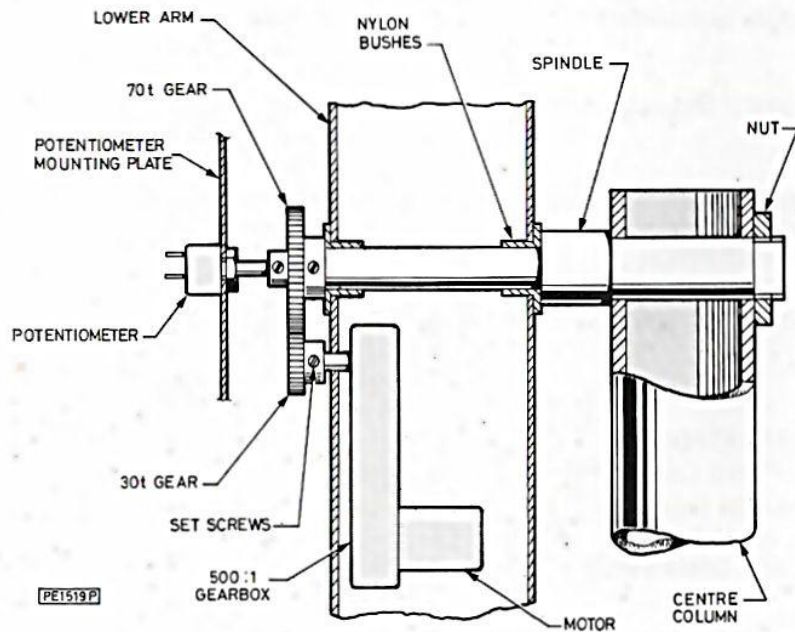


Figura 41 Estrutura mecânica do eixo 1 do robô

O punho do braço robótico (Figura 42) é constituído pelos eixos 3 e 4. Através da combinação dos movimentos destes eixos pode-se obter a elevação e/ou rotação do punho (ver Figura 43).

O eixo 5 corresponde à garra, sendo também controlado por um motor de corrente contínua. Por forma a manter a distribuição de peso do braço, o servomotor encontra-se no braço anterior sendo que a garra é atuada através de um cabo de aço, similar aos usados nos travões de bicicleta. Desta forma é possível controlar o nível de abertura da garra adaptando-se ao tipo de objeto a manipular.

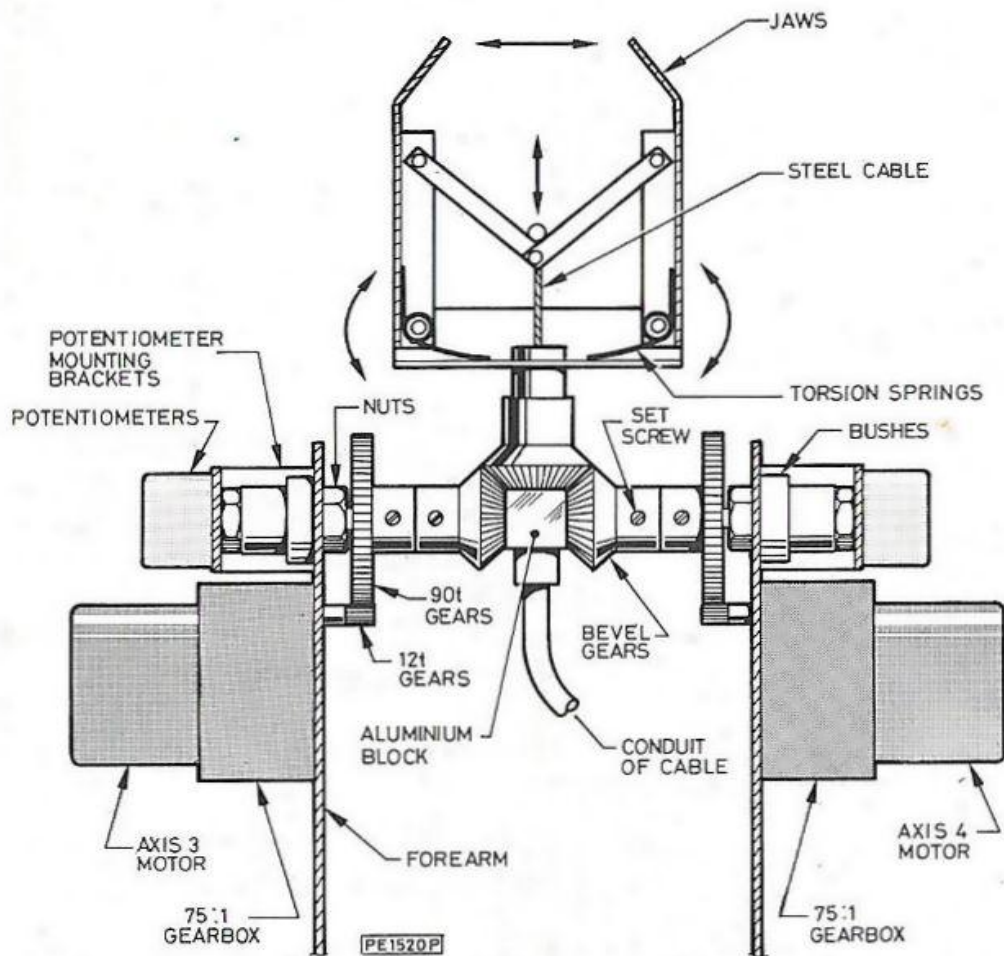


Figura 42 Estrutura mecânica do punho

Na Figura 43 está representada do lado esquerdo a estrutura do punho e à direita tem-se uma tabela onde é possível determinar a elevação (E) e a rotação (R) que o punho pode efetuar atuando a junta 3 e a junta 4.

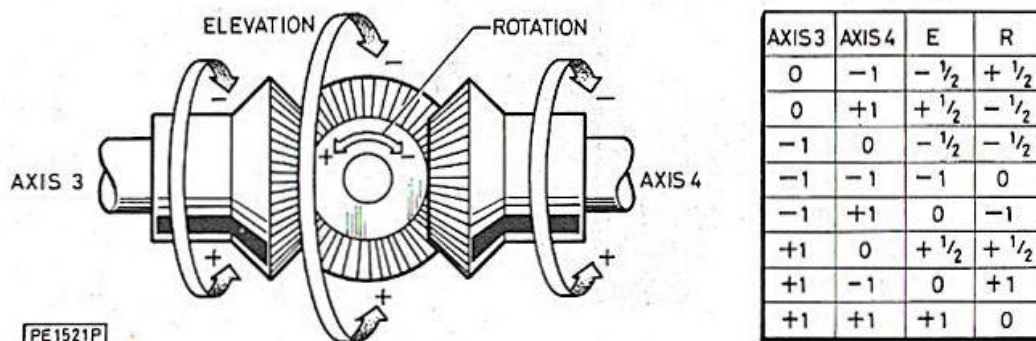


Figura 43 Rotação do punho

O braço robótico MENTOR é composto por um pequeno manipulador (Figura 44) semelhante ao robô MENTOR, que tem como principal objetivo comandar o braço robótico principal através deste elemento, permitindo a atuação do robô de uma forma direta sem qualquer requisito computacional. Este é essencialmente constituído por 6 eixos, conectados a 6 potenciômetros, sendo uma “imitação” à escala reduzida do braço robótico adaptado à mão humana, sendo assim fácil controlar o braço.

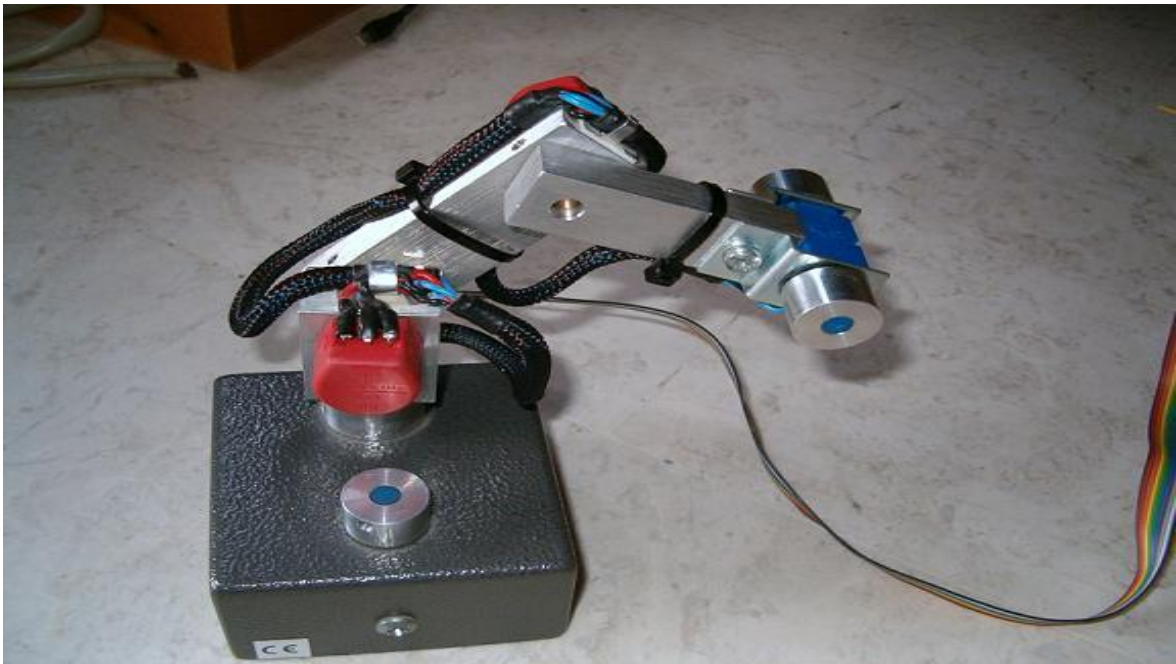


Figura 44 Manipulador do braço robótico MENTOR

4.3. FRIENDLYARM MICRO2440

O *board* FriendlyArm Micro2440 é um módulo de desenvolvimento dotado de um processador *Advanced Risc Machine* (ARM) e com a capacidade de executar um sistema operativo multi-tarefa. Este *board* é composto por uma unidade de processamento mais uma placa de periféricos. O *board* teve o intuito de servir como consola para controlar o robô e de fornecer ao utilizador todos os dados para o controlo do robô, execução e criação de um programa [27].

A unidade de processamento (Figura 45) tem as seguintes características:

- Dimensão da placa: 63 × 52 mm
- CPU: Samsung S3C2440A ARM920T, com frequência nominal de 400 MHz e podendo atingir uma frequência máxima de 533 MHz

- RAM: 64 MB SDRAM, com barramento de 32 bits
- Memória: 1GB NAND Flash e 2 MB NOR Flash com BIOS
- Periféricos: Série, SPI, USB, LCD, CMOS Camara Interface
- Entradas e saídas analógicas
- Indicadores luminosos: 4 LEDs
- Conector de expansão com passo de 2,0 mm
- Conector de *Debug* JTAG: 10 pinos com passo de 2,0 mm
- Sistemas operativos que suporta:
 - Windows CE 5 e Windows CE 6
 - Linux 2.6
 - Android



Figura 45 Unidade de processamento

A *board* dos periféricos (Figura 46) tem as seguintes características:

- Dimensão do *board*: 180 x 130 mm
- EEPROM: 1024 Bytes (I2C)
- *Slot* para cartões de memória SD
- 3 portas RS232, com conector DB9
- 4 portas USB-A *Host* 1.1
- 1 porta USB-B *Device* 1.1

- Conector de saída de áudio de 3,5 mm
- Conector de entrada de áudio mono de 3,5 mm
- Conector de Ethernet RJ-45 10/100M (DM9000)
- RTC: Real Time Clock with battery (CR1220)
- *Beeper*: PWM buzzer
- Conetor de 20 pinos com passo de 2,0 mm para interface com uma câmara CMOS
- Conetor de 41 pinos para displays do FriendlyARM
- Conetor para *board* VGA
- Paineil táctil (*Touch Panel*) resistivo de 4 pinos
- Entradas para o utilizador: 6 *push buttons* e 1 ADC
- Conetor de expansão de E/S (2,0 mm)
- Alimentação de 5 VDC com conector de 3,5 mm de diâmetro e 1,35 mm de profundidade

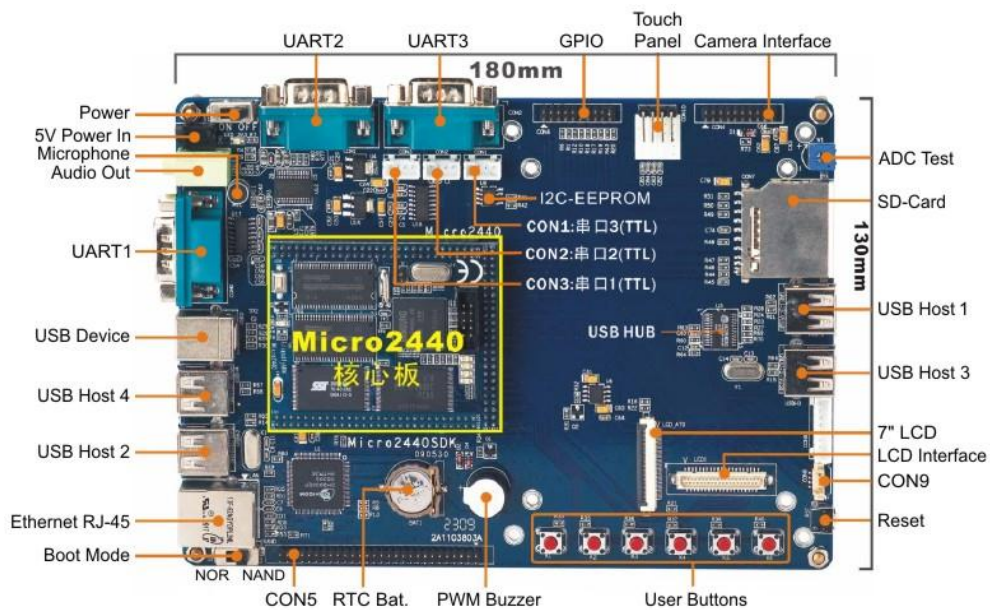


Figura 46 Controlador FriendlyArm Micro2440+SDK Board

O *board* final, isto é a unidade de processamento, o *display* LCD e a placa de periféricos têm o seguinte aspeto (Figura 47).



Figura 47 Controlador FriendlyArm Micro2440+SDK Board com display

4.4. MICROSOFT VISUAL STUDIO

O Microsoft Visual Studio é uma ferramenta de desenvolvimento integrado (em inglês *Integrated Development Environment* - IDE) da Microsoft. É usado para o desenvolvimento de aplicações de console e interfaces gráficas (em inglês *Graphical User Interface* - GUI). Junto com o Windows Forms, pode também ser usado para *sites*, aplicações *web* e serviços *web* em código nativo ou em código suportado pelas plataformas Microsoft Windows, Windows Mobile, Windows CE, .NET, .NET Compact Framework e Microsoft Silverlight.

O Visual Studio suporta diferentes linguagens de programação, sendo estas: C/C++ (Visual C++), C# (Visual C#), Basic (Visual Basic) e F#. Também suporta XML/XSLT, HTML/XHTML, Javascript, CSS, M, Python e Ruby, sendo que para estas três últimas linguagens é necessário a instalação de um pacote de *software* separadamente.

4.5. FONTE DE ALIMENTAÇÃO ATX

As fontes de alimentação *Advanced Technology Extended* (ATX) (ver Figura 48) são as responsáveis por distribuir energia elétrica a todos os componentes do computador. Por isso, uma fonte de qualidade é essencial para manter o bom funcionamento do equipamento.



Figura 48 Fonte de alimentação ATX

Essencialmente, as fontes de alimentação são equipamentos responsáveis por fornecer energia aos dispositivos do computador, convertendo corrente alternada (CA ou em inglês *Alternate Current - AC*) em corrente contínua (CC ou em inglês *Direct Current - DC* ou *Voltage Direct Current - VDC*), numa tensão apropriada para uso em aparelhos eletrônicos.

Nos computadores usa-se um tipo de fonte conhecido como fonte comutada, isto é, um padrão que faz uso de condensadores e bobinas no processo de conversão de energia. A vantagem disso é que há menos geração de calor, já que um mecanismo da fonte simplesmente desativa o fluxo de energia ao invés de dissipar um possível excesso. Além disso, há menor consumo, pois a fonte consegue utilizar praticamente toda a energia que “entra” no dispositivo. Por se tratar de um equipamento que gera campo eletromagnético (já que é capaz de trabalhar com frequências altas), as fontes comutadas devem ser blindadas para evitar interferência em outros aparelhos e no próprio computador.

Na Tabela 6 encontra-se a descrição de cada pino do conector ATX.

Tabela 6 Conector ATX, cores e significados

Tensão	Pino	Cor	Cor	Pino	Tensão
+3,3 V	1			13	+3,3 V
+3,3 V	2			14	-12 V
Terra	3			15	Terra
+5 V	4			16	PS_ON
Terra	5			17	Terra
+5 V	6			18	Terra
Terra	7			19	Terra
<i>Power OK</i>	8			20	-5 V (opcional)
+5 VSB	9			21	+5 V
+12 V	10			22	+5 V
+12 V	11			23	+5 V
+3,3 V	12			24	Terra

Os cinco tipos de tensões de uma fonte ATX são:

- +5 V: utilizado na alimentação de *chips*, como processadores, *chipsets*, módulos de memória e as portas USB;
- -5 V: aplicada em dispositivos periféricos, como ratos e teclados;
- +12 V: usada em dispositivos que contenham motores, como disco rígidos (cujo motor é responsável por girar os discos) e *drives* de CD ou DVD (que possuem motores para abrir a gaveta e para girar o disco);
- -12 V: utilizada na alimentação de barramentos de comunicação, como o antigo *Industry Standard Architecture* (ISA);
- +3,3 V: usada por *chips* (principalmente pelo processador), reduzindo o consumo de energia.

As fontes ATX oferecem a possibilidade de se desligarem via *software*, pois contam com um sinal *Transistor-Transistor Logic* (TTL) chamado *Power Supply On* (PS_ON). Quando o computador está em uso a placa-mãe (*motherboard*) mantém um nível de tensão baixo

para o PS_ON, já quando não em uso o nível de tensão do PS_ON permanece alto. Esse sinal de ativação e desativação pode partir de recursos como:

- Soft On/Off: ativação e desativação da fonte via software;
- Wake-on-LAN: ativação e desativação da fonte via placa de rede;
- Wake-on-Modem: ativação e desativação da fonte via placa de fax modem.

O sinal PS_ON depende da existência do sinal 5 VSB (*Standby*). Esse recurso proporciona ao computador entrar em modo descanso, ou seja, permite que determinados circuitos sejam alimentados quando as tensões em corrente contínua estão suspensas, mantendo ativa apenas a tensão de 5 V. Com isso é possível o computador manter-se ligado mesmo que placa de vídeo ou discos rígidos estejam desativados.

