



Universidade do Porto

---

Faculdade de Engenharia

**FEUP**

**Departamento de Engenharia Electrotécnica e de  
Computadores**

***Simulação Dinâmica e Controlo de Robôs  
Industriais***

**Dissertação submetida para provas de Mestrado em  
Engenharia Electrotécnica e de Computadores**

**Nuno Miguel Fonseca Ferreira**

**Porto, Setembro de 1999**



# Índice

Agradecimentos	i
Resumo	ii
CAPÍTULO 1	
1.1 - Introdução	1
1.2 - Objectivos	2
1.3 - Estrutura da Dissertação	2
CAPÍTULO 2	
2.1 - Introdução	4
2.2 - Manipuladores Robóticos	6
2.2.1 - Região de Trabalho “WorkSpace”	7
2.2.2 - Problemas Geométricos Relacionados com o Espaço de Trabalho	8
2.3 - Introdução à Cinemática	10
2.3.1 - Cinemática directa do robô RR	12
2.3.2 - Cinemática directa do robô RP	13
2.3.3 - Cinemática inversa do robô RR	14
2.3.4 - Cinemática inversa do robô RP	15
2.4 - Planeamento de trajectórias	16
2.4.1 - Planeamento de Trajectórias no Espaço Cartesiano.	19
2.4.2 - Planeamento de Trajectórias no Espaço das Juntas	19
2.5 - Dinâmica do Robô	20
2.5.1 - Dinâmica Inversa	21
2.5.2 - Dinâmica Directa	21
2.5.3 - Dinâmica Inversa do Robô RR	22
2.5.4 - Dinâmica Inversa do Robô RP	22
2.6 - Fenómenos Dinâmicos nas Juntas dos Robôs	23
2.6.1 - Juntas dos Robôs com Folgas	24
2.6.2 - Juntas dos Robôs com Flexibilidade	25
2.6.3 - Juntas dos Robôs com Atrito	26
2.6.4 - Juntas dos Robôs com Limitações dos Binários	26
2.4.5 - Juntas dos Robôs com Limitações dos Sensores	27
2.7 - Análise Dinâmica e Controlo de Robôs	27
2.7.1 - Controladores PID	28
2.7.2 - Controladores baseados em modelos	29
2.7.3 - Controladores Não Lineares	32
2.7.4 - Controlo Baseado no Espaço Cartesiano	34
2.8 - Controlo Híbrido Posição/Força de Robôs	37
2.9 - Conclusões	41
CAPÍTULO 3 - O Programa de Simulação RobLib	
3.1 - Introdução	42
3.2 - Robôs Disponíveis	43
3.3 - Estrutura Geral do RobLib	43
3.4 - Planeamento de trajectórias de posição	48
3.5 - O planeamento da trajectória de posição e força	51
3.6 - Algoritmos de Controlo de Posição	54
3.7 - Algoritmos de Controlo de Posição/Força	63
3.8 - Base de dados	65
3.9 - Gráficos do simulador	71
3.10 - Conclusões	74
CAPÍTULO 4 - Análise Dinâmica e Estudo de Algoritmos de Controlo	
4.1 - Introdução	76
4.2 - Simulação de Algoritmos de Controlo	77
4.3 - Controlo nas juntas	80
4.3.1 - Algoritmos de controlo linear	80
4.3.2 - Algoritmos de controlo não lineares	85
4.3.3 - Algoritmos de controlo baseados em modelos	89
4.4 - Controladores baseados no espaço cartesiano	91
4.4.1 - Jacobiano Transposto e Jacobiano Inverso : Sub-controladores lineares	91