O *Power OK* é um recurso que funciona como uma proteção. A sua função é comunicar com a fonte, a fim de verificar o correto funcionamento, ou seja, operando com tensões aceitáveis para o bom funcionamento e sem riscos de danificar algum componente do computador. Caso esse sinal não exista, ou seja interrompido, o computador geralmente desliga automaticamente.

Para este trabalho é necessário ter duas tensões: 3,3 Volt e 5 Volt para a eletrônica e 12 Volt para alimentar os motores. As fontes de alimentação ATX oferecem estas tensões, têm a vantagem de terem perdas pequenas e custos reduzidos (entre 12€ a 20€). Uma solução para se obter as duas tensões seria o desenvolvimento de *hardware* com base em fontes de tensões comutadas (oferecem perdas pequenas), só que o custo dos componentes, placas de circuitos impresso e corrente que oferecem não se compara às fontes de alimentação ATX.

4.6. UNIDADE DE PROCESSAMENTO

As unidades de processamentos podem ser do tipo microprocessador (μP) ou microcontrolador (μC). A diferença entre os dois tipos é que o microcontrolador possui um microprocessador mais os periféricos, isto é, o microprocessador possui a Unidade Lógica Aritmética (ULA ou, em inglês, *Arithmetic Logic Unit* - ALU), registo de dados e registo de endereços e o microcontrolador, além desses três periféricos, tem mais a memória e periféricos de entrada e saída (E/S ou, em inglês, *Input and Output* - I/O).

Os microprocessadores e os microcontroladores podem ser de arquitetura de conjunto complexo de instruções (*Complex Instruction Set Computing – CISC*) ou conjunto reduzido de instruções (*Reduced Instruction Set Computing – RISC*). A diferença entre CISC e RISC é que, em RISC para conseguir fazer certa instrução complexa tem de se combinar instruções simples e em CISC tem-se uma instrução que faz logo o desejado.

Para este trabalho em questão foi escolhida para unidade de processamento um microcontrolador, por ter memória e periféricos embutidos e preços baixos.

4.6.1. STM32VLDISCOVERY

Para a escolha do microcontrolador que controlará o robô (controle dos *drivers* dos motores, leitura dos potenciômetros e cálculo da cinemática do robô), foram tidos em conta alguns critérios como, o preço, as especificações técnicas (memória, número de E/S e protocolos de comunicações), o conhecimento já adquirido sobre a família do microcontrolador escolhido, que reduz o tempo de desenvolvimento, capacidade de processamento e a facilidade de utilização de sistema operativo multi-tarefa [29].

Estes requisitos colocaram a escolha no microcontrolador ARM CORTEX M3. Para facilitar a utilização deste microprocessador (programação e acesso aos pinos), e porque o custo de integração dos componentes necessários para o utilizar era superior, decidiu-se utilizar a placa de desenvolvimento STM32VLDISCOVERY (Figura 49), que inclui o programador e acesso por pinos 0.1” (2,54 mm) de espaçamento a todos os pinos do microprocessador.

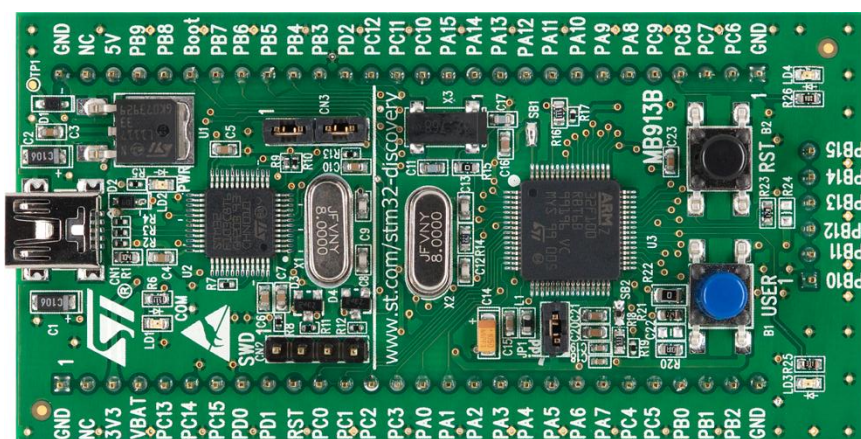


Figura 49 Placa de desenvolvimento STM32VLDISCOVERY

A placa de desenvolvimento é composta por dois microcontroladores, sendo que um deles é utilizado para programar o outro, dois LEDs, dois botões (um de *reset* e outro para qualquer utilização) e cinquenta e um I/Os.

O microcontrolador é o STM32F100RBT6B e tem as seguintes características:

- Processador ARM 32-bit CortexTM-M3 CPU;
- Frequência máxima de 24 MHz;
- 128 kbytes de memória Flash;
- 8 kbytes de SRAM;
- Conversor analógico digital de 12 bits, com 16 canais;
- 3 portos USART;
- 2 interfaces I2C;
- 2 interfaces SPI;
- 12 *Timers*;
- Custo médio de 10 €.

4.7. CONTROLADOR DE MOTORES

Um controlo de motor (ou mais comum *driver* de motor) é um dispositivo, ou um conjunto de dispositivos eletrónicos, que serve para regular o desempenho de um motor elétrico. Um *driver* de motor pode incluir um meio manual ou automático para iniciar e/ou parar o motor, efetuar a seleção para rodar para a frente ou para trás, regular a velocidade, regular ou limitar o binário e proteger contra falhas e sobrecargas.

Cada motor elétrico tem que ter algum tipo de controlo, este pode ter diferentes características e complexidades, isto dependendo da tarefa a que o motor está sujeito. O exemplo mais simples de controlador é um interruptor para ligar um motor a uma fonte de energia. Os *drivers* de motores mais complexos fazem uso de um circuito em H, mais conhecido como ponte H e que é explicado a seguir.

4.7.1. PONTE H

A ponte H (Figura 50) é um circuito eletrônico ou mecânico em forma de H, que permite que uma tensão seja aplicada a uma determinada carga em qualquer direção. Estes circuitos são frequentemente utilizados em robótica ou em outras aplicações onde é necessário colocar um motor de corrente contínua (CC) a girar no sentido dos ponteiros do relógio, no sentido contrário aos ponteiros de relógio, ou pará-lo.

O termo ponte H deriva da representação gráfica típica de tal circuito. Uma ponte H é construída com quatro interruptores de estado sólido (eletrônica) ou mecânico (relé).

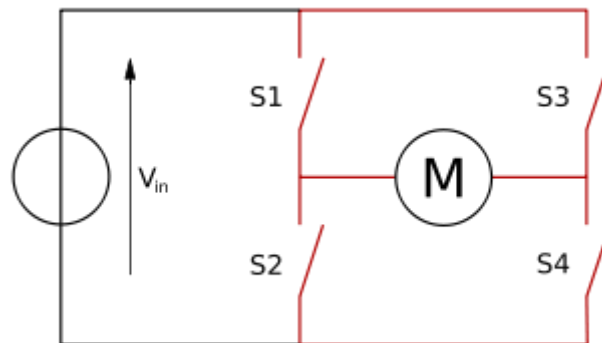


Figura 50 Ponte H

A ponte H, como já referido anteriormente, é usada para inverter o sentido de rotação do motor, mas também pode ser usada para travar o motor. Isto é possível efetuar através da combinação que se dá aos interruptores S1, S2, S3 e S4. Quando os interruptores S1 e S4 estão fechados e os interruptores S2 e S3 abertos (Figura 51), é aplicada uma tensão positiva ao motor e este gira no sentido dos ponteiros de relógio.

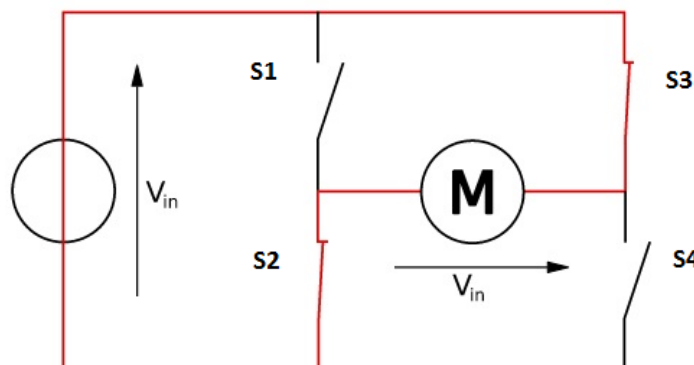


Figura 51 Ponte H – condução positiva, motor gira no sentido dos ponteiros do relógio

Se se inverter o estado dos quatro interruptores (S1 e S3 abertos e S2 e S4 fechados) (Figura 52), o motor gira em sentido contrário aos ponteiros de relógio, isto porque é aplicada ao motor uma tensão invertida, permitindo esta operação.

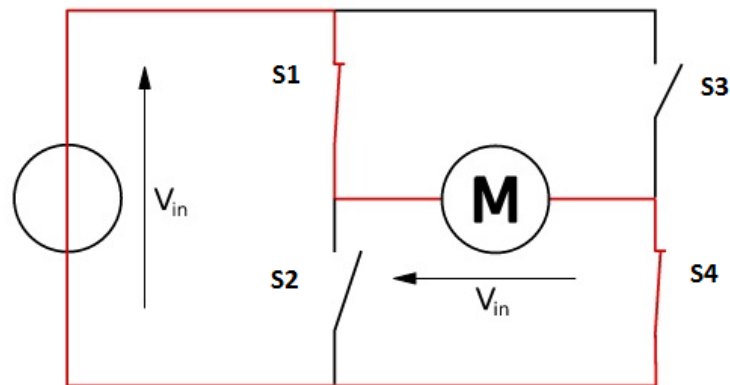


Figura 52 Ponte H – condução negativa, motor gira no sentido contrário aos ponteiros do relógio

Para travar o motor temos duas opções, uma quando se tem S1 e S3 fechados e S2 e S4 abertos (travar o motor à alimentação) (Figura 53 da esquerda) e a outra opção é quando se tem S1 e S3 abertos e S2 e S4 fechados (travar motor à massa) (Figura 53 da direita).

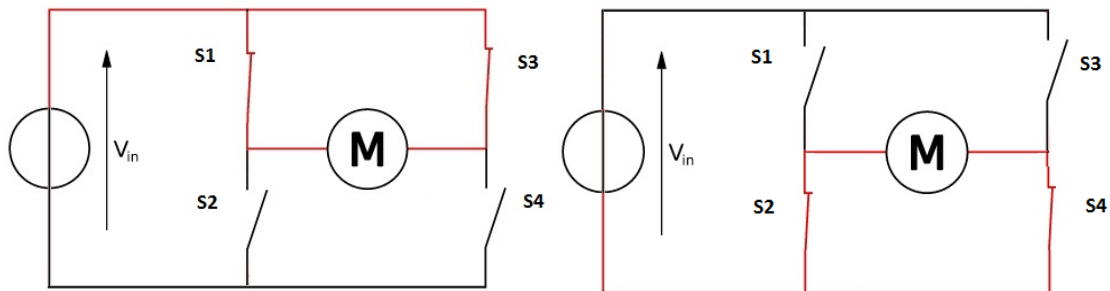


Figura 53 Ponte H – motor travado, à esquerda através da fonte, à direita através da massa

Na ponte H nunca se pode ter S1 e S2 fechados e/ou S3 e S4 ao mesmo tempo, porque provocam um curto-circuito na fonte de alimentação de entrada. Esta situação é conhecida em termos técnicos como *shoot-through*.

Na Tabela 7 tem-se, de forma resumida, o estado do motor consoante o estado dos quatro interruptores.

Tabela 7 Ponte H – estado do motor consoante os interruptores

Interruptores				Estado do motor
S1	S2	S3	S4	
<i>Off</i>	<i>Off</i>	<i>Off</i>	<i>Off</i>	Destravado
<i>On</i>	<i>Off</i>	<i>On</i>	<i>Off</i>	Roda no sentido dos ponteiros do relógio
<i>Off</i>	<i>On</i>	<i>Off</i>	<i>On</i>	Roda no sentido contrário aos ponteiros de relógio
<i>On</i>	<i>Off</i>	<i>On</i>	<i>Off</i>	Trava
<i>Off</i>	<i>On</i>	<i>Off</i>	<i>On</i>	Trava
<i>On</i>	<i>On</i>	<i>Off</i>	<i>Off</i>	<i>Shoot-Through</i>
<i>Off</i>	<i>Off</i>	<i>On</i>	<i>On</i>	<i>Shoot-Through</i>
<i>On</i>	<i>On</i>	<i>On</i>	<i>On</i>	<i>Shoot-Through</i>

As pontes H de estado sólido são normalmente construídas com transístores, podendo estes ser do tipo PNP, BJT e MOSFET. Os modelos mais eficientes de *drivers* de motores usam MOSFETs, porque a resistência ON (resistência medida quando estão a conduzir) apresenta valores baixos, fazendo com que o dispositivo tenha perdas menores.

4.7.2. ST VNH5180

Para a escolha do *driver* do motor foi necessário que este tivesse os seguintes requisitos:

- Fácil implementação
- Baixo preço
- Número reduzido de sinais para controlo da ponte H
- Corrente elétrica suportada igual ou superior a 400 mA
- Tensão elétrica suportada igual ou superior a 12 Volt
- Baixa resistência de condução

O *driver* ST VN5180 tem como principais características:

- Tensão elétrica máxima do motor 41 Volt
- Corrente elétrica máxima de 8 A
- Proteção contra sobrecarga
- Frequência máxima de PWM - 20 kHz
- Resistência de condução (*per leg*) de 180 mΩ

Este *driver* precisa de três sinais para controlar o movimento do motor, dois sinais para ativar a ponte H, e um sinal para a modelação da largura do impulso que vai para o motor. A Tabela 8 mostra a tabela de verdade das condições normais de funcionamento do *driver* utilizado.

Tabela 8 Tabela de verdade do ST VN5180

INA	INB	ENA	ENB	OUTA	OUTB	CS	<i>Operating Mode</i>
1	1	1	1	H	H	<i>High Impedance</i>	<i>Brake to Vcc</i>
	L					<i>Clockwise (CW)</i>	
0	1			L	H	<i>I_{sense} = I_{out} / k</i>	<i>Counterclockwise (CCW)</i>
	0				L	<i>High Impedance</i>	<i>Brake to GND</i>

5. IMPLEMENTAÇÃO DO SISTEMA

Neste capítulo são explicados, de forma sucinta, todos os passos realizados na implementação do trabalho. A primeira secção diz respeito aos cálculos matemáticos, relativos à cinemática e o planeamento de trajetórias.

Depois encontra-se uma secção onde é demonstrado e explicado o *hardware* proposto para o controlo do braço robótico MENTOR, isto porque o *hardware* que este braço robótico possuía era muito rudimentar.

Por fim, há uma secção referente ao *software* desenvolvido, na consola e *firmware* na placa de controlo do robô.

5.1. CÁLCULO NUMÉRICO

Esta secção apresenta todos os cálculos necessários para obter ângulos para posições desejadas e vice-versa. Para se obter as equações da cinemática direta optou-se pelo uso do formalismo de Denavit-Hartenberg, por ser o método mais recomendado e simples para robôs com mais de três juntas. A cinemática inversa, como já foi referido, só é possível

através de desenvolvimento trigonométrico. Mas antes de se começar a calcular as expressões matemáticas da cinemática direta, tem-se que começar a colocar um sistema de eixos cartesianos (x,y,z) em cada junta seguindo as regras da convecção de Denavit-Hartenberg.

O primeiro passo é localizar e identificar o sistema de coordenadas da origem. O eixo z_0 é paralelo ao eixo de rotação da junta 1, o eixo x_0 fica com a orientação de metade do ângulo máximo da junta 1 e por fim coloca-se o eixo y_0 segundo a regra da mão direita. Na Figura 54 mostra-se o sistema de coordenadas da origem.

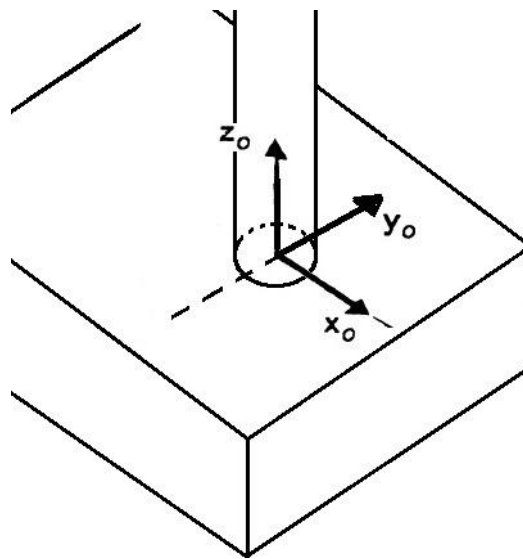


Figura 54 Sistema de coordenadas da origem

O eixo z_i é sempre paralelo ao eixo de rotação da junta. O sistema de eixos 1 foi transladado para o topo do primeiro elo (translação em 185 mm medindo em z_0), em que $\theta_1 = 0^\circ$ é coincidente com o sistema de coordenadas da origem. Na Figura 55 pode-se observar onde se encontra este sistema de coordenadas.

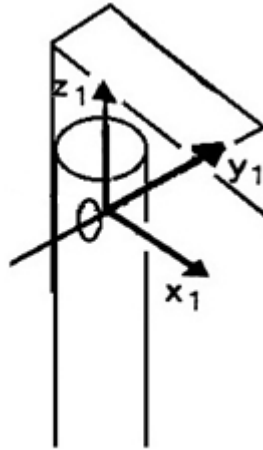


Figura 55 Sistema de coordenadas para a junta 1

O sistema de coordenadas da junta 2 tem a mesma origem que o sistema de coordenadas da junta 1, só que aqui existe uma rotação positiva de 90° em torno do eixo x_1 . A Figura 56 representa o sistema de coordenadas para a junta 2.

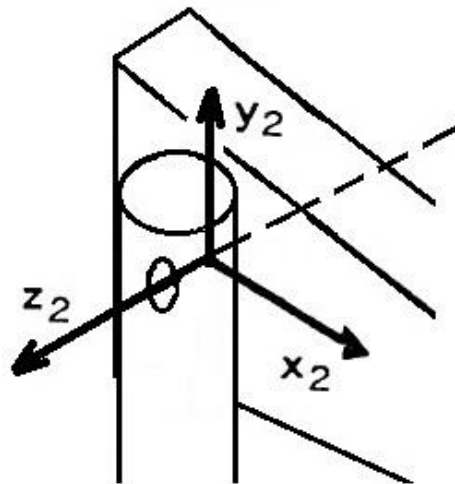


Figura 56 Sistema de coordenadas para a junta 2

Para a junta 3 o sistema de coordenadas localiza-se como mostra a Figura 57 e tem a mesma orientação que o sistema de coordenada da junta 2 e uma translação de 165 mm em x_2 .

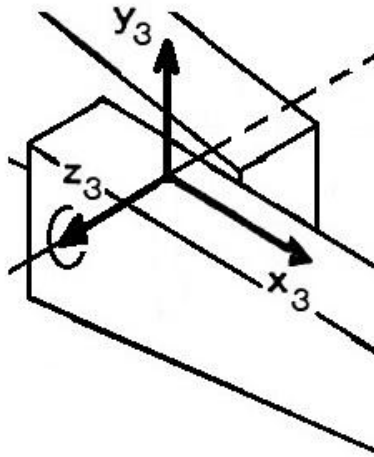


Figura 57 Sistema de coordenadas para a junta 3

O sistema de coordenadas da junta 4, foi analisado com atenção de modo a se conseguir uma relação entre o sistema de coordenadas da junta 3 e o sistema de coordenadas da garra, a solução para este sistema de coordenadas encontra-se representado na Figura 58. Este sistema de coordenadas sofreu uma rotação negativa de 90° em torno do eixo z_3 , em relação ao sistema de coordenadas da junta 3, e sofrendo depois uma rotação negativa de 90° em torno de x_4 o que vai resultar no sistema de coordenadas da garra.

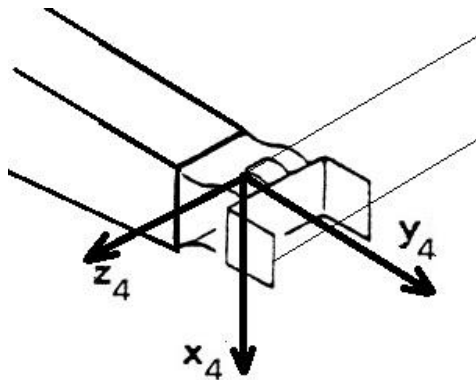


Figura 58 Sistema de coordenadas para a junta 4

Por fim, o último passo é localizar os sistemas de coordenadas para garra, em que o eixo de abertura e fecho da garra é paralelo ao eixo y e o eixo z é paralelo ao eixo da orientação da garra, segundo a regra da mão direita. A Figura 59 ilustra o sistema de coordenadas da garra.

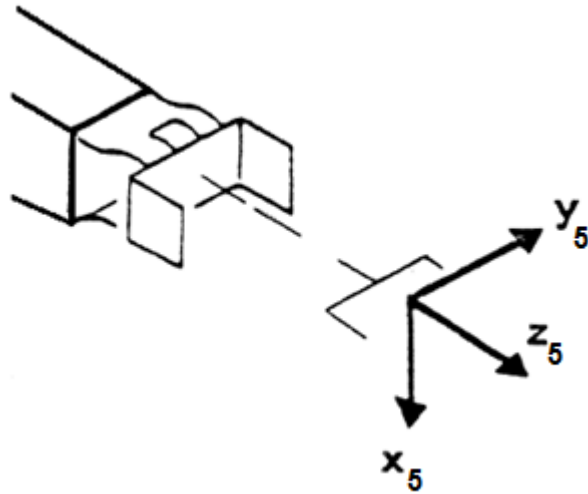


Figura 59 Sistema de coordenadas para a garra

Por fim, na Figura 60, apresentam-se os vários sistemas de coordenadas, sendo que estes seguem todas as regras da convenção de Denavit-Hartenberg.

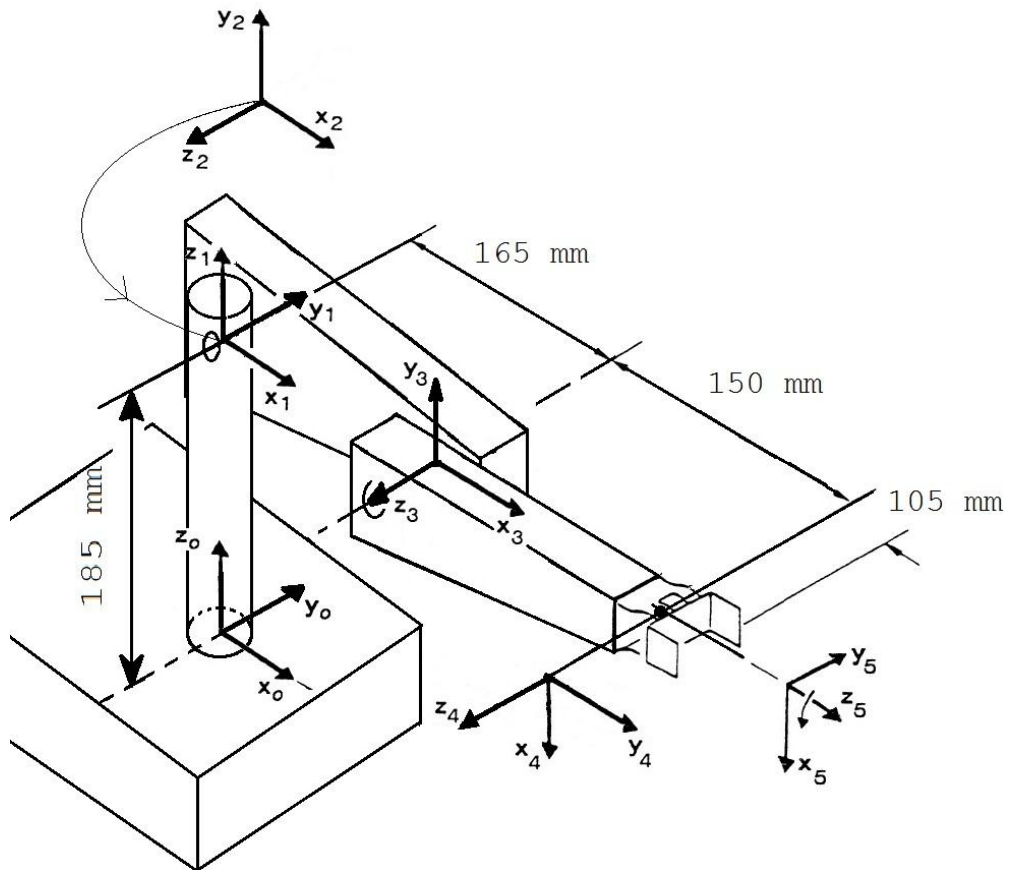


Figura 60 Sistema de coordenadas do braço robótico

5.1.1. CINEMÁTICA DIRETA SEGUNDO DENAVIT-HARTENBERG

Depois de se ter colocado os eixos de coordenadas em cada junta (secção 5.1.), procedese à obtenção da matriz ${}^{i-1}A_i$ que contém a translação e rotação da junta i em relação à junta $i-1$. Quando se obtiverem as matrizes procede-se à sua multiplicação originando a matriz T , que consiste na translação, rotação e posição da garra em relação à origem. Para a obtenção da matriz ${}^{i-1}A_i$ é necessário obter as seguintes variáveis:

- θ_i - ângulo formado entre x_{i-1} e x_i medido em z_{i-1}
- α_{i-1} - ângulo formado entre z_{i-1} e z_i , e medido em x_i
- a_{i-1} - distância ao longo de x_i desde a intersecção de x_i com z_{i-1}
- d_i - distância ao longo de z_{i-1} desde a intersecção de x_i com z_{i-1}

Na equação (51) tem-se a representação da matriz ${}^{i-1}A_i$ e das variáveis anteriormente referidas que compõem a respetiva matriz.

$${}^{i-1}A_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{i-1} \\ \sin\theta_i \cdot \cos\alpha_{i-1} & \cos\theta_i \cdot \cos\alpha_{i-1} & -\sin\alpha_{i-1} & -\sin\alpha_{i-1} \cdot d_i \\ \sin\theta_i \cdot \sin\alpha_{i-1} & \cos\theta_i \cdot \sin\alpha_{i-1} & \cos\alpha_{i-1} & \cos\alpha_{i-1} \cdot d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (51)$$

5.1.1.1. Sistema de coordenadas da junta 1 visto pelo sistema de coordenadas da junta 0

Na Tabela 9 encontram-se de forma resumida as rotações e translações que são feitas do sistema de coordenadas da origem para o sistema de coordenadas 1.

Tabela 9 Translação e rotação da junta 1, vista pela junta 0

Rotação em Z	θ_i	θ_1
Translação em Z	d_i	185
Translação em X	a_{i-1}	0
Rotação em X	α_{i-1}	0°

Através dos valores anteriormente referidos obtém-se a equação (52), sendo esta referente à matriz de rotação e translação entre os dois primeiros sistemas de coordenadas.

$${}^0A_1 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 185 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (52)$$

5.1.1.2. Sistema de coordenadas da junta 2 visto pelo sistema de coordenadas da junta 1

Na Tabela 10 tem-se a rotação e translação da junta 2 vista pela junta 1.

Tabela 10 Translação e rotação da junta 2, vista pela junta 1

Rotação em Z	θ_i	θ_2
Translação em Z	d_i	0
Translação em X	a_{i-1}	0
Rotação em X	α_{i-1}	90°

A equação (53) corresponde à matriz ${}^{i-1}A_i$ da junta 2 vista pela junta 1.

$${}^1A_2 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (53)$$

5.1.1.3. Sistema de coordenadas da junta 3 visto pelo sistema de coordenadas da junta 2

A Tabela 11 mostra as translações e rotações que a junta 2 sofre para originar o sistema de coordenadas da junta 3.

Tabela 11 Translação e rotação da junta 3, vista pela junta 2

Rotação em Z	θ_i	θ_3
Translação em Z	d_i	0
Translação em X	a_{i-1}	165
Rotação em X	α_{i-1}	0°

Na equação (54) tem-se representado a matriz ${}^{i-1}A_i$ em relação à junta 2 e 3.

$${}^2A_3 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & 165 \\ \sin\theta_3 & \cos\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (54)$$

5.1.1.4. Sistema de coordenadas da junta 3 visto pelo sistema de coordenadas da junta 4

Na Tabela 12 tem-se representado as translações e rotações que a junta 3 sofre para originar o sistema de coordenadas da junta 4. A rotação em z , como já tinha sido referido na secção 5.1, sofreu uma rotação de -90° em torno do eixo z para melhor rotação e translação do sistema de coordenadas da junta 4 para o sistema de coordenadas da garra.

Tabela 12 Translação e rotação da junta 4, vista pela junta 3

Rotação em Z	θ_i	$\theta_4 - 90^\circ$
Translação em Z	d_i	150
Translação em X	a_{i-1}	0
Rotação em X	α_{i-1}	0°

A equação (55) refere-se à matriz de rotação e translação da junta 4 vista pela junta 3.

$${}^3A_4 = \begin{bmatrix} \cos(\theta_4 - 90^\circ) & -\sin(\theta_4 - 90^\circ) & 0 & 150 \\ \sin(\theta_4 - 90^\circ) & \cos(\theta_4 - 90^\circ) & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (55)$$

5.1.1.5. Sistema de coordenadas da garra visto pelo sistema de coordenadas da junta 4

Por fim tem-se o sistema de coordenadas da garra vista pela junta 4. A Tabela 13 mostra as rotações e translações que a junta 4 sofre para originar o sistema de coordenadas da garra.

Tabela 13 Translação e rotação da junta 4, vista pela junta 5

Rotação em Z	θ_i	θ_5
Translação em Z	d_i	105
Translação em X	a_{i-1}	0
Rotação em X	α_{i-1}	-90°

A equação (56) é referente à última matriz de rotações e translações da junta i vista pela junta $i-1$.

$${}^4A_5 = \begin{bmatrix} \cos\theta_5 & -\sin\theta_5 & 0 & 0 \\ 0 & 0 & 1 & 105 \\ -\sin\theta_5 & \cos\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (56)$$

A Tabela 14 é um resumo das rotações e translações para cada uma das situações anteriormente referidas.

Tabela 14 Orientações e rotações em torno do eixo x e z

Junta	θ_i	α_{i-1}	d_i	a_{i-1}
1	θ_1	0	0	0
2	θ_2	90°	185	0
3	θ_3	0	0	165
4	θ_4-90	0	150	0
5	θ_5	-90°	105	0

Como já tinha sido referido, depois da obtenção das matrizes ${}^{i-1}A_i$ procede-se à multiplicação destas para a originar a matriz \mathbf{T} . Na equação (57) tem-se a multiplicação e na equação (58) o significado de cada termo da matriz \mathbf{T} , sendo que n , s e a são as rotações que o braço robótico faz em relação à origem.

$$T = {}^0A_1 \times {}^1A_2 \times {}^2A_3 \times {}^3A_4 \times {}^4A_5 \quad (57)$$

$$T = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (58)$$

Da expressão (58) retiram-se as expressões (59), (60) e (61), referentes à posição do robô para a posição das juntas conhecidas.

$$p_x = \cos(\theta_1)[165\cos(\theta_2) + 150\cos(\theta_2 + \theta_3) - 105\sin(\theta_2 + \theta_3 + \theta_4 - 90)] \quad (59)$$

$$p_y = \sin(\theta_1)[165\cos(\theta_2) + 150\cos(\theta_2 + \theta_3) - 105\sin(\theta_2 + \theta_3 + \theta_4 - 90)] \quad (60)$$

$$p_z = 185 + 165\sin(\theta_2) + 150\sin(\theta_2 + \theta_3) + 105\cos(\theta_2 + \theta_3 + \theta_4 - 90) \quad (61)$$

5.1.2. CINEMÁTICA INVERSA

A cinemática inversa é um dos processos mais importantes para o controlo de um robô, pois o utilizador na maioria das vezes escolhe a posição final do atuador do robô no espaço e não o valor que as juntas tomam.

O cálculo da cinemática inversa é complexo, porque não existe nenhuma fórmula matemática para calcular o valor dos ângulos das juntas para uma posição final para robôs com mais de dois graus de liberdade. No Apêndice A encontra-se um formulário com algumas expressões da trigonometria. A convenção de Denavit-Hartenberg é um método matemático que permite calcular facilmente o valor da posição final através dos ângulos que as juntas tomam, para robôs com mais de dois graus de liberdade. A seguir vai-se demonstrar e explicar como é que se calcula a cinemática inversa para o braço robótico MENTOR.

Na cinemática inversa do robô MENTOR há quatro tipos de configuração que o braço robótico pode tomar, que são:

- Braço direito para baixo
- Braço direito para cima
- Braço esquerdo para baixo
- Braço esquerdo para cima

As configurações braço direito, braço esquerdo dependem dos ângulos das juntas de base; se o ângulo total da base for inferior a 180° não se consegue ter as duas configurações; se o ângulo da junta da base for, por exemplo, 360° podem-se ter as duas configurações para qualquer ponto. O braço robótico MENTOR possui um ângulo total de 210° na junta da base (primeira junta) o que significa que só se consegue obter as duas configurações em certos pontos da área de trabalho. A segunda junta do braço tem um ângulo total de 180° , o que faz com que seja muito complicado de se obter as configurações de braço direito e braço esquerdo, logo a configuração que se vai usar é a configuração braço esquerdo, porque é a configuração normal do braço robótico MENTOR.

Na Figura 61 mostra-se que só nos 30° de cada extremo ($210^\circ - 180^\circ = 30^\circ$) se pode utilizar a configuração braço direito (*right arm*) e braço esquerdo (*left arm*), porque é nos pontos dessa área que se consegue inverter os valores de uma dada posição em x e y , e por outro lado estarem dentro da área de trabalho do braço robótico.

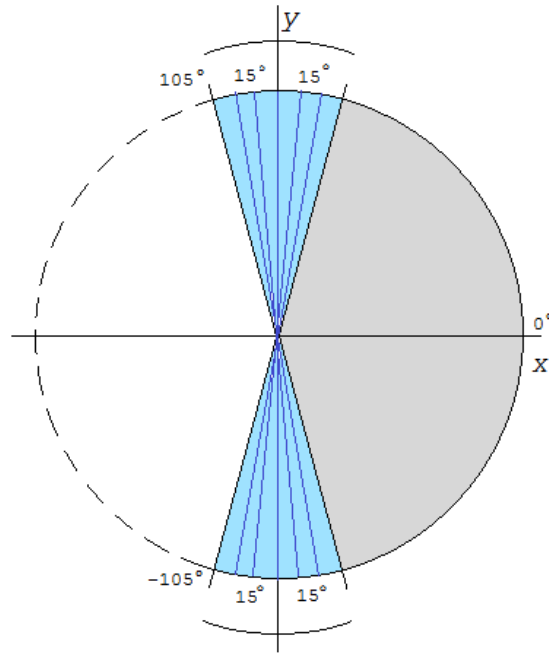


Figura 61 Área em que o braço robótico permite a configuração braço direito e braço esquerdo

Resumindo o que foi dito anteriormente só em pontos muito específicos é que se consegue obter no braço robótico MENTOR a configuração braço esquerdo e braço direito.

De forma a simplificar os cálculos da cinemática inversa optou-se por não introduzir a rotação da garra (*roll*) ou θ_5 , sendo que esta variável fica ao encargo do utilizador, para ainda simplificar mais os cálculos. O valor de θ_5 depende do valor de θ_4 , o que significa que quando se obtiver o valor de θ_4 para uma dada posição escolhida, vai-se alcançar o valor de θ_5 para o utilizador escolher a orientação da garra.

Para se calcular o valor de θ_1 é simples, visto que este depende de três elementos, da abcissa e ordenada da posição desejada e do *offset* das juntas, porque este valor só depende do plano xOy , como se pode verificar na Figura 62. O braço robótico MENTOR não tem *offset*, isto é, a distância entre o centro do primeiro elo com o segundo é a mesma distância entre o centro do segundo elo com o terceiro.

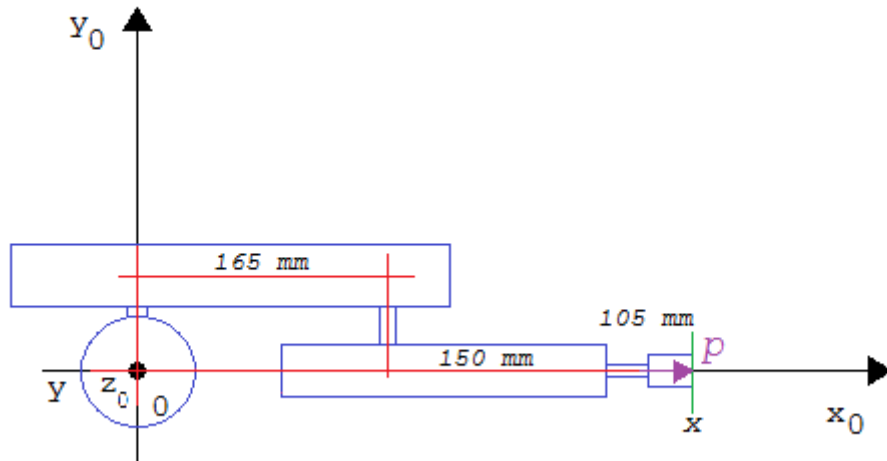


Figura 62 Esquema do braço robótico MENTOR, vista de topo

O ângulo da junta 1 (θ_1) é obtido pelo arco tangente de círculo completo ($atan2$). Na equação (62) tem-se θ_1 para a configuração braço esquerdo e na equação (63) para a configuração braço direito.

$$\theta_1^L = atan2\left(\frac{y}{x}\right) \quad (62)$$

$$\theta_1^R = atan2\left(\frac{y}{x}\right) \pm 180^\circ \quad (63)$$

Com todas estas simplificações só se precisa de calcular θ_2 , θ_3 e θ_4 e estes ângulos vão depender do valor de z da posição escolhida, do vetor p_1 (vetor de x com y) e da orientação³ do último elo em relação à base, como se mostra na Figura 63. Também nessa figura está representado a configuração *Upper Elbow* ou *Above Arm* (representado a verde) e *Lower Elbow* ou *Below Arm* (representado a preto).