4.4.2 – Jacobiano Transposto e Jacobiano Inverso : Sub-controladores não lineares	93
4.4.3 - Controlador baseado no modelo do robô no espaço operacional	95
4.5 - Simulação de trajetórias	96
4.6 - Algoritmos de controlo baseados no Modelo do Robô	100
4.7 - Fenómenos dinâmicos nas juntas dos robôs	103
4.8 - Controlo do robô RP	112
4.9 - Simulação do Algoritmo Controlo Híbrido	114
4.9.1 - Simulação do controlo de posição	116
4.9.2 - Simulação do controlo de força	117
4.9.3 - Simulação dos fenómenos dinâmicos nas juntas no controlo de posição segundo o eixo y e força segundo o eixo x	120
4.10 - Conclusões	125
<b>CAPÍTULO 5 - Manipulabilidade de Sistemas Robóticos com Um e Dois Braços</b>	
5.1 - Introdução	126
5.2 - A Manipulabilidade do Robô RR	127
5.3 - Cinemática de Dois Robôs em Cooperação	127
5.3.1 - A cinemática directa do robô A	128
5.3.2 - A cinemática inversa do robô A	128
5.3.3 - A cinemática directa do robô B	129
5.3.4 - A cinemática inversa do robô B	129
5.4 - A Distribuição Normal Bidimensional	129
5.5 - RobLib – Manipulabilidade	133
5.6 - Resultados Numéricos	136
5.6.1 - Simulações ao Robô RR	136
5.6.2 - Simulações a dois robôs a trabalhar em cooperação	141
5.7 - Conclusões	153
<b>CAPÍTULO 6 – Conclusões</b>	
6.1 - Introdução	155
6.2 - Conclusões do trabalho	155
6.3 - Perspectivas de Desenvolvimento Futuro	156
<b>Bibliografia</b>	<b>158</b>

# Lista de Figuras

Figura 2.1	- Robô RR.	6
Figura 2.2	- Robô RP.	6
Figura 2.3	- Área de trabalho do robô RR com $q_1 \text{ min} = -\pi$ e $q_1 \text{ max} = \pi$ .	7
Figura 2.4	- Área de trabalho do robô RP.	7
Figura 2.5	- Pontos não alcançáveis na trajetória.	8
Figura 2.6	- Planeamento da trajetória nas juntas.	9
Figura 2.7	- Singularidades do robô RR.	9
Figura 2.8	- Trajetórias com diferentes soluções.	10
Figura 2.9	- Cinemática directa $\{q_1, q_2\} \rightarrow \{x, y\}$ do robô RR.	11
Figura 2.10	- Cinemática inversa $\{x, y\} \rightarrow \{q_1, q_2\}$ do robô RR.	11
Figura 2.11	- Cinemática do Robô RP.	13
Figura 2.12	- Diagrama temporal da trajetória para aceleração a degrau.	17
Figura 2.13	- Diagrama temporal da trajetória para aceleração sinusoidal.	18
Figura 2.14	- Robô com juntas ideais.	23
Figura 2.15	- Robô com folgas nas engrenagens.	24
Figura 2.16	- Robô com flexibilidade nas juntas.	25
Figura 2.17	- Modelização do atrito.	26
Figura 2.18	- Controladores lineares do tipo PID.	28
Figura 2.19	- O controlador do tipo compensação por acção “antecipativa”.	30
Figura 2.20	- O controlo do binário calculado.	31
Figura 2.21	- Controlador de estrutura variável.	32
Figura 2.22	- Controlador do inverso do Jacobiano.	34
Figura 2.23	- Controlador do Jacobiano transposto.	35
Figura 2.24	- Controlador baseado no modelo do robô.	37
Figura 2.25	- Robô em interacção com o ambiente com inclinação $90^\circ$ .	37
Figura 2.26	- Modelo do ambiente.	38
Figura 2.27	- Sistema de coordenadas do robô em interacção com o ambiente.	38
Figura 2.28	- Controlador híbrido.	39
Figura 3.1	- Janela Principal do programa RobLib.	44
Figura 3.2	- Opção da escolha do tipo de robô.	44
Figura 3.3	- Opção da escolha do tipo de fenómenos nas juntas.	44
Figura 3.4	- Janela de visualização dos parâmetros do robô RR.	45
Figura 3.5	- Janela de visualização dos parâmetros do robô RP.	45
Figura 3.6	- Janela de visualização da região de trabalho do robô RR sem limitações nas juntas $\{q_1, q_2\}$ .	46
Figura 3.7	- Janela de visualização da região de trabalho do robô RR com limitações nas juntas (i.e. com $-180^\circ \leq q_1 \leq 0^\circ$ ).	46
Figura 3.8	- Janela de visualização da região de trabalho do robô RP sem limitações nas juntas.	47
Figura 3.9	- Janela informativa.	47
Figura 3.10	- Janela do manual do utilizador.	48
Figura 3.11	- Opção Trajectory para escolha do tipo de trajetória.	48
Figura 3.12	- Janela de inserção dos parâmetros da trajetória (no espaço operacional).	49
Figura 3.13	- Janela de aviso dos limites.	49
Figura 3.14	- Janela do robô RR em movimento.	50
Figura 3.15	- Janela do robô RP em movimento.	50
Figura 3.16	- Janela da base de dados das trajetórias.	51
Figura 3.17	- Janela da base de dados das trajetórias.	51
Figura 3.18	- Janela de opção de planeamento de trajetória de posição e de força.	52
Figura 3.19	- Janela de introdução dos dados.	52
Figura 3.20	- Janela da introdução do modelo do ambiente.	53