Para calcular o valor de θ_2 , θ_3 e θ_4 através de uma dada posição, tem que se voltar a calcular a matriz da translação e rotação da cinemática direta através da convenção de Denavit-Hartenberg, só que o primeiro eixo começa na junta 2 e vai até à garra e não se contabiliza a junta 5.

³ Se não for dada a orientação do último elo à base (ângulo ϕ), para a cinemática inversa, tem-se um número infinito de valores para θ_2 , θ_3 e θ_4 , isto é, cada um dos três ângulos depende de um ou dos dois outros ângulos, fazendo com que se tenha uma infinidade de condições para uma posição desejada.

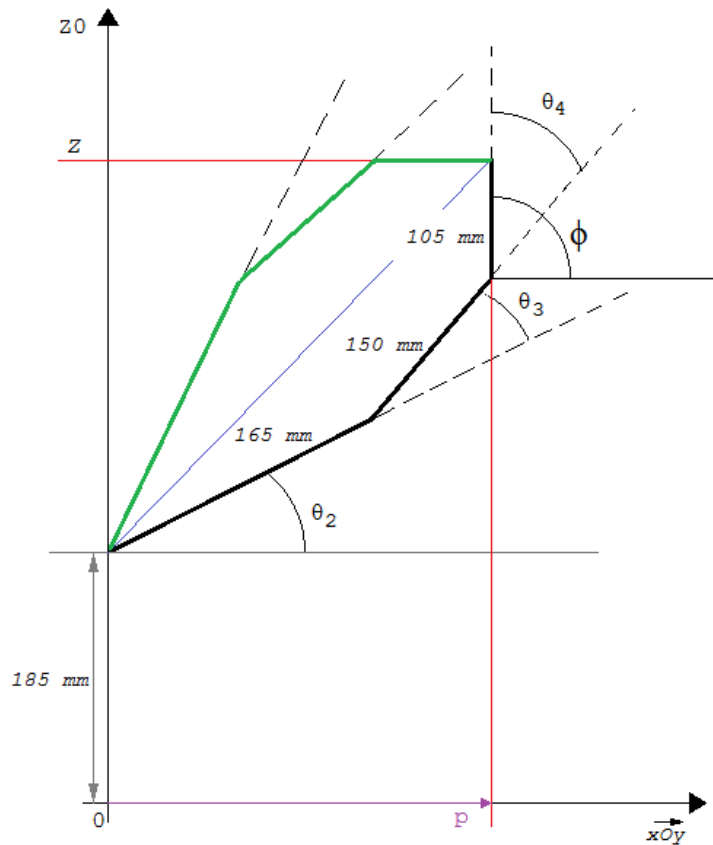


Figura 63 Esquema do braço robótico MENTOR, vista de lado

O vetor p é calculado da seguinte forma:

$$p = \sqrt{x^2 + y^2} \quad (64)$$

Na Tabela 15 encontra-se resumido o valor das variáveis a usar para o cálculo da cinemática direta segundo a convenção de Denavit-Hartenberg para o esquema da Figura 63.

Tabela 15 Cinemática direta para as juntas 2, 3 e 4 segundo Denavit-Hartenberg

i	θ_i	α_{i-1}	d_i	a_{i-1}
2	θ_2	0	0	0
3	θ_3	0	0	165
4	θ_4	0	0	150
5	0	0	0	105

A equação (65) corresponde à matriz \mathbf{T} da cinemática direta para o esquema anteriormente referido.

$$T_2^5 = \begin{bmatrix} \cos_{234} & -\sin_{234} & 0 & L_1 \cos_2 + L_2 \cos_{23} + L_3 \cos_{234} \\ \sin_{234} & \cos_{234} & 0 & L_1 \sin_2 + L_2 \sin_{23} + L_3 \sin_{234} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (65)$$

De forma a minimizar o tamanho da matriz \mathbf{T} anteriormente referida, usaram-se variáveis cujo valor é:

- $L_0 = 185$ mm
- $L_1 = 165$ mm
- $L_2 = 150$ mm
- $L_3 = 105$ mm
- $\cos_2 = \cos(\theta_2)$
- $\cos_{23} = \cos(\theta_2 + \theta_3)$
- $\cos_{234} = \cos(\theta_2 + \theta_3 + \theta_4)$
- $\sin_2 = \sin(\theta_2)$
- $\sin_{23} = \sin(\theta_2 + \theta_3)$
- $\sin_{234} = \sin(\theta_2 + \theta_3 + \theta_4)$

Através da matriz da equação (65) obtêm-se as equações (66), (67) e (68), referentes à posição do robô no plano zOp , sendo p o plano formado pelos eixos de coordenadas xOy .

$$p = \sqrt{x^2 + y^2} = L_1 \cos(\theta_2) + L_2 \cos(\theta_2 + \theta_3) + L_3 \cos(\theta_2 + \theta_3 + \theta_4) \quad (66)$$

$$q = z - L_0 = L_1 \sin(\theta_2) + L_2 \sin(\theta_2 + \theta_3) + L_3 \sin(\theta_2 + \theta_3 + \theta_4) \quad (67)$$

$$\phi = \theta_2 + \theta_3 + \theta_4 = \text{atan2}(\sin(\theta_2 + \theta_3 + \theta_4), \cos(\theta_2 + \theta_3 + \theta_4)) \quad (68)$$

Assumindo que são dadas as coordenadas cartesianas, x , y , z e o ângulo de orientação ϕ^4 , e se quer encontrar expressões analíticas para os ângulos das juntas θ_2 , θ_3 e θ_4 . Nesse caso, se se substituir a expressão (68) em (66) e (67) pode-se eliminar θ_4 obtendo duas equações em θ_2 e θ_3 , dadas por:

⁴ Ângulo de orientação é obtido através da soma dos ângulos formadores pela junta 2, 3 e 4, ou pelo ângulo que a junta 4 tem em relação ao sistema de coordenadas da base.

$$p - L_3 \cos(\phi) = L_1 \cos(\theta_2) + L_2 \cos(\theta_2 + \theta_3) \quad (69)$$

$$q - L_3 \sin(\phi) = L_1 \sin(\theta_2) + L_2 \sin(\theta_2 + \theta_3) \quad (70)$$

Nas equações (69) e (70) tem-se do lado esquerdo os membros que se conhecem e do lado direito as incógnitas a calcular.

$$pl = p - L_3 \cos(\phi) = \sqrt{x^2 + y^2} - L_3 \cos(\phi) \quad (71)$$

$$ql = q - L_3 \sin(\phi) = z - L_0 - L_3 \sin(\phi) \quad (72)$$

Com as expressões (71) e (72), consegue-se obter um “manipulador RR”, e assim conseguem-se retirar as equações da cinemática inversa para este tipo de robô. Na Figura 64 ilustra-se a configuração que será usada para o cálculo da cinemática inversa. O ângulo de orientação ϕ faz com que seja possível eliminar o ângulo θ_4 e o elo L_3 , restando só os ângulos θ_2 e θ_3 , para facilitar no cálculo da cinemática inversa.

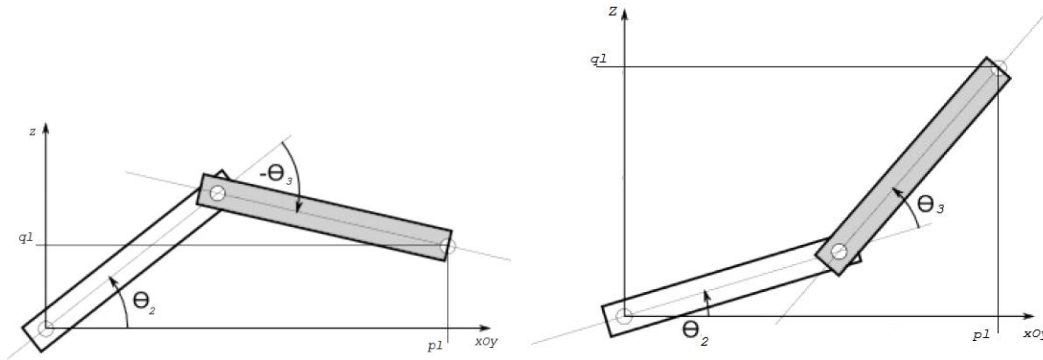


Figura 64 Configuração *Upper Elbow* à esquerda e *Lower Elbow* à direita

Através da cinemática direta retiram-se as expressões:

$$pl = L_1 \cos(\theta_2) + L_2 \cos(\theta_2 + \theta_3) = \sqrt{x^2 + y^2} - L_3 \cos(\phi) \quad (73)$$

$$ql = L_1 \sin(\theta_2) + L_2 \sin(\theta_2 + \theta_3) = z - L_0 - L_3 \sin(\phi) \quad (74)$$

A hipotenusa da configuração planar RR é obtida por:

$$pl^2 + ql^2 = L_1^2 + L_2^2 + 2L_1L_2[\cos(\theta_2) \cdot \cos(\theta_2 + \theta_3) + \sin(\theta_2) \cdot \sin(\theta_2 + \theta_3)] \quad (75)$$

Em seguida, usa-se as seguintes igualdades:

$$\cos(x \pm y) = \cos(x) \cdot \cos(y) \mp \sin(x) \cdot \sin(y) \quad (76)$$

$$\sin(x \pm y) = \sin(x) \cdot \cos(y) \pm \cos(x) \cdot \sin(y) \quad (77)$$

Portanto, obtém-se a expressão:

$$\begin{aligned}
pl^2 + ql^2 &= L_1^2 + L_2^2 + 2L_1L_2[\cos(\theta_2) \cdot (\cos(\theta_2) \cdot \cos(\theta_3) - \sin(\theta_2) \cdot \sin(\theta_3)) + \\
&\quad \sin(\theta_2) \cdot (\sin(\theta_3) \cdot \cos(\theta_3) + \cos(\theta_2) \cdot \sin(\theta_3))] \\
&= L_1^2 + L_2^2 + 2L_1L_2[\cos^2(\theta_2) \cdot \cos(\theta_3) + \sin^2(\theta_3) \cdot \cos(\theta_3)] \\
&= L_1^2 + L_2^2 + 2L_1L_2 \cdot \cos(\theta_3)
\end{aligned} \tag{78}$$

Através da expressão (78) retira-se a equação:

$$\cos(\theta_3) = \frac{pl^2 + ql^2 - L_1^2 - L_2^2}{2L_1L_2} \tag{79}$$

A partir daqui, pode-se obter o ângulo diretamente usando a função *arco cosseno*, no entanto esta função é muito imprecisa para pequenos ângulos. A maneira típica de evitar esta imprecisão é converter de forma a poder utilizar a função *atan2*. Pela trigonometria sabe-se que o quadrado do *seno* mais o quadrado do *cosseno* de qualquer ângulo é sempre igual a um (expressão (80)).

$$\begin{aligned}
\cos^2(\theta_3) + \sin^2(\theta_3) &= 1 \\
\sin(\theta_3) &= \pm\sqrt{1 - \cos(\theta_3)^2}
\end{aligned} \tag{80}$$

Através das expressões (79) e (80) obtém-se a expressão (81), onde é possível obter as configurações *Lower Elbow* e *Upper Elbow*.

$$\begin{aligned}
\theta_3 &= \text{atan2}(\sin(\theta_3), \cos(\theta_3)) \\
\theta_3 &= \text{atan2}\left(\pm\sqrt{1 - \cos(\theta_3)^2}, \cos(\theta_3)\right) \\
\theta_3 &= \text{atan2}\left(\pm\sqrt{1 - \left[\frac{pl^2 + ql^2 - L_1^2 - L_2^2}{2L_1L_2}\right]^2}, \frac{pl^2 + ql^2 - L_1^2 - L_2^2}{2L_1L_2}\right)
\end{aligned} \tag{81}$$

A configuração *Upper Elbow* é obtida com o sinal negativo na expressão (80) e a *Lower Elbow* com o sinal positivo.

Para se obter θ_2 reescreveu-se as equações não-lineares originais usando uma mudança de variável com ajuda do esquema Figura 65, obtém-se as seguintes equações:

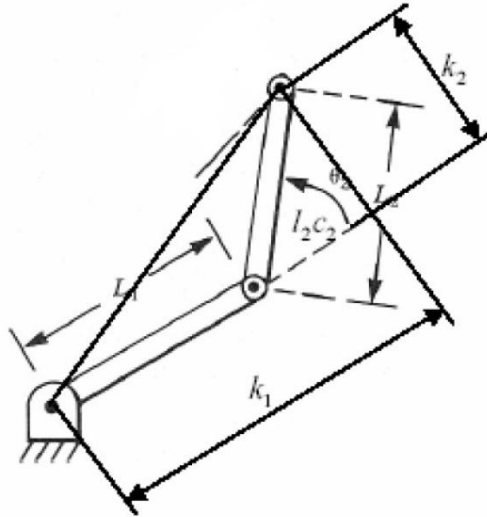


Figura 65 Esquemático para o cálculo de θ_2

$$pl = L_1 \cos(\theta_2) + L_2 \cos(\theta_2 + \theta_3) \quad (82)$$

$$ql = L_1 \sin(\theta_2) + L_2 \sin(\theta_2 + \theta_3) \quad (83)$$

$$pl = k_1 \cos(\theta_2) - k_2 \sin(\theta_2) \quad (84)$$

$$ql = k_1 \sin(\theta_2) + k_2 \cos(\theta_2) \quad (85)$$

onde

$$k_1 = L_1 + L_2 \cos(\theta_2) \quad (86)$$

$$k_2 = L_2 \sin(\theta_2) \quad (87)$$

Em seguida, muda-se a forma de escrever as constantes k_1 e k_2 , através do auxílio da Figura 66:

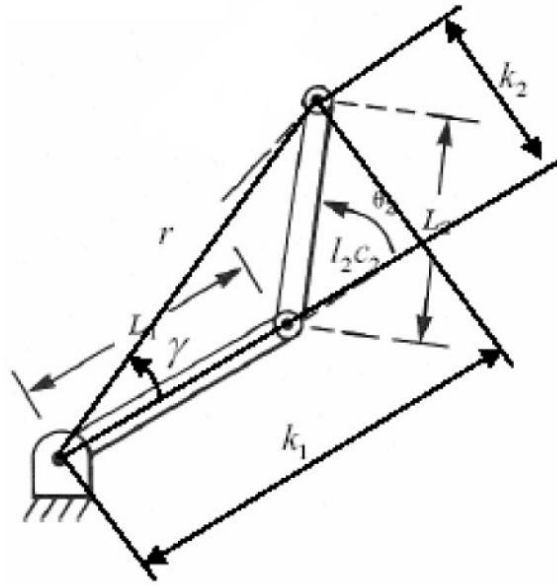


Figura 66 Esquemático para o cálculo de k_1 e k_2

$$r = \sqrt{k_1^2 + k_2^2} \quad (88)$$

$$\gamma = \text{atan2}(k_2, k_1) \quad (89)$$

$$k_1 = r \cdot \cos(\gamma) \quad (90)$$

$$k_2 = r \cdot \sin(\gamma) \quad (91)$$

Com as expressões (84), (85), (90) e (91) retiram-se as expressões:

$$\frac{pl}{r} = \cos(\gamma) \cdot \cos(\theta_2) - \sin(\gamma) \cdot \sin(\theta_2) \quad (92)$$

$$\frac{ql}{r} = \cos(\gamma) \cdot \sin(\theta_2) + \sin(\gamma) \cdot \cos(\theta_2) \quad (93)$$

ou

$$pl = r \cdot \cos(\gamma) \cdot \cos(\theta_2) - r \cdot \sin(\gamma) \cdot \sin(\theta_2) \quad (94)$$

$$ql = r \cdot \cos(\gamma) \cdot \sin(\theta_2) + r \cdot \sin(\gamma) \cdot \cos(\theta_2) \quad (95)$$

ou ainda

$$\frac{pl}{r} = \cos(\gamma + \theta_2) \quad (96)$$

$$\frac{ql}{r} = \sin(\gamma + \theta_2) \quad (97)$$

Aplicando a função $atan2$ às expressões (96) e (97), obtêm-se as equações:

$$\gamma + \theta_2 \cdot atan2\left(\frac{ql}{r}, \frac{pl}{r}\right) = atan2(ql, pl) \quad (98)$$

$$\theta_2 = atan2(ql, pl) - atan2(k_2, k_1) \quad (99)$$

com as expressões 86, 87 e 99 alcança-se a expressão:

$$\theta_2 = atan2\left(z - L_0 - L_3 \cdot \sin(\phi), \sqrt{x^2 + y^2} - L_3 \cdot \cos(\phi)\right) - atan2(L_2 \sin(\theta_2), L_1 + L_2 \cos(\theta_2)) \quad (100)$$

Por fim, depois de se obter θ_2 e θ_3 , tira-se o valor de θ_4 através da equação:

$$\theta_4 = \phi - \theta_2 + \theta_3 \quad (101)$$

5.1.3. PLANEAMENTO DE TRAJETÓRIAS

O planeamento de trajetórias é um passo importante no controlo do movimento do braço robótico. O movimento pode ser efetuado no espaço operacional (em x , y e z) ou no espaço das juntas ($\theta_1, \dots, \theta_n$). Uma trajetória linear é efetuada no espaço operacional, enquanto uma trajetória livre é elaborado no espaço das juntas.

No braço robótico MENTOR vai-se implementar um planeamento de trajetórias no espaço das juntas, denominado *MoveJ* (*Joint move*), e no espaço operacional denominado de *MoveL* (*Linear move*).

O planeamento no espaço operacional é obtido através da equação de uma linha reta no espaço tridimensional. Conhecendo dois pontos, um ponto inicial $P(x_i; y_i; z_i)$ e um ponto final $Q(x_f; y_f; z_f)$, procede-se à aquisição do vetor \mathbf{V} que irá passar pelos os dois pontos e ter a direção do movimento.