Figura 3.21	- Movimento (desejado) do robô em interação com o ambiente.	54
Figura 3.22	- Janelas de selecção dos controladores de posição.	54
Figura 3.23	- Janela da opção do controlador de posição não linear do tipo CEV com um controlo no espaço das juntas.	55
Figura 3.24	- Controladores de estrutura variável.	57
Figura 3.25	- Janela de visualização do progresso da simulação.	57
Figura 3.26	- Janela da alteração dos parâmetros e fenómenos dinâmicos das juntas do robô.	58
Figura 3.27	- Janela de selecção de opções para um controlador linear no Espaço das juntas.	59
Figura 3.28	- Controlador do tipo “Computed Torque”.	59
Figura 3.29	- Controlador do tipo Feedforward com sub-controlador do tipo PID.	60
Figura 3.30	- Controlador do tipo Feedforward com sub-controlador do tipo VSS-FOM-PD2.	60
Figura 3.31	- Controlo baseado no modelo do robô e com acção no espaço operacional.	61
Figura 3.32	- Controlo no espaço operacional, com sub-controladores de estrutura variável e com o jacobiano transposto.	61
Figura 3.33	- Controlo no espaço operacional, com sub-controladores de estrutura variável com o jacobiano inverso.	62
Figura 3.34	- Controlo no espaço operacional, com sub-controladores PID com o jacobiano transposto.	62
Figura 3.35	- Controlo no espaço operacional, com sub-controladores PID e com o jacobiano inverso.	63
Figura 3.36	- Controlo de posição e de força.	63
Figura 3.37	- Controlador híbrido com sub-controladores não lineares.	64
Figura 3.38	- Controlador híbrido com sub-controladores lineares.	64
Figura 3.39	- Janela de acesso à consulta da base de dados.	65
Figura 3.40	- Janela principal da base de dados.	66
Figura 3.41	- Janela com um Robô RR e uma algoritmo de controlo VSS.	67
Figura 3.42	- Tipo de junta a seleccionar.	67
Figura 3.43	- Selecção do tipo de trajectória.	68
Figura 3.44	- Controlador não linear de estrutura variável.	68
Figura 3.45	- Opção do controlador do tipo baseado em modelos.	69
Figura 3.46	- Modelo do robô para a trajectória seleccionada.	69
Figura 3.47	- Tipo de variável a visualizar graficamente.	70
Figura 3.48	- Controladores de posição/força.	71
Figura 3.49	- Janela de acesso aos gráficos.	71
Figura 3.50	- Janela de selecção dos gráficos.	72
Figura 3.51	- Janela de selecção da variável a observar.	72
Figura 3.52	- Gráfico da evolução temporal da variável x.	73
Figura 3.53	- “Zoom” do gráfico da evolução temporal da variável x.	73
Figura 3.54	- Selecção da simulação dos controladores de força.	74
Figura 3.55	- Simulação da variável força do controlador híbrido.	74
Figura 4.1	- Trajectória de referência para o robô RR.	78
Figura 4.2	- Robô RR com juntas ideais.	78
Figura 4.3	- Esquema das simulações para os algoritmos existentes no RobLiB.	79
Figura 4.4	- (a) evolução temporal $T1(t)$ do binário da junta 1, (b) evolução temporal $T2(t)$ do binário da junta 2, (c) evolução temporal $x(t)$ da posição x, (d) evolução temporal $y(t)$ da posição y, para o controlador proporcional com os parâmetros da tabela 4.2 (i.e. $f_n=10\text{Hz}$ ), considerando a vermelho a trajectória planeada e verde a simulação.	81
Figura 4.5	- (a) evolução temporal $T1(t)$ do binário da junta 1, (b) evolução temporal $T2(t)$ do binário da junta 2, (c) evolução temporal $x(t)$ da posição x, (d) evolução temporal $y(t)$ da posição y, para o controlador proporcional com os parâmetros da tabela 4.3 (i.e. $f_n=100\text{Hz}$ ), considerando a vermelho a trajectória planeada e verde a simulação.	82
Figura 4.6	- (a) evolução temporal $T1(t)$ do binário da junta 1, (b) evolução temporal $T2(t)$ do binário da junta 2, (c) evolução temporal $x(t)$ da posição x, (d) evolução temporal $y(t)$ da posição y, para o controlador proporcional integral com os parâmetros da tabela 4.2 (i.e. $f_n=10\text{Hz}$ ), considerando a vermelho a trajectória planeada e azul a	82

	simulação.	
Figura 4.7	- (a) evolução temporal T1(t) do binário da junta 1, (b) evolução temporal T2(t) do binário da junta 2, (c) evolução temporal x(t) da posição x, (d) evolução temporal y(t) da posição y, para o controlador proporcional derivativo com os parâmetros da tabela 4.2 (i.e. $f_n=10\text{Hz}$ ), considerando a vermelho a trajectória planeada e azul a simulação.	83
Figura 4.8	- (a) evolução temporal T1(t) do binário da junta 1, (b) evolução temporal T2(t) do binário da junta 2, (c) evolução temporal x(t) da posição x, (d) evolução temporal y(t) da posição y, para o controlador proporcional derivativo com os parâmetros da tabela 4.3 (i.e. $f_n=100\text{Hz}$ ), considerando a vermelho a trajectória planeada e azul a simulação.	83
Figura 4.9	- (a) evolução temporal T1(t) do binário da junta 1, (b) evolução temporal T2(t) do binário da junta 2, (c) evolução temporal x(t) da posição x, (d) evolução temporal y(t) da posição y, para o controlador proporcional integral derivativo com os parâmetros da tabela 4.2 (i.e. $f_n=10\text{Hz}$ ), considerando a vermelho a trajectória planeada e azul a simulação.	84
Figura 4.10	- (a) evolução temporal T1(t) do binário da junta 1, (b) evolução temporal T2(t) do binário da junta 2, (c) evolução temporal x(t) da posição x, (d) evolução temporal y(t) da posição y, para o controlador proporcional integral derivativo com os parâmetros da tabela 4.3 (i.e. $f_n=100\text{Hz}$ ), considerando a vermelho a trajectória planeada e cor de rosa a simulação.	84
Figura 4.11	- (a) evolução temporal T1(t) do binário da junta 1, (b) evolução temporal T2(t) do binário da junta 2, (c) evolução temporal x(t) da posição x, (d) evolução temporal y(t) da posição y, para o controlador VSS-FOM-PDD2 com os parâmetros da tabela 4.4 (i.e. $f_n=10\text{Hz}$ ), considerando a vermelho o trajectória planeada e a azul a simulação.	86
Figura 4.12	- (a) evolução temporal T1(t) do binário da junta 1, (b) evolução temporal T2(t) do binário da junta 2, (c) evolução temporal x(t) da posição x, (d) evolução temporal y(t) da posição y, para o controlador VSS-FOM-PDD2 com os parâmetros da tabela 4.5 (i.e. $f_n=100\text{Hz}$ ), considerando a vermelho a trajectória planeada e azul a simulação.	86
Figura 4.13	- (a) evolução temporal T1(t) do binário da junta 1, (b) evolução temporal T2(t) do binário da junta 2, (c) evolução temporal x(t) da posição x, (d) evolução temporal y(t) da posição y para o controlador VSS-FOM-PID com os parâmetros da tabela 4.4 (i.e. $f_n=10\text{Hz}$ ), considerando a vermelho a trajectória planeada e preto a simulação.	87
Figura 4.14	- (a) evolução temporal T1(t) do binário da junta 1, (b) evolução temporal T2(t) do binário da junta 2, (c) evolução temporal x(t) da posição x, (d) evolução temporal y(t) da posição y, para o controlador VSS-SOM-PDD2 com os parâmetros da tabela 4.4 (i.e. $f_n=10\text{Hz}$ ), considerando a vermelho a trajectória planeada e cinzento a simulação.	87
Figura 4.15	- (a) evolução temporal T1(t) do binário da junta 1, (b) evolução temporal T2(t) do binário da junta 2, (c) evolução temporal x(t) da posição x, (d) evolução temporal y(t) da posição y, para o controlador VSS-FOM-PDD2 com os parâmetros da tabela 4.5 (i.e. $f_n=100\text{Hz}$ ), considerando a vermelho a trajectória planeada e cinzento a simulação.	88
Figura 4.16	(a) evolução temporal T1(t) do binário da junta 1, (b) evolução temporal T2(t) do binário da junta 2, (c) evolução temporal x(t) da posição x, (d) evolução temporal y(t) da posição y, para o controlador VSS-SOM-PID com os parâmetros da tabela 4.4 (i.e. $f_n=10\text{Hz}$ ), considerando a vermelho a trajectória planeada e azul a simulação.	88
Figura 4.17	- (a) evolução temporal T1(t) do binário da junta 1, (b) evolução temporal T2(t) do binário da junta 2, (c) evolução temporal x(t) da posição x, (d) evolução temporal y(t) da posição y, para o controlador VSS-SOM-PID com os parâmetros da tabela 4.5 (i.e. $f_n=100\text{Hz}$ ), considerando a vermelho a trajectória planeada e azul a simulação.	89
Figura 4.18	- (a) evolução temporal T1(t) do binário da junta 1, (b) evolução temporal T2(t) do binário da junta 2, (c) evolução temporal x(t) da posição x, (d) evolução temporal y(t) da posição y, para o controlador Feedforward com os parâmetros da tabela 4.4 (i.e. $f_n=10\text{Hz}$ ), considerando a vermelho a trajectória planeada e a azul a simulação.	90