O vetor \vec{V} é dado por:

$$\vec{V} = (x_f - x_i; y_f - y_i; z_f - z_i) \quad (102)$$

A equação de linha reta entre os pontos P e Q é:

$$Q = P + \vec{V} \cdot t, \quad t \in [0; 1] \quad (103)$$

Daqui retira-se:

$$Q(t) = \begin{cases} X(t) = x_i + (x_f - x_i) \cdot t \\ Y(t) = y_i + (y_f - y_i) \cdot t, \\ Z(t) = z_i + (z_f - z_i) \cdot t \end{cases} \quad t \in [0; 1] \quad (104)$$

Em computação, a variável t é substituída por (n/k) , sendo que k é o número de pontos que se quer entre os dois pontos e $n \in [0; k]$, como mostra a expressão (105).

$$Q(n) = \begin{cases} X(n) = x_i + (x_f - x_i) \cdot \frac{n}{k} \\ Y(n) = y_i + (y_f - y_i) \cdot \frac{n}{k}, \\ Z(n) = z_i + (z_f - z_i) \cdot \frac{n}{k} \end{cases} \quad n \in [0; k] \quad (105)$$

Para o planeamento de trajetórias no espaço das juntas optou-se por usar uma interpolação cúbica, onde a velocidade inicial e final são nulas e oferece um movimento suave. A interpolação cúbica é dada por:

$$\theta(t) = a_0 + a_1 \cdot t + a_2 t^2 + a_3 t^3 \quad (106)$$

Com as restrições de velocidade inicial e final nulas, retiram-se as seguintes expressões:

$$a_0 = \theta_i \quad (107)$$

$$a_1 = 0 \quad (108)$$

$$a_2 = \frac{3}{t_f^2} (\theta_f - \theta_i) \quad (109)$$

$$a_3 = -\frac{2}{t_f^3} (\theta_f - \theta_i) \quad (110)$$

Sendo que t_f é igual ao valor final do tempo pretendido para efetuar o movimento.

Substituindo as expressões (107), (108), (109) e (110) na expressão (106) obtém-se:

$$\theta(t) = \theta_i + \frac{3}{t_f^2} (\theta_f - \theta_i) \cdot t^2 - \frac{2}{t_f^3} (\theta_f - \theta_i) \cdot t^3 \quad (111)$$

Para o cálculo é usada a equação:

$$\theta(n) = \theta_i + \frac{3}{k^2} (\theta_f - \theta_i) \cdot n^2 - \frac{2}{k^3} (\theta_f - \theta_i) \cdot n^3, n \in [0; k] \quad (112)$$

5.2. HARDWARE

Como considerações iniciais convém referir que a implementação do circuito de controlo do robô, descrito neste capítulo, teve em conta vários fatores para além da funcionalidade pretendida.

O primeiro fator é a tecnologia de montagem *Surface Mount Device* (SMD) escolhida pela necessidade de minimizar o custo de produção e espaço ocupado, uma vez que o *hardware* desenvolvido é para ser integrado num produto acabado onde o espaço disponível é limitado. Há também que referir que alguns dos componentes escolhidos, entre eles os *drivers* dos motores, não estão disponíveis em outras tecnologias de montagem. Quanto à implementação propriamente dita esta será descrita nas subsecções 5.2.1 a 5.2.3, com as principais funcionalidades descritas em pormenor, bem como os circuitos que permitem o seu funcionamento.

Nesta secção mostra-se o *hardware* necessário para o controlo do braço robótico MENTOR, tendo em consideração os pontos referidos anteriormente. O braço robótico MENTOR é composto por seis motores DC de 12 Volt, de 250 mA nominais, e seis potenciômetros de 10 k Ω . Logo, o controlador do robô deverá controlar seis *drivers* de motores DC e ler os seis potenciômetros que vão dar o *feedback* da posição do braço robótico no espaço e a orientação de cada junta.

5.2.1. PLACA DE DRIVER DOS MOTORES

A placa de *driver* do motor é composta pelo ST VNH5180 da ST Microelectronics, que é um *driver* de motores para tensões máximas até quarenta e um Volt e oito Ampere de corrente máxima. Optou-se por desenvolver uma placa para cada motor, porque se ocorrer uma avaria basta só reparar uma placa de *driver* do motor. O esquema do circuito eletrónico para a placa de *driver* do motor encontra-se apresentado na Figura 67.

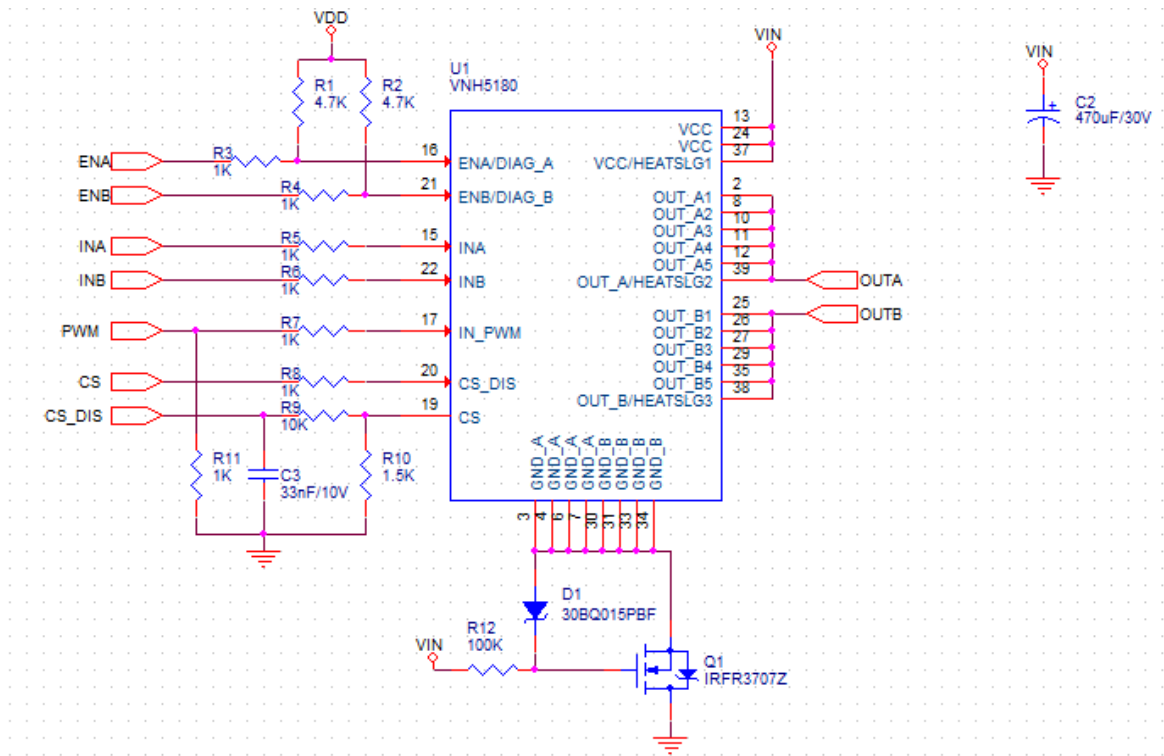


Figura 67 Driver da placa de controlo do motor DC

Uma forma da placa de *driver* do motor ser amigável para o utilizador é informá-lo do sentido em que a corrente passa pelo motor; para este efeito colocaram-se dois LED na saída do *driver*, como mostra a Figura 68, sendo que um dos LED tem o ânodo ligado à saída OUTA e o outro LED tem o ânodo ligado à saída OUTB, fazendo com que seja possível identificar o sentido da corrente através de sinais luminosos.

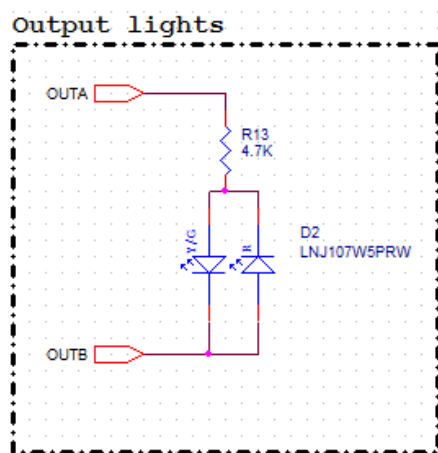


Figura 68 Indicador luminoso para o utilizador do sentido da corrente

Para concluir a placa de *driver* do motor, colocaram-se três conectores, um que irá ser para ligar o motor, outro para a alimentação e outro para o controlo do *driver*, conforme se pode ver na Figura 69.

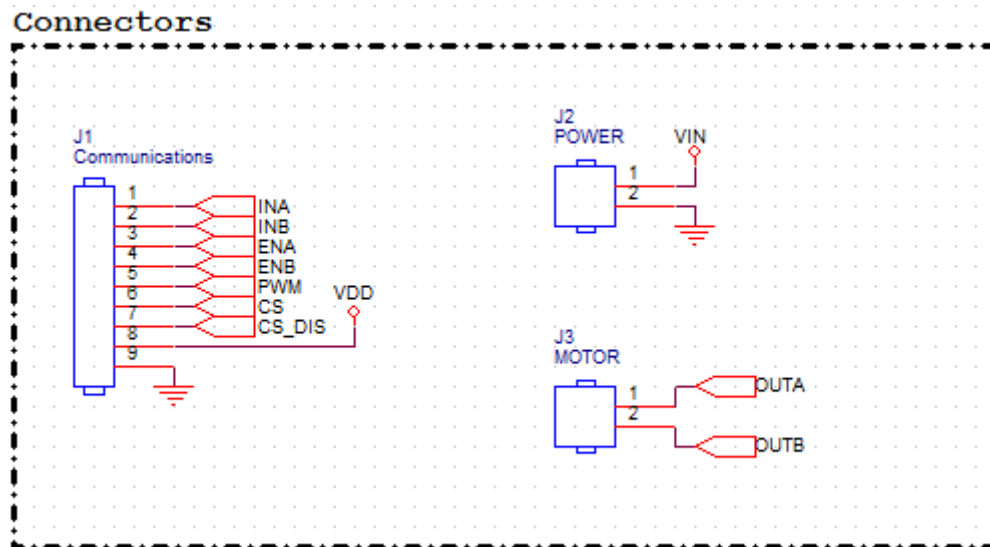


Figura 69 Conectores de alimentação e controlo do motor

5.2.2. CONTROLADOR DO ROBÔ

O controlador do robô é composto por uma unidade de processamento ARM Cortex-M3. O módulo STM32VLDISCOVERY, utilizado para o controlo da aplicação, possui uma interface que permite aceder a todos os pinos do microcontrolador, sendo assim possível implementar circuitos para a utilização dos periféricos necessários, mas também outros de utilidade reconhecida para facilitar a implementação de funcionalidades futuras. A placa de controlo é só composta por um microcontrolador (Figura 70) e conectores de I/Os (Figura 71).

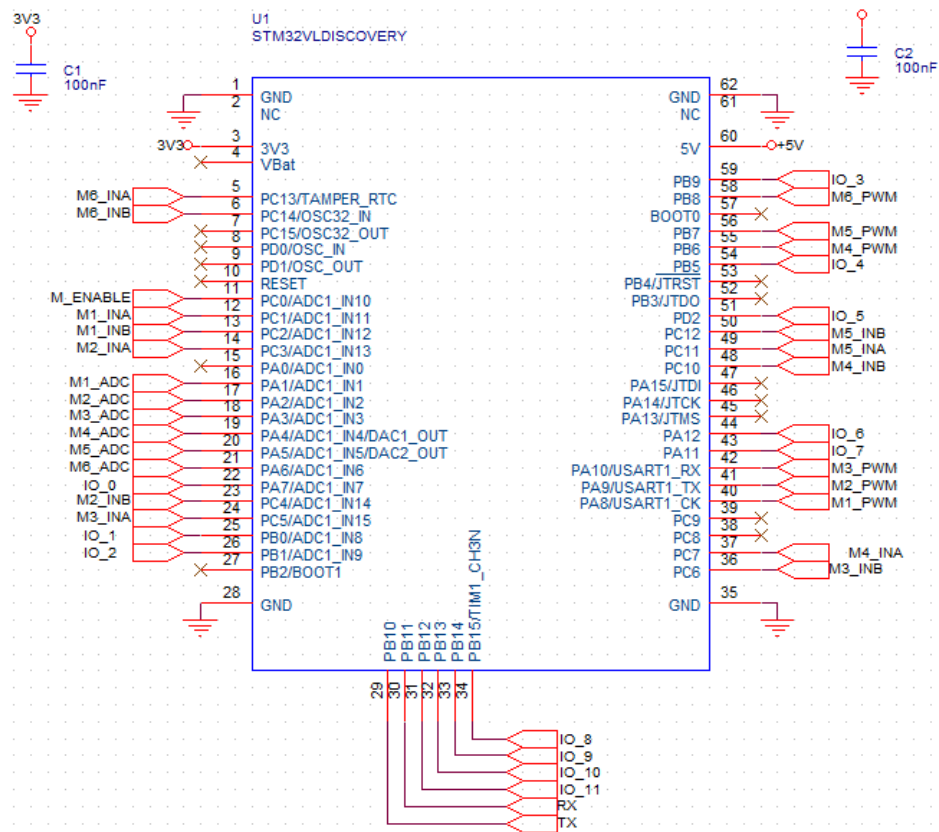


Figura 70 Módulo STM32VLDISCOVERY

Devido a este circuito ser a base do controlador do braço robótico, necessita de receber vários sinais externos e enviar vários sinais e, como tal, esta placa tem vários conetores, como mostra a Figura 71.

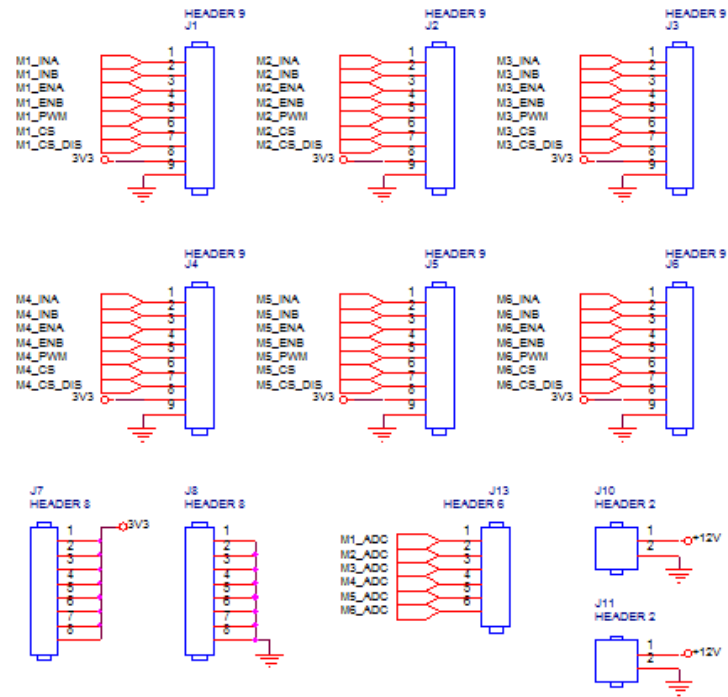


Figura 71 Conectores da placa de controle do robô

5.2.3. JOYSTICK PARA A CONSOLA

Para melhor manipulação e controle da posição do braço robótico, optou-se por juntar uma placa que contenha botões e um *joystick* de dois eixos. Esta placa é composta por um microcontrolador que irá ler os dois potenciômetros do *joystick* e o estado dos botões. De seguida vai mandar valores e ações para a consola, para esta depois enviar para a placa de controle, onde irá ou não atuar sobre as juntas do robô. Para microcontrolador usou-se um Microchip PIC18LF2550, com tensões de alimentação entre 2 Volt a 5,5 Volt dependendo da frequência a que este trabalha, um porto USB e um porto RS-232 para comunicações.

A Figura 72 representa o esquema em ORCAD do microcontrolador usado para controlar e processar os dados da placa do *joystick* para a consola.

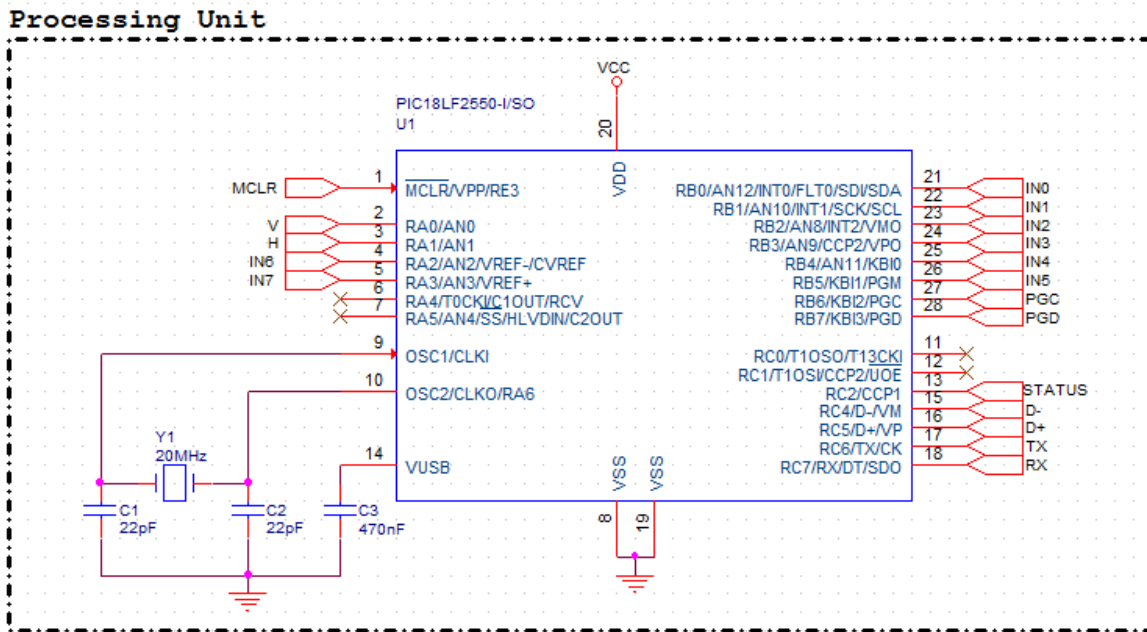


Figura 72 Microcontrolador da placa do joystick

O joystick é composto por dois potenciômetros de 10 kΩ e um switch. O esquema elaborado em ORCAD para a representação do joystick encontra-se representado na Figura 73.

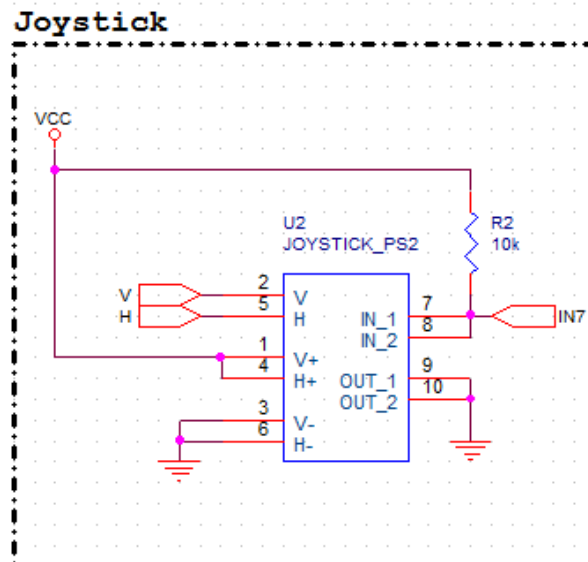


Figura 73 Joystick

De modo a facilitar o controlo e manipulação do robô, colocaram-se switches para abertura e fecho da garra e para a manipulação do Menu da consola. O esquemático dos switches encontra-se ilustrado na Figura 74.

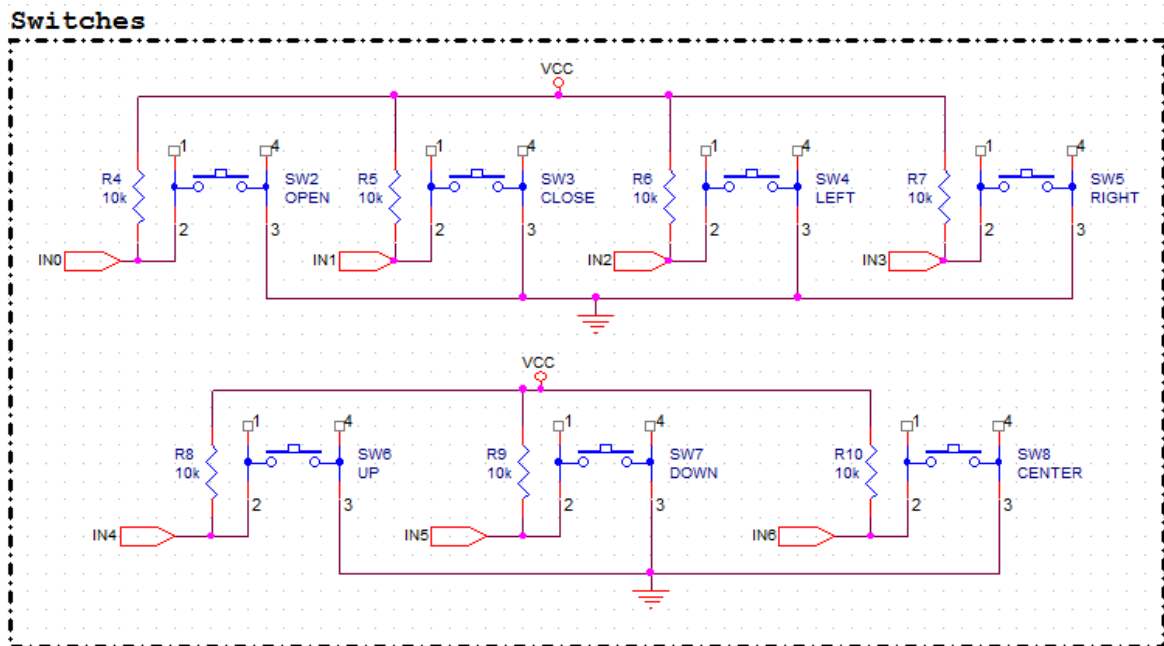


Figura 74 Switches da placa do joystick

5.3. SOFTWARE

O *software* implementado foi dividido em duas vertentes, uma para *software* de alto nível e outra para de baixo nível. O *software* para alto nível foi desenvolvido no Microsoft Visual Studio, em linguagem C#, e teve como finalidade criar uma interface gráfica para a consola (FriendlyArm) de modo que seja amigável para o utilizador e que este consiga controlar o braço robótico sem qualquer problema.

O desenvolvimento de *software* para baixo nível teve como objetivo implementar *firmware* para o STM32VLDISCOVERY, no IDE Keil uVision4, em linguagem C. O STM32VLDISCOVERY tem como finalidade receber instruções da consola e controlar o braço robótico. Também foi desenvolvido um *firmware* para o microcontrolador Microchip PIC18F2550 no IDE MPLAB IDE v8.88, com compilador Hi-Tech C Compiler for PIC18 MCUs em linguagem C, tendo como objetivo principal enviar ações do *joystick* e dos botões para a consola.

As comunicações entre os vários dispositivos são do tipo RS-232, possuindo as seguintes características:

- Tamanho dos dados: 8 bits
- Velocidade de transmissão: 115200 kbps

- Bits de paragem: 1 *stop bit*
- Bits de paridade: nenhum
- Controlo de fluxo: nenhum

Para a comunicação entre o *joystick* e a consola optou-se pelo USB, que se baseia no conceito Cliente-Servidor, onde a consola é o cliente e o *board* que contém o *joystick* é o servidor. Sempre que a consola necessite de ler dados do *board* envia um pedido e o *board* responde. Neste tipo de comunicação tem-se um *array* para envio e receção de dados, sendo que a primeira posição do *array* corresponde ao tipo de pedido que o cliente faz e as restantes posições são referentes à informação para esse pedido. No lado do servidor a primeira posição do *array* corresponde ao pedido do cliente e as restantes à informação do pedido.

5.3.1. MICROSOFT VISUAL STUDIO

A aplicação desenvolvida em Microsoft Visual Studio, na versão 2008 (esta é a versão mais recente suportada pelo FriendlyArm Micro2440 e pelo Windows CE 6.0), foi desenvolvida com o intuito de colocar o FriendlyArm Micro2440 a trabalhar como uma consola para melhor controlo do braço robótico MENTOR. A linguagem usada no Microsoft Visual Studio foi o C# e teve como base conhecimentos prévios desta linguagem. Para controlo dos objetos usados no Visual Studio usou-se a biblioteca de objetos .NET Framework Compact 3.5.

A aplicação desenvolvida para a consola consiste em apresentar um menu simples e amigável, com quatro opções, que são: Programação On-line, Mover Braço, Executar programa e Modificar programa. A Figura 75 ilustra o Menu desenvolvido para a consola.

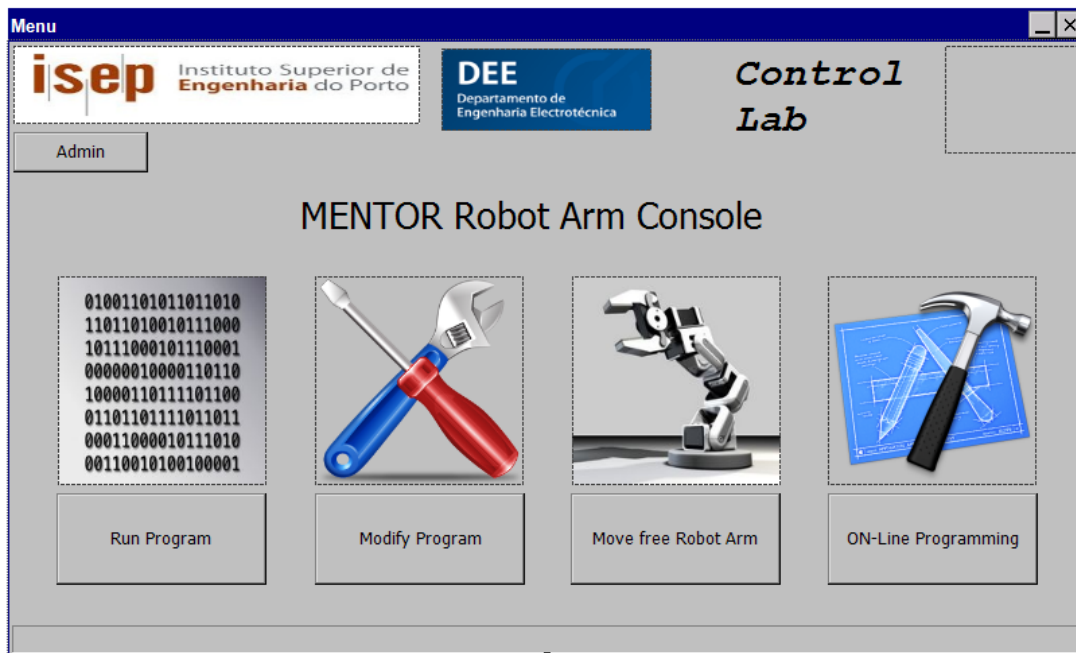


Figura 75 Menu da consola

Todas as opções anteriormente mencionadas comunicam com o controlador do MENTOR através de uma comunicação série (RS-232). Os programas desenvolvidos encontram-se no formato “.txt”, sendo fáceis de editar e de desenvolver num computador móvel. A aplicação, ao iniciar, envia por RS-232 o comando “# CONNECTED”, e só quando receber “OK” é que esta ativa os botões do Menu, ficando até lá desabilitados.

A opção “Executar programa” (*Run Program*) consiste em enviar as linhas de código por RS-232 para o controlador do braço robótico MENTOR, que depois as irá traduzir em movimentos e/ou ações. A Figura 76 ilustra a opção “Executar programa”. A primeira linha de cada programa começa com “Start”, mas a primeira linha que a consola envia para o controlo é “# EXECUTE”, que vai informar o controlador que todos os dados depois do comando “# EXECUTE” serão para mover e/ou efetuar ações no braço robótico. Por cada movimento e/ou ação que o braço efetue será enviada para a consola o valor da posição das juntas e o estado da garra. A consola só envia a próxima linha ou comando quando o controlo do braço robótico enviar para esta “OK”; se não enviar a consola não executa nenhuma linha de código.

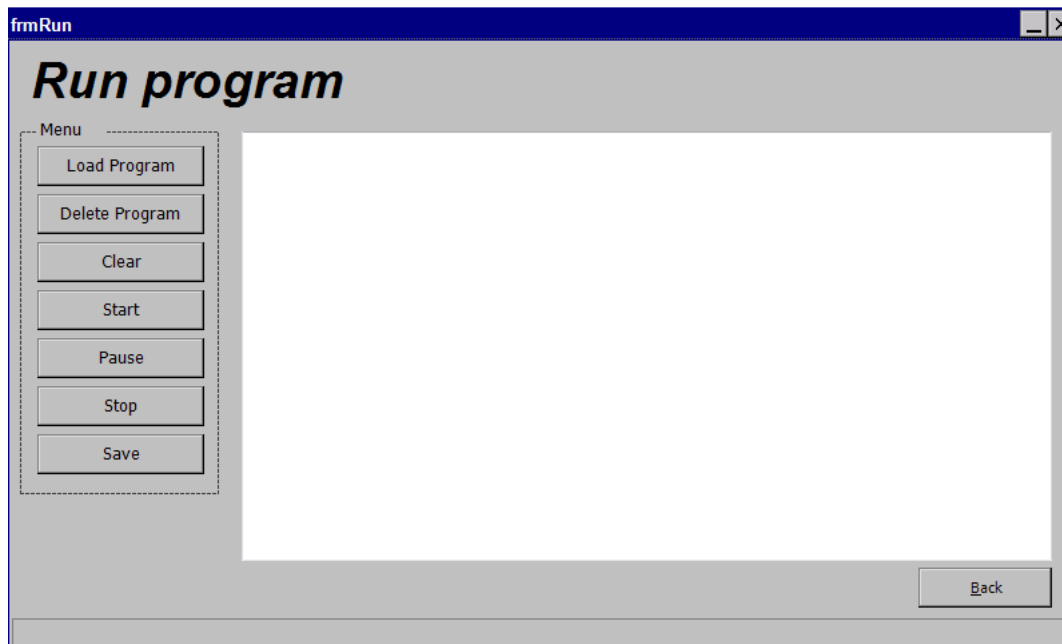


Figura 76 Opção “Executar programa” da consola

De seguida vem a opção “Programação On-Line” (*ON-Line Programming*), que consta no desenvolvimento de um programa de controlo com recurso ao controlador do braço robótico MENTOR para a obtenção da posição que o manipulador se encontra. Para a obtenção da posição, a consola envia o comando “# CAPTURE” e o controlador do braço responde com a posição das juntas, posição espacial e o estado da garra. Nesta opção também existe a opção de mover o braço robótico, mas aqui o objetivo é ajudar no desenvolvimento do programa de controlo. A Figura 77 mostra esta opção.

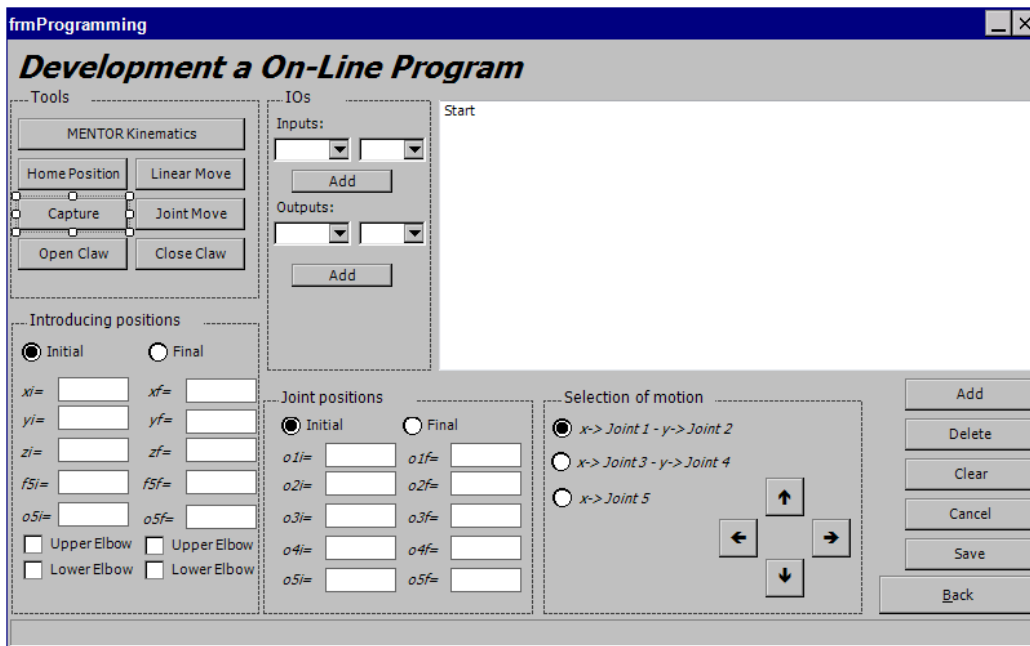


Figura 77 Opção “Programação On-Line” da consola

A seguir vem a opção “Mover braço” (*Move Robot Arm*), que consiste no envio do comando “# MOVE”, para o controlador entrar no modo MOVE e, de seguida, todos os comandos enviados são do tipo mover junta para o lado positivo, lado negativo e junta parada, isto juntamente com a abertura e fecho da garra. O controlador enviará também para a consola a posição da junta para que exista um *feedback* entre ambos. A Figura 78 ilustra a interface desenvolvida para esta opção.

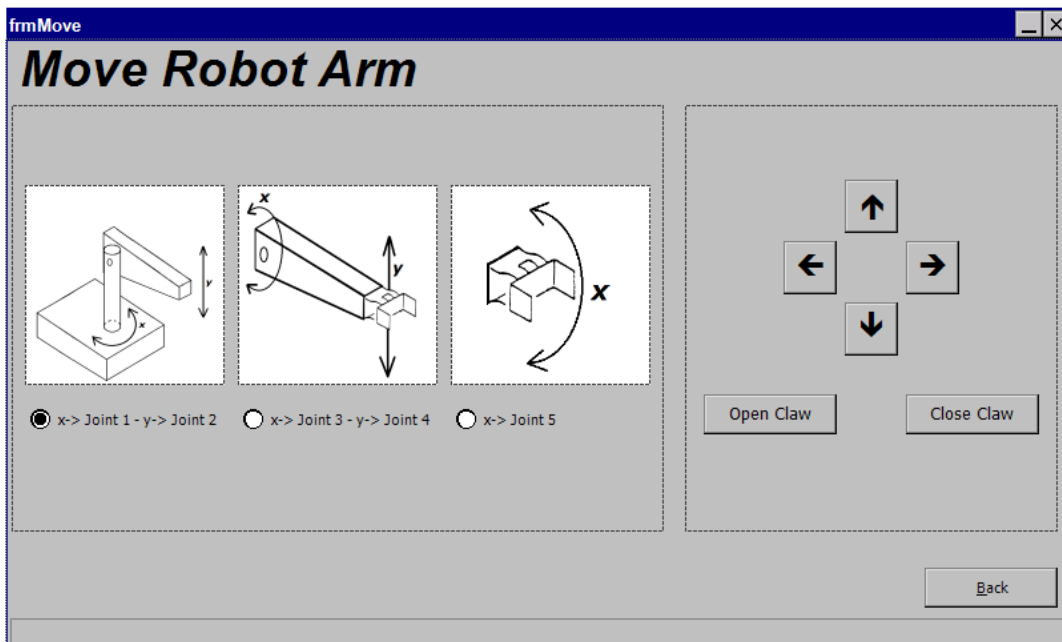


Figura 78 Opção “Move braço” da consola

Por fim, tem-se a última opção, “Modificar programa” (*Modify Program*), que serve para modificar um programa desenvolvido na consola ou um programa desenvolvido em recurso a outra tecnologia, como um computador. Na Figura 79 pode-se observar a respetiva opção.

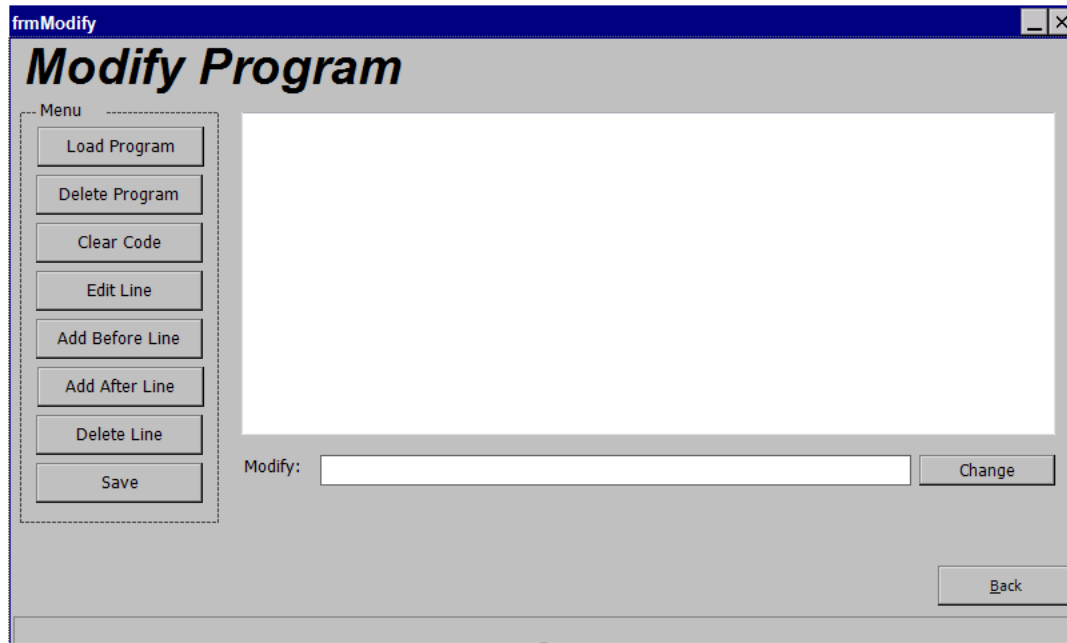


Figura 79 Opção “Modificar programa” da consola

Em todas as opções, exceto na opção “Mover braço”, existe a possibilidade de gravar os ficheiros. Estes são gravados na memória da consola e o nome deles é dado pelo dia, mês, ano, hora e minuto em que é gravado, como exemplo “01_11_12_10_05.txt”. Também todas as opções, exceto a opção “Modificar programa”, recebem dados provenientes do controlador do MENTOR e estão disponíveis na consola na barra inferior, sendo a informação recebida do tipo valor da posição das juntas e valor da posição espacial onde o braço robótico se encontra.