	claro a simulação.	
Figura 4.19	- (a) evolução temporal T1(t) do binário da junta 1, (b) evolução temporal T2(t) do binário da junta 2, (c) evolução temporal x(t) da posição x, (d) evolução temporal y(t) da posição y, para o controlador Computed Torque com os parâmetros da tabela 4.2 (i.e. $f_n=10\text{Hz}$ ), considerando a vermelho o trajectória planeada e roxo a simulação.	90
Figura 4.20	- (a) evolução temporal T1(t) do binário da junta 1, (b) evolução temporal T2(t) do binário da junta 2, (c) evolução temporal x(t) da posição x, (d) evolução temporal y(t) da posição y, para o controlador Computed Torque com os parâmetros da tabela 4.3 (i.e. $f_n=100\text{Hz}$ ), considerando a vermelho a trajectória planeada e cor de rosa a simulação.	91
Figura 4.21	- (a) evolução temporal T1(t) do binário da junta 1, (b) evolução temporal T2(t) do binário da junta 2, (c) evolução temporal x(t) da posição x, (d) evolução temporal y(t) da posição y, para o controlador PD pelo método Jacobiano Inverso com os parâmetros da tabela 4.3 (i.e. $f_n=10\text{Hz}$ ), considerando a vermelho a trajectória planeada e a verde claro a simulação.	92
Figura 4.22	- (a) evolução temporal T1(t) do binário da junta 1, (b) evolução temporal T2(t) do binário da junta 2, (c) evolução temporal x(t) da posição x, (d) evolução temporal y(t) da posição y, para o controlador PD pelo método Jacobiano Transposto com os parâmetros da tabela 4.3 (i.e. $f_n=10\text{Hz}$ ), considerando a vermelho a trajectória planeada e preto a simulação.	92
Figura 4.23	- (a) evolução temporal T1(t) do binário da junta 1, (b) evolução temporal T2(t) do binário da junta 2, (c) evolução temporal x(t) da posição x, (d) evolução temporal y(t) da posição y, para o controlador VSS-FOM-PDD2 pelo método Jacobiano Inverso com os parâmetros da tabela 4.4 (i.e. $f_n=10\text{Hz}$ ), considerando a vermelho a trajectória planeada e cor de rosa a simulação.	93
Figura 4.24	- (a) evolução temporal T1(t) do binário da junta 1, (b) evolução temporal T2(t) do binário da junta 2, (c) evolução temporal x(t) da posição x, (d) evolução temporal y(t) da posição y, para o controlador VSS-FOM-PDD2 pelo método Jacobiano Inverso com os parâmetros da tabela 4.5 (i.e. $f_n=100\text{Hz}$ ), considerando a vermelho a trajectória planeada e roxo a simulação.	94
Figura 4.25	- (a) evolução temporal T1(t) do binário da junta 1, (b) evolução temporal T2(t) do binário da junta 2, (c) evolução temporal x(t) da posição x, (d) evolução temporal y(t) da posição y, para o controlador VSS-FOM-PDD2 pelo método Jacobiano Transposto com os parâmetros da tabela 4.4 (i.e. $f_n=10\text{Hz}$ ), considerando a vermelho a trajectória planeada e preto a simulação.	94
Figura 4.26	- (a) evolução temporal T1(t) do binário da junta 1, (b) evolução temporal T2(t) do binário da junta 2, (c) evolução temporal x(t) da posição x, (d) evolução temporal y(t) da posição y, para o controlador VSS-FOM-PDD2 pelo método Jacobiano Transposto com os parâmetros da tabela 4.5 (i.e. $f_n=100\text{Hz}$ ), considerando a vermelho a trajectória planeada e preto a simulação.	95
Figura 4.27	- (a) evolução temporal T1(t) do binário da junta 1, (b) evolução temporal T2(t) do binário da junta 2, (c) evolução temporal x(t) da posição x, (d) evolução temporal y(t) da posição y, para o controlador baseado no modelo do robô no espaço cartesiano com os parâmetros da tabela 4.4 (i.e. $f_n=10\text{Hz}$ ).	96
Figura 4.28	- (a) evolução temporal T1(t) do binário da junta 1, (b) evolução temporal T2(t) do binário da junta 2, (c) evolução temporal x(t) da posição x, (d) evolução temporal y(t) da posição y, para o controlador PD com os parâmetros da tabela 4.2 (i.e. $f_n=10\text{Hz}$ ) e o controlador VSS-FOM-PDD2 com os parâmetros da tabela 4.4 (i.e. $f_n=10\text{Hz}$ ).	97
Figura 4.29	- (a) evolução temporal da trajectória y(x), (b) ampliação evolução temporal da trajectória y(x), (c) evolução temporal x(t) da posição x, (d) evolução temporal y(t) da posição y, para o controlador PD com os parâmetros da tabela 4.2 (i.e. $f_n=10\text{Hz}$ ) e o controlador VSS-FOM-PDD2 com os parâmetros da tabela 4.4 (i.e. $f_n=10\text{Hz}$ ).	98
Figura 4.30	- (a) evolução temporal da trajectória y(x), (b) evolução temporal T1(t) do binário da junta 1, (c) evolução temporal x(t) da posição x, (d) evolução temporal y(t) da posição y, para o controlador PD com os parâmetros da tabela 4.2 (i.e. $f_n=10\text{Hz}$ ) e o controlador VSS-FOM-PDD2 com os parâmetros da tabela 4.4 (i.e. $f_n=10\text{Hz}$ ).	99
Figura 4.31	- (a) evolução temporal da trajectória y(x), (b) evolução temporal T1(t) do binário	101