5.3.2. CONTROLO DO ROBÔ

O controlo do robô consiste em receber dados da consola e atuar sobre o robô. A comunicação entre a consola e o controlador do robô é feita através de RS-232. A Figura 80 mostra um diagrama de blocos que relaciona a consola com o braço robótico.

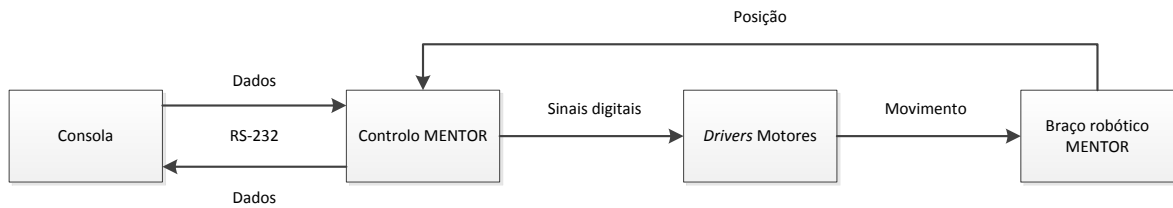


Figura 80 Relação entre a consola, controlo e MENTOR

O controlador começa por fazer as inicializações e depois só inicia quando receber da consola “# CONNECTED”, enviando para a consola “OK”, para esta ativar as opções que a compõem. Depois da comunicação entre o controlador e a consola estar estabelecida, o controlador do MENTOR fica outra vez à espera. Se receber o comando “# EXECUTE”, deve responder com “OK”, ficando a saber que todos os futuros dados serão para movimentar e/ou realizar ações no braço robótico. Os comandos que a consola irá enviar, e que se irão traduzir em movimento e/ou ações são do tipo:

- CLAW_OPEN
- CLAW_CLOSE
- MOVEL(...)
- MOVEJ(...)

Os dois primeiros comandos são relativos a ações da garra, os outros comandos são relativos a movimento. O comando MOVEL(...) é relativo ao braço robótico efetuar uma trajetória linear. Para tal é necessário enviar uma posição inicial, uma posição final, uma orientação inicial e final da junta cinco, o ângulo φ para a posição inicial e final escolhidas, e a configuração que o robô toma para a posição inicial e final (configuração *Upper Elbow* e *Lower Elbow*). O controlador pega nos dados iniciais e finais, coloca no planeamento de trajetórias para movimentos lineares, une os dois pontos através de um vetor, divide a união dos dois pontos em n pontos e depois procede ao cálculo da cinemática inversa para cada ponto. Por cada ponto calculado irá mover o braço robótico e enviar para a consola o valor da posição do robô no espaço, a posição das juntas e o estado da garra, pois só irá receber outro comando quando chegar à posição final e enviar para a consola “OK”.

O controlador entra no modo MOVE quando receber da consola o comando “# MOVE”, e este responde com “OK”, depois irá receber um comando do tipo “MOVE(junta 1, junta 2,

junta 3, junta 4, junta 5, Claw)”. Dentro do comando “MOVE(...)”, o valor da “junta *i*” contém a direção que a junta deve mover ou parar, isto é, se o valor for zero a junta deve estar bloqueada, se for um, a junta desloca-se no sentido contrário aos ponteiros do relógio e se for menos um desloca-se no sentido dos ponteiros do relógio. Por cada movimento e/ou ação que exerça deve enviar para a consola a posição no espaço, a posição das juntas e o estado da garra. Quando esta informação chegar à consola será exibida na barra inferior para o utilizador. O controlador só move a junta *i* quando receber ordem para tal ou se ainda não recebeu ordem para parar e não atingiu o valor limite da junta.

O modo CAPTURE do controlador serve para obter a posição do robô no espaço, a posição das juntas e o estado da garra. Para realizar a programação *On-Line*, o controlador só entra nesse modo quando receber o comando “# CAPTURE” e, de seguida, envia os dados das posições para a consola.

O último modo em que o controlador pode entrar é o modo de HOME. Quando recebe o comando “# HOME”, este coloca o robô com todas as juntas a zero graus, o que dá ao robô a posição nas abcissas de 420 mm, nas ordenadas de 0 mm e de altura 185 mm.

O programa *main* deve conter uma rotina que leia os dados recebidos pela porta série; se os dados recebidos começarem por “#”, significa que o programa deve entrar num modo específico. Por exemplo, se receber o comando “# EXECUTE” deverá ir para a rotina EXECUTE, todos os outros dados que não sejam iniciados com # são dados que irão ser usados na rotina.

Para verificar os dados recebidos através da consola utilizou-se a função *if*, que verifica se o primeiro carácter é ‘#’; se for, a variável *Mode* irá conter os dados recebidos exceto com ‘#’.

```
if(strcmp(GetParam(Data_Received,0,' '),"#")==0)
    strcpy(Mode, GetParam(Data_Received,1,' '));
```

A seguir encontra-se a função *GetParam(...)*, que retorna a análise da mensagem.

```
/**
 * Function that returns the parse the string
 * msg, nr number, using the separator sep
 * @param[in] msg      string target
 * @param[in] nr       number of occurrence
 * @param[in] sep      separator character
 * @return Return pointer to Parameter in a
 * comma delimited list
 */
```

```

char * GetParam( const char * msg, int nr, char
sep)
{
char *dst, *src;
int cnt = 0;

src = (char *)msg;
do
{
dst = buf;
while (*src && (*src != sep))
*dst++ = *src++;
src++;
} while (cnt++ < nr);
*dst = 0;
return buf; //buf variável global do tipo char
*buf
}

```

A seleção do modo de funcionamento da unidade de processamento é feita da seguinte forma:

```

switch(Mode)
{
case 'EXECUTE':
//function EXECUTE
break;

case 'MOVE':
//function MOVE
break;

case 'CAPTURE':
//function CAPTURE
break;

case 'HOME':
//function HOME
break:

default:
GPIOC.GPIO_BIT_0=0; //Disable all
motors
break;
}

```

5.4. CONCLUSÕES

O cálculo da cinemática direta foi simples de realizar depois de se perceber como funciona a convenção de Denavit-Hartenberg e a regra da mão direita. Já o cálculo da cinemática inversa foi demorado, pois levou muito tempo a entender que sem o ângulo de orientação ϕ (ângulo θ_4 medido em relação à base do robô) ter-se-ia um número infinito de soluções para determinadas posições no espaço. As expressões da cinemática inversa calculadas permitem obter a configuração *Upper Elbow* e *Lower Elbow* para determinadas posições no espaço, o que dificultou ainda mais a obtenção das mesmas.

Para o planeamento de trajetórias, foi implementado uma interpolação cúbica para movimentos no espaço das juntas e para o espaço operacional usou-se a equação de uma linha reta que une dois pontos no espaço tridimensional.

O *hardware* desenvolvido teve como base uma unidade de controlo, onde se lê os valores da posição das juntas através de potenciômetros, controlam-se seis *drivers* de motores e pode-se efetuar um elevado processamento a uma frequência elevada. Também aqui foi desenvolvido um pequeno *board*, que contém um *joystick* e botões, com a finalidade de ajudar o utilizador a conseguir uma melhor manipulação do braço robótico.

A fim de finalizar o capítulo da implementação foi necessário desenvolver *software* para a consola, controlo do MENTOR e *joystick*.

O trabalho implementado, encontra-se apresentado no Anexo A, assim como os ficheiros de MATLAB para o cálculo da cinemática direta e inversa do braço robótico MENTOR, uma aplicação destinada ao cálculo da cinemática, ficheiros relativos à implementação da interface gráfica, *software* desenvolvido e o projeto do *hardware* desenvolvido.

6. TESTES E VALIDAÇÃO DO SISTEMA DESENVOLVIDO

Este capítulo baseia-se nos testes e validação de todo o trabalho implementado. Aqui serão discutidos todos os resultados obtidos.

6.1. CÁLCULO NUMÉRICO

Nesta secção validam-se os resultados obtidos na secção 5.1 com uma aplicação desenvolvida (Figura 81) no Visual Studio, em linguagem C#, para essa finalidade.

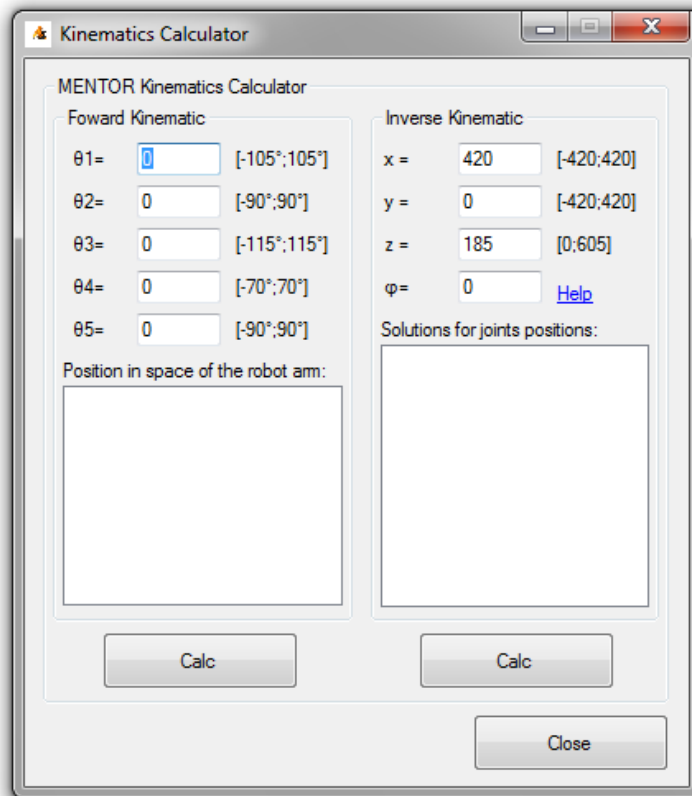


Figura 81 Aplicação desenvolvida para o cálculo da cinemática direta e inversa

6.1.1. CINEMÁTICA DIRETA

As expressões da cinemática direta devem ser validadas, para que não introduzam erros nos seus cálculos. Para tal procede-se à validação e comparação de valores teóricos conhecidos com valores de simulação.

- Para $\theta_1=0^\circ$, $\theta_2=0^\circ$, $\theta_3=0^\circ$, $\theta_4=0^\circ$, $\theta_5=0^\circ$, tem-se a posição calculada $x=420$, $y=0$ e $z=185$ e esta mesma posição se obtém em simulação, como mostra a Figura 82.

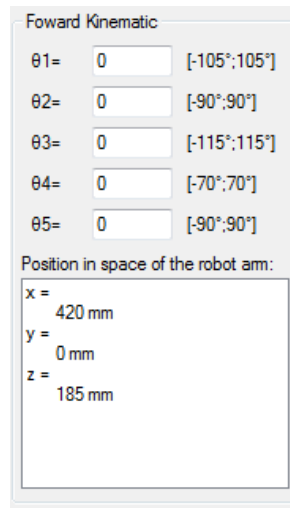


Figura 82 Teste da cinemática direta para a primeira iteração

- Para $\theta_1=45^\circ$, $\theta_2=0^\circ$, $\theta_3=0^\circ$, $\theta_4=0^\circ$, $\theta_5=0^\circ$, tem-se teoricamente a posição $x=296,9848$, $y=296,9848$ e $z=185$, tal como em simulação, como mostra a Figura 83.

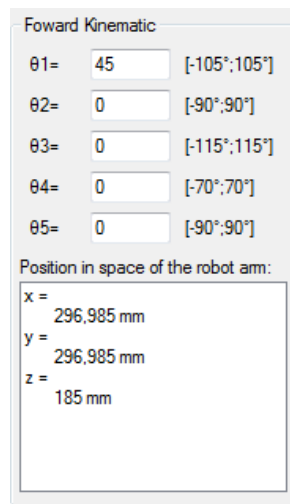


Figura 83 Teste da cinemática direta para a segunda iteração

- Para $\theta_1=-90^\circ$, $\theta_2=0^\circ$, $\theta_3=0^\circ$, $\theta_4=0^\circ$, $\theta_5=0^\circ$, tem-se teoricamente a posição $x=0$, $y=-420$ e $z=185$, e o mesmo resultado se obtém em simulação, como ilustra a Figura 84.

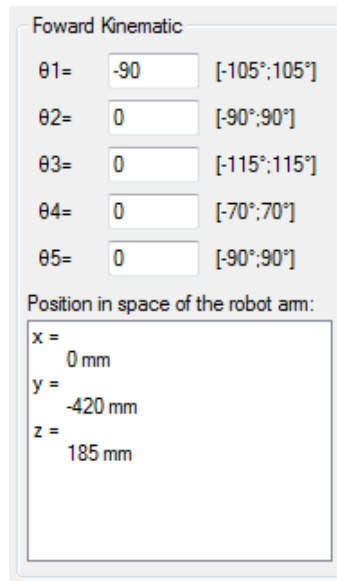


Figura 84 Teste da cinemática direta para a terceira iteração

- Para $\theta_1=0^\circ$, $\theta_2=90^\circ$, $\theta_3=0^\circ$, $\theta_4=0^\circ$, $\theta_5=0^\circ$, tem-se teoricamente a posição $x=0$, $y=0$ e $z=605$, sendo os mesmos valores obtidos em simulação, como apresenta a Figura 85.

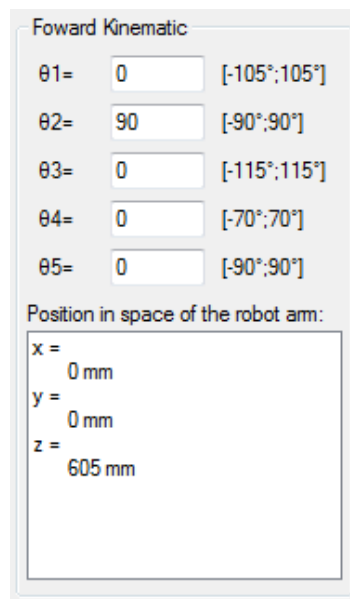


Figura 85 Teste da cinemática direta para a quarta iteração

- Para $\theta_1=-45^\circ$, $\theta_2=0^\circ$, $\theta_3=45^\circ$, $\theta_4=-45^\circ$, $\theta_5=0^\circ$, tem-se teoricamente a posição $x=265,919$, $y=-265,919$ e $z=290,066$, e os mesmos valores são obtidos em simulação, como mostra a Figura 86.

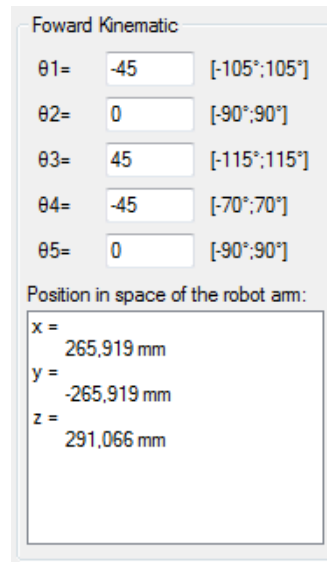


Figura 86 Teste da cinemática direta para a quinta iteração

6.1.2. CINEMÁTICA INVERSA

As expressões obtidas na secção 5.1.2 para o cálculo da cinemática inversa, também vão ser validadas através da aplicação referida na secção 6.1.

- Para a posição $x=420$, $y=0$, $z=185$ e $\varphi=0$, tem-se teoricamente uma só solução da cinemática inversa, que corresponde aos seguintes ângulos, $\theta_1=0^\circ$, $\theta_2=0^\circ$, $\theta_3=0^\circ$, $\theta_4=0^\circ$, $\theta_5=0^\circ$. Na Figura 87 mostram-se os resultados obtidos em simulação.

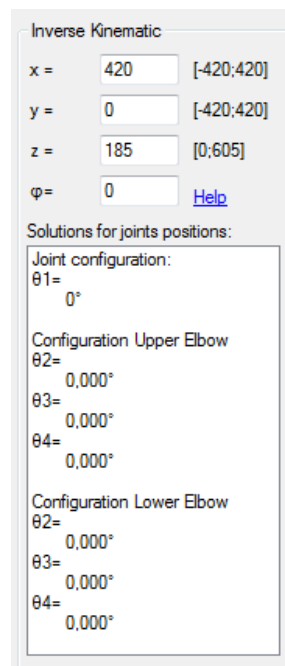


Figura 87 Teste da cinemática inversa para a primeira iteração

- Para a posição $x=376,066$, $y=0$, $z=291,066$ e $\varphi=0$, tem-se teoricamente a solução correspondente aos seguintes ângulos, $\theta_1=0^\circ$, $\theta_2=0^\circ$, $\theta_3=45^\circ$, $\theta_4=-45^\circ$, $\theta_5=0^\circ$. A Figura 88 mostra que para a dada posição final obtêm-se duas configurações do braço.

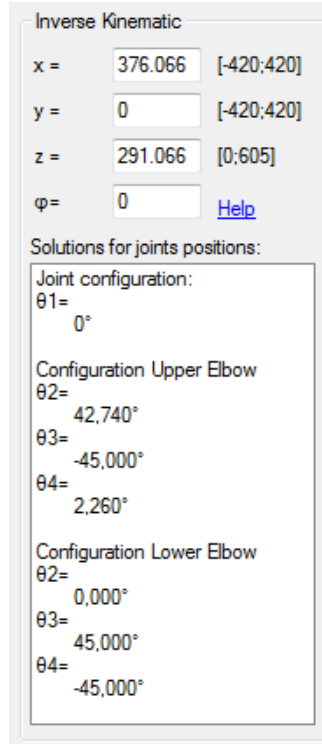


Figura 88 Teste da cinemática inversa para a segunda iteração

- Para a posição $x=226,095$; $y=-226,095$; $z=449,899$ e $\varphi=40$, obtêm-se teoricamente a solução correspondente aos ângulos $\theta_1=45^\circ$, $\theta_2=30^\circ$, $\theta_3=20^\circ$, $\theta_4=-10^\circ$, $\theta_5=0^\circ$. Na Figura 89 pode-se ver que para a dada posição final se obtêm duas configurações do braço.

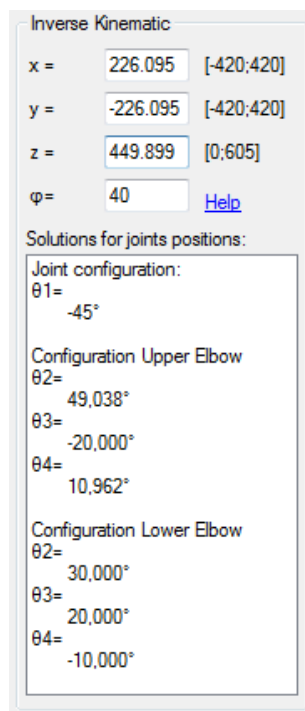


Figura 89 Teste da cinemática inversa para a terceira iteração

6.2. TRABALHO DESENVOLVIDO

De todo o trabalho desenvolvido em termos de *software* e *hardware* só se testou o *software* desenvolvido em Microsoft Visual Studio para o *board* FriendlyArm. Não foi possível verificar o *hardware* e o *software* referentes ao controlo do MENTOR, porque o Departamento de Engenharia Eletrotécnica do Instituto Superior de Engenharia do Porto, não tinha fundos monetários para fazer a aquisição das placas de circuito impresso (PCI ou em inglês *Printed Circuit Board* - PCB) para o controlo do MENTOR e para o *driver* do motor, bem como os componentes que compõem as PCBs. O valor da implementação deste projeto encontra-se calculado no Anexo B.

6.2.1. INTERFACE GRÁFICA - CONSOLA

Como referido na secção anterior, só foi possível testar a aplicação para a consola, porque o candidato a possuía e a pôde testar ao máximo. Como tal são demonstrados testes relativos aos dados que a consola envia e os dados que recebe. Estes testes encontram-se num ficheiro no Anexo A.

6.3. CONCLUSÕES

Nesta secção comentam-se, por tópicos, os resultados obtidos em cada secção deste capítulo, de forma mais sucinta para o leitor.

Na secção 6.1.1 os resultados obtidos foram muito satisfatórios: no caso da primeira, terceira e quarta iteração obteve-se em alguns casos x e y próximos de zero, sendo que o valor teórico é zero. Isto deve-se ao facto do *software* utilizado efetuar cálculos em vírgula flutuante⁵. A obtenção destes valores pode ser obtida através dos ficheiros de MATLAB ou de uma aplicação que estão contidos no Anexo A.

Os resultados obtidos na secção 6.1.2 foram inesperados, porque a equação da cinemática inversa calculada e obtida permite obter dois valores, um para cada configuração que o braço robótico pode adotar para atingir a mesma posição final. Se os valores de posição introduzidos, ou o valor do ângulo φ , não estiverem corretos será emitida uma mensagem de erro.

Para finalizar, devido aos custos de implementação deste projeto e à não existência de fundos monetários para a implementação do mesmo, não foi possível testar o *hardware* desenvolvido nem testar e corrigir pequenos erros do *software* desenvolvido.

⁵ Em vírgula flutuante pode-se representar um número em diversas formas, como por exemplo o número 1234, pode ser representado por $1,234 \times 10^3$ ou $123,4 \times 10^1$, e assim ao se efetuar cálculos em vírgula flutuante o resultado final pode ser uma aproximação do valor real esperado.

7. CONCLUSÕES

Ao longo do presente documento tentou-se mostrar e explicar como surgiu a robótica, que tipos de configuração de robôs estão disponíveis no mercado, quais são as partes que compõem um braço robótico, que tipo de acionamento estes podem ter e quais os sensores são usados para o controlo do mesmo.

Ao nível teórico também se falou na matemática por de trás do controlo de um robô, a qual serve para obter uma posição no espaço conhecendo o valor das juntas ou conhecer o valor das juntas para uma dada posição (cinemática do robô). Na robótica a matemática não se resume só a posicionamento mas também às velocidades e acelerações que os atuadores devem tomar para satisfazer determinados movimentos (dinâmica do robô). Para finalizar tem-se o planeamento de trajetórias e os sistemas de controlo, sendo o planeamento de trajetórias usado para a união de dois pontos, quer no espaço operacional quer no espaço das juntas, e os sistemas de controlo são usados para o controlo de velocidade e aceleração que os atuadores devem ter para uma determinada trajetória.

O capítulo a seguir à parte teórica da robótica apresenta a arquitetura do sistema onde são mencionados os principais componentes e tecnologias para a realização e implementação do projeto em causa. Os mais salientados foram o *board* FriendlyArm que, devido às suas características técnicas, que permitiram usá-lo como uma consola para controlo do braço

robótico, o *board* STM32VLDISCOVERY, que devido ao seu pequeno e poderoso processador, bem como ao baixo preço de aquisição do mesmo beneficiou para efetuar o sistema de controlo do braço robótico, e o *software* de desenvolvimento Microsoft Visual Studio, devido ao poder de computação e simplicidade para o desenvolvimento interfaces gráficas.

De seguida, apresenta-se o capítulo da implementação onde se explicou que o braço robótico MENTOR carecia de um *upgrade* a nível de *software* e *hardware*, pois este não era amigável para o utilizador e era limitado. Este capítulo foi dividido em cálculo numérico, *hardware* e *software*. O cálculo numérico surgiu devido a inexistência das expressões que compõem a cinemática direta e inversa do braço robótico. Neste capítulo não foi possível efetuar um controlo dinâmico devido à escassa informação sobre o sistema de acionamento que o braço robótico MENTOR dispõe, pois para efetuar um controlo dinâmico é necessário saber as características do sistema de acionamento.

O capítulo que se seguiu foi o dos testes realizados relativos à implementação do projeto, sendo que este capítulo ficou incompleto devido às razões mencionadas relacionadas com a falta de fundos monetários. Os resultados obtidos foram satisfatórios em relação à implementação matemática e a aplicação desenvolvida é de fácil utilização e perceção.

De forma a finalizar o presente documento, será apresentada uma síntese e uma perspetiva futura resultante da realização do trabalho. A síntese que se pode concluir com a realização do trabalho foi que, os robôs podem auxiliar na realização de tarefas que podem causar perigo ao ser Humano e trabalhar continuamente sem paragem, tendo o contra, de colocar o posto de trabalho Humano em risco. A precisão, repetibilidade e resolução que os robôs possuem hoje em dia, é elevada devido ao sistema de controlo implementado, tipo de implementação no planeamento de trajetórias e da disposição de sensores de alta resolução no mercado.

Uma das perspetivas futuras deste trabalho já foi mencionada na secção 3.6, e consiste no desenvolvimento de expressões matemáticas para simplificar o cálculo da cinemática inversa de um manipulador com mais de três graus de liberdade, já que o formalismo de Denavit-Hartenberg permitiu simplificar o cálculo das expressões da cinemática direta para manipuladores com mais de três graus de liberdade. Tendo duas formas simples para o utilizador calcular a cinemática de um robô, a matemática envolvida na robótica poderá ser

vista com outros olhos e o desenvolvimento no âmbito da robótica será superior. A outra perspectiva futura, consiste em colocar o braço robótico MENTOR operacional, e como tal, solicitar ao Departamento de Engenharia Eletrotécnica do Instituto Superior de Engenharia do Porto, para dispor verbas monetárias para a compra das placas de circuitos impresso e respectivos componentes.

Por fim sugeria uma atualização do braço robótico MENTOR existente no Laboratório de Controlo do Instituto Superior de Engenharia do Porto, que consistia na alteração dos potenciómetros por codificadores óticos baseados na codificação de Gray (codificadores absolutos), pois estes não sofrem um desgaste mecânico como os potenciómetros. Os codificadores de Gray também têm a vantagem de não ser preciso utilizar portas analógicas, bastando só portas digitais para efetuar a leitura da posição.

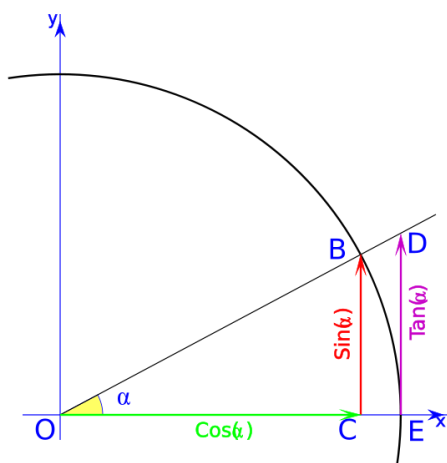
Referências Documentais

- [1] Rosheim; Mark Elling. *Leonardo's Lost Robots*. Springer, 2006, p. 69
- [2] Robô da Unimation; Acedido a 1 de Novembro de 2012;
<http://www.xtimeline.com/evtview.aspx?id=113072>
- [3] Braço de Standford; Acedido a 1 de Novembro de 2012;
<http://infolab.stanford.edu/pub/voy/museum/pictures/display/1-Robot.htm>
- [4] Braço robótico PUMA; Acedido a 1 de Novembro de 2012;
<http://newplans.net/Robotics/Laboratory/act.puma.htm>
- [5] KUKA FAMULUS; Acedido a 1 de Novembro de 2012; <http://www.ifr.org/history/>
- [6] Honda P3; Acedido a 1 de Novembro de 2012;
http://en.wikipedia.org/wiki/Honda_P3
- [7] Sony QRIO; Acedido a 1 de Novembro de 2012; <http://realitypod.com/2010/06/top-10-modern-robots/sonyqrio/>
- [8] ABB FlexPicker; Acedido a 1 de Novembro de 2012;
<http://www.robotmatrix.org/Parallelrobot.htm>
- [9] KUKA KR 16; Acedido a 1 de Novembro de 2012; http://www.kuka-robotics.com/en/products/industrial_robots/low/kr16_2/
- [10] Porto Editora Dicionário; Acedido a 27 de Outubro de 2012;
<http://www.infopedia.pt/lingua-portuguesa/rob%C3%B4>
- [11] Isaac Asimov, *I, Robot*, New York: Doubleday & Company, 1950
- [12] Rosheim, Mark E.; *Robot Evolution: The Development of Anthrobotics*; Wiley; 1994; ISBN 0-471-02622-0
- [13] Esquema robô articulado; Acedido a 1 de Novembro de 2012;
<http://www.robotmatrix.org/ArticulateRobotic.htm>
- [14] Configurações de braços robóticos; Acedido a 1 de Novembro de 2012;
http://www.societyofrobots.com/robot_arm_tutorial.shtml
- [15] Latombe, Jean-Claude; *Robot Motion Planning*; Springer; 1990; ISBN 0-7923-9129-2
- [16] Plataforma de Stewart; Acedido a 1 de Novembro de 2012;
<http://www.physikinstrumente.com/en/news/fullnews.php?newsid=155>
- [17] Robô de acionamento hidráulico; Acedido a 1 de Novembro de 2012;
<http://journal.fluid-power.net/journal/issue13/fprcentre13.html>
- [18] Robô de acionamento pneumático; Acedido a 1 de Novembro de 2012;
<http://ailab.ifi.uzh.ch/robots>

- [19] Potenciometro angular; Acedido a 1 de Novembro de 2012;
http://www.thelearningpit.com/hj/plcs_files/plcs-642.gif
- [20] Spong, Mark W.; Hutchinson, Seth; Vidyasagar, M.; Robot Modeling and Control; Wiley, 2005; ISBN 978-0471649908
- [21] Siciliano, Bruno; Khatib, Oussama - Springer Handbook of Robotics: Springer+Business Media; 2008. ISBN 978-3-540-23957-4, e-ISBN 978-3-540-30301-5
- [22] Craig, John J. – Introduction to Robotics Mechanics and Control; Addison-Wesley Publishing Company, Inc; 1989. ISBN 0-201-09528-9
- [23] Nof, Shimon Y. - Handbook of Industrial Robotics; John Wiley & Sons, Inc; 1999. ISBN 0-471-17783-0
- [24] Campos, M.; Kumar, V.; Bajcsy R.; "Kinematic Identification of Linkages," *Proceedings of the 3rd International Conference on Advances in Robot Kinematics*, Ferrara, Italy, Sep. 7-9, 1992
- [25] Kumar, V.; Introduction to Robot Geometry and Kinematics; Acedido a 27 de Outubro de 2012;
<http://www.seas.upenn.edu/~meam520/notes02/IntroRobotKinematics5.pdf>
- [26] Documentação do braço robótico MENTOR consultar Anexo A
- [27] FriendlyArm; Acedido a 1 de Novembro de 2012; <http://www.friendlyarm.net/>
- [28] ATX Documentation v2.2, Acedido a 1 de Novembro de 2012;
http://www.formfactors.org/developer%5Cspecs%5Catx2_2.pdf
- [29] STM32VLDISCOVERY, Acedido a 1 de Novembro de 2021;
http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/USER_MANUAL/CD00267113.pdf

Apêndice A. Relações trigonométricas

A seguinte figura ajuda a entender a trigonometria.



O cosseno do ângulo α é obtido através da divisão do cateto adjacente (OC) pela hipotenusa (OB), e é dado pela expressão:

$$\cos\alpha = \frac{\text{cateto adjacente}}{\text{hipotenusa}} \quad (1)$$

O seno do ângulo α é obtido através da divisão do cateto oposto (BC) pela hipotenusa (OB), e é dado por:

$$\sin\alpha = \frac{\text{cateto oposto}}{\text{hipotenusa}} \quad (2)$$

A tangente de um ângulo α é obtida pela expressão:

$$\tan\alpha = \frac{\sin\alpha}{\cos\alpha} = \frac{\text{cateto oposto}}{\text{cateto adjacente}} \quad (3)$$

Pelo teorema de Pitágoras sabe-se que a soma do quadrado dos catetos é igual ao quadrado da hipotenusa, na trigonometria o quadrado da hipotenusa é obtido através do quadrado do cateto adjacente com o quadrado do cateto oposto, então sabe-se que:

$$\cos^2\alpha + \sin^2\alpha = 1 \quad (4)$$

Nas expressões (5) e (6) têm-se duas relações usadas no cálculo trigonométrico.

$$\cos(x \pm y) = \cos(x) \cdot \cos(y) \mp \sin(x) \cdot \sin(y) \quad (5)$$

$$\sin(x \pm y) = \sin(x) \cdot \cos(y) \pm \cos(x) \cdot \sin(y) \quad (6)$$

Para uma equação não linear do tipo:

$$P\cos(\alpha) + Q\sin(\alpha) + R = 0 \quad (7)$$

Pode-se definir:

$$\cos(\gamma) = \frac{P}{\sqrt{P^2+Q^2}} \quad \text{e} \quad \sin(\gamma) = \frac{Q}{\sqrt{P^2+Q^2}} \quad (8)$$

Note-se que é sempre possível determinar γ usando a função atan2 :

$$\gamma = \text{atan2}\left(\frac{Q}{\sqrt{P^2+Q^2}}, \frac{P}{\sqrt{P^2+Q^2}}\right) \quad (9)$$

Agora rescreve-se a equação (7) como:

$$\cos(\alpha)\cos(\gamma) + \sin(\alpha)\sin(\gamma) + \frac{R}{\sqrt{P^2+Q^2}} = 0 \quad (10)$$

ou

$$\cos(\alpha - \gamma) = -\frac{R}{\sqrt{P^2+Q^2}} \quad (11)$$

Isto dá duas soluções para α em termos do ângulo conhecido γ :

$$\alpha = \gamma + \sigma \cos^{-1}\left(\frac{-R}{\sqrt{P^2+Q^2}}\right), \sigma = \pm 1 \quad (12)$$

Anexo A. Documentação do trabalho

O candidato optou pela colocação de toda a informação relativa ao projeto num CD e na sua página pessoal do Instituto Superior de Engenharia do Porto, ave.dee.isep.ipp.pt/~1060380. Nesta página pode-se obter os ficheiros do *hardware*, *software*, cálculo numérico e o documento escrito.

Anexo B. Custos de implementação

No presente anexo são apresentados os custos da implementação do presente projeto.

Componente	Descrição	Quantidade	Custo total
PCB	Controlo do MENTOR – dimensão 80 mm × 125 mm, dupla face, <i>SolderMask</i> , serigrafia e furos metalizados	1	85,00 € [1]
	<i>Driver</i> do motor – dimensão 40 mm × 40 mm, dupla face, <i>SolderMask</i> , serigrafia e furos metalizados	6	85,00 € [1]
	Placa do <i>joystick</i> – dimensão 40 mm × 62 mm, face simples, <i>SolderMask</i> , serigrafia e furos metalizados	4	40,00 € [1]
Processador	STM32VLDISCOVERY	1	8,33 € [2]
	FriendlyArm Micro2440 + 7’’ LCD Display + SDK <i>board</i>	1	185,00 € [3]
	Microchip PIC18F2550 - SMD	1	4,35 € [2]
<i>Driver</i>	ST VN5180	6	3,14 € [2]
Resistências	1 k Ω	100	2,20 € [2]
	1,5 k Ω	100	2,20 € [2]
	4,7 k Ω	100	2,20 € [2]
	10 k Ω	100	2,20 € [2]
	100 k Ω	100	2,20 € [2]
Condensadores	100 nF/10 V	100	1,20 € [2]
	470 uF/30 V	10	5,00 € [2]
	33 nF/10 V	100	1,00 € [2]
	22 pF/10 V	100	1,00 € [2]

	470 nF/10 V	100	1,00 € [2]
Sinalização	LED	4	2,00 € [2]
MOSFET	IRFR3707Z	6	0,96 € [2]
Cristal	20 MHz	1	0,30 € [2]
Botões	<i>Switches Push-Buttons</i>	10	0,10 € [2]
Conectores	Mini USB-B	1	0,38 € [2]
	ATX – 24 pinos	1	0,92 € [2]
	1×12	20	2,00 € [2]
	1×6	20	2,00 € [2]
	1×4	20	2,00 € [2]
	1×2	20	2,00 € [2]
Outros	<i>Joystick</i>	1	2,60 € [4]
	Fonte de alimentação ATX	1	20,00 €
		Total:	500,00 € [5]

[1] – Circuito Total, Acedido a 1 de Novembro de 2012, <http://www.circuitotal.pt/>

[2] – Mouser Electronics, Acedido a 1 de Novembro de 2012, <http://pt.mouser.com/>

[3] – GeeTech, Acedido a 1 de Novembro de 2012, <http://www.geeetech.com/1g-micro2440-with-7-inch-tft-lcd-sdk-p-237.html>

[4] – DealExtreme, Acedido a 1 de Novembro de 2012, <http://dx.com/p/genuine-ps2-ps3-controller-repair-parts-3-pin-analog-stick-black-107125?item=19>

[5] – O valor previsto contém uma pequena margem para imprevistos.