	da junta 1, (c) evolução temporal $x(t)$ da posição $x$ , (d) evolução temporal $y(t)$ da posição $y$ , para a simulação 1 os ganhos de posição e de velocidade são nulos, na simulação 2 os ganhos de posição de velocidade são baixos e para a simulação 3 os ganhos de posição de velocidade são elevados.	
Figura 4.32	Esquema das simulações para os algoritmos Computed Torque e Feedforward.	101
Figura 4.33	- (a) evolução temporal da trajectória $y(x)$ , (b) evolução temporal $T1(t)$ do binário da junta 1, (c) evolução temporal $x(t)$ da posição $x$ , (d) evolução temporal $y(t)$ da posição $y$ , para a simulação 1 os ganhos de posição e de velocidade são nulos, na simulação 2 os ganhos de posição de velocidade são baixos e para a simulação 3 os ganhos de posição de velocidade são elevados.	102
Figura 4.34	- (a) evolução temporal da trajectória $y(x)$ , (b) evolução temporal $T1(t)$ do binário da junta 1, (c) evolução temporal $x(t)$ da posição $x$ , (d) evolução temporal $y(t)$ da posição $y$ , para a simulação 1 os ganhos de posição e de velocidade são nulos, na simulação 2 os ganhos de posição de velocidade são baixos e para a simulação 3 os ganhos de posição de velocidade são elevados.	102
Figura 4.35	- Esquema de simulações de fenómenos dinâmicos nas juntas.	103
Figura 4.36	- (a) evolução temporal $T1(t)$ do binário da junta 1, (b) evolução temporal $T2(t)$ do binário da junta 2, (c) evolução temporal $x(t)$ da posição $x$ , (d) evolução temporal $y(t)$ da posição $y$ , para o controlador VSS-FOM-PDD2 com os parâmetros da tabela 4.4 (i.e. $f_n=10\text{Hz}$ ) e para o robô com os parâmetros da tabela 4.6, considerando a vermelho a trajectória planeada e azul a simulação.	105
Figura 4.37	- (a) evolução temporal $T1(t)$ do binário da junta 1, (b) evolução temporal $T2(t)$ do binário da junta 2, (c) evolução temporal $x(t)$ da posição $x$ , (d) evolução temporal $y(t)$ da posição $y$ , para o controlador VSS-FOM-PDD2 com os parâmetros da tabela 4.5 (i.e. $f_n=100\text{Hz}$ ) e para o robô com os parâmetros da tabela 4.6, considerando a vermelho a trajectória planeada e azul a simulação.	105
Figura 4.38	- Para $f_n=100\text{Hz}$ , (a) evolução temporal $T1(t)$ do binário da junta 1, (b) evolução temporal $T2(t)$ do binário da junta 2, (c) evolução temporal $x(t)$ da posição $x$ , (d) evolução temporal $y(t)$ da posição $y$ , para o controlador VSS-FOM-PDD2 com os parâmetros da tabela 4.5 (i.e. $f_n=100\text{Hz}$ ) e para o robô com os parâmetros da tabela 4.7, considerando a vermelho a trajectória planeada e azul a simulação.	106
Figura 4.39	- (a) evolução temporal $T1(t)$ do binário da junta 1, (b) evolução temporal $T2(t)$ do binário da junta 2, (c) evolução temporal $x(t)$ da posição $x$ , (d) evolução temporal $y(t)$ da posição $y$ , para o controlador VSS-FOM-PDD2 com os parâmetros da tabela 4.4 (i.e. $f_n=10\text{Hz}$ ) e para o robô com os parâmetros da tabela 4.8, considerando a vermelho a trajectória planeada e azul a simulação.	107
Figura 4.40	- (a) evolução temporal $T1(t)$ do binário da junta 1, (b) evolução temporal $T2(t)$ do binário da junta 2, (c) evolução temporal $x(t)$ da posição $x$ , (d) evolução temporal $y(t)$ da posição $y$ , para o controlador VSS-FOM-PDD2 com os parâmetros da tabela 4.4 (i.e. $f_n=10\text{Hz}$ ) e para o robô com os parâmetros da tabela 4.9, considerando a vermelho a trajectória planeada e azul a simulação.	107
Figura 4.41	- (a) evolução temporal $T1(t)$ do binário da junta 1, (b) evolução temporal $T2(t)$ do binário da junta 2, (c) evolução temporal $x(t)$ da posição $x$ , (d) evolução temporal $y(t)$ da posição $y$ , para o controlador VSS-FOM-PDD2 com os parâmetros da tabela 4.4 (i.e. $f_n=10\text{Hz}$ ) e para o robô com os parâmetros da tabela 4.10, considerando a vermelho a trajectória planeada e azul a simulação.	108
Figura 4.42	- (a) evolução temporal $T1(t)$ do binário da junta 1, (b) evolução temporal $T2(t)$ do binário da junta 2, (c) evolução temporal $x(t)$ da posição $x$ , (d) evolução temporal $y(t)$ da posição $y$ , para o controlador VSS-FOM-PDD2 com os parâmetros da tabela 4.5 (i.e. $f_n=100\text{Hz}$ ) e para o robô com os parâmetros da tabela 4.10, considerando a vermelho a trajectória planeada e azul a simulação.	108
Figura 4.43	- Modelo da saturação nos actuadores.	109
Figura 4.44	- (a) evolução temporal $T1(t)$ do binário da junta 1, (b) evolução temporal $T2(t)$ do binário da junta 2, (c) evolução temporal $x(t)$ da posição $x$ , (d) evolução temporal $y(t)$ da posição $y$ , para o controlador VSS-FOM-PDD2 com os parâmetros da tabela 4.4 (i.e. $f_n=10\text{Hz}$ ) e para o robô com os parâmetros da tabela 4.11, considerando a vermelho a trajectória planeada e azul a simulação.	109
Figura 4.45	- Modelo da quantificação finita dos sensores de posição.	110
Figura 4.46	- (a) evolução temporal $T1(t)$ do binário da junta 1, (b) evolução temporal $T2(t)$	110

	do binário da junta 2, (c) evolução temporal $x(t)$ da posição $x$ , (d) evolução temporal $y(t)$ da posição $y$ , para o controlador VSS-FOM-PDD2 com os parâmetros da tabela 4.4 (i.e. $f_n=10\text{Hz}$ ) e para um sensor com $n=3$ níveis de quantificação das posições angulares, considerando a vermelho a trajectória planeada e azul a simulação.	
Figura 4.47	- (a) evolução temporal $T1(t)$ do binário da junta 1, (b) evolução temporal $T2(t)$ do binário da junta 2, (c) evolução temporal $x(t)$ da posição $x$ , (d) evolução temporal $y(t)$ da posição $y$ , para o controlador VSS-FOM-PDD2 com os parâmetros da tabela 4.4 (i.e. $f_n=10\text{Hz}$ ) e para um sensor com $n=4$ níveis de quantificação das posições angulares, considerando a vermelho a trajectória planeada e azul a simulação.	111
Figura 4.48	- Robô RP ideal.	112
Figura 4.49	- Trajectória do Robô RP.	113
Figura 4.50	- (a) evolução da trajectória $y(x)$ , (b) evolução temporal $x(t)$ da posição do eixo $x$ , (c) evolução temporal $y(t)$ da posição do eixo $y$ , (d) evolução temporal $T1(t)$ do binário da junta 1, para todos os controladores considerou-se a $f_n=10\text{Hz}$ .	114
Figura 4.51	- Controlador híbrido.	115
Figura 4.52	- Movimento do robô na superfície de restrição.	116
Figura 4.53	- (a) evolução temporal $T1(t)$ do binário da junta 1 (b) evolução temporal $T2(t)$ do binário da junta 2, (c) evolução temporal $x(t)$ da posição do eixo $x$ , (d) evolução temporal $y(t)$ da posição do eixo $y$ , para $f_n=10\text{Hz}$ .	117
Figura 4.54	- Robô RR e superfície de restrição.	118
Figura 4.55	- (a) evolução temporal $F1(t)$ , (b) evolução temporal $F2(t)$ , (c) evolução temporal $x(t)$ da posição do eixo $x$ , (d) evolução temporal $y(t)$ da posição do eixo $y$ , para os parâmetros do controlador na tabela 4.19 e da superfície de restrição na tabela 4.20.	119
Figura 4.56	- (a) evolução temporal $F1(t)$ , (b) evolução temporal $F2(t)$ , (c) evolução temporal $F1(t)$ , (d) evolução temporal $F2(t)$ , para os parâmetros do controlador na tabela 4.19 e da superfície de restrição na tabela 4.21. As figuras a) e b) são para $f_n=10\text{Hz}$ e as figuras c) e d) para $f_n=100\text{Hz}$ .	119
Figura 4.57	- (a) evolução temporal $F1(t)$ , (b) erro $e_p$ de seguimento de trajectória $y(x)$ , (c) evolução temporal $x(t)$ da posição $x$ , (d) evolução temporal $y(t)$ da posição $y$ , para o robô ideal, com o controlador de posição da tabela 4.23 e para o controlador de força da tabela 4.24.	121
Figura 4.58	- (a) evolução temporal $F1(t)$ , (b) erro $e_p$ de seguimento de trajectória $y(x)$ , (c) evolução temporal $x(t)$ da posição $x$ , (d) evolução temporal $y(t)$ da posição $y$ , para o robô com juntas flexíveis, com o controlador de posição da tabela 4.26 e para o controlador de força da tabela 4.27, considerando a vermelho a trajectória planeada e azul a simulação.	122
Figura 4.59	- (a) evolução temporal $F1(t)$ , (b) erro $e_p$ de seguimento de trajectória $y(x)$ , (c) evolução temporal $x(t)$ da posição $x$ , (d) evolução temporal $y(t)$ da posição $y$ , para o robô com atrito não linear, com o controlador de posição da tabela 4.29 e para o controlador de força da tabela 4.30, considerando a vermelho a trajectória planeada e preto a simulação.	123
Figura 4.60	- (a) evolução temporal $F1(t)$ , (b) erro $e_p$ de seguimento de trajectória $y(x)$ , (c) evolução temporal $x(t)$ da posição $x$ , (d) evolução temporal $y(t)$ da posição $y$ , para o robô com folgas nas juntas, com o controlador de posição da tabela 4.32 e para o controlador de força da tabela 4.33, considerando a vermelho a trajectória planeada e azul a simulação.	124
Figura 5.1	- Dois robôs a trabalhar em cooperação.	128
Figura 5.2	- Movimentos aleatórios do robô com o cotovelo para cima no espaço das juntas $\{O, q1, q2\}$ dentro de uma determinada esfera e respectivo mapeamento para o espaço operacional $\{O, x, y\}$ .	130
Figura 5.3	- Elipse calculada através do mapeamento para um robô.	130
Figura 5.4	- Elipse obtida no mapeamento para um braço robô.	132
Figura 5.5	- Janelas correspondentes ao robô RR, com a configuração pretendida, a selecção dos parâmetros do robô, número de pontos, número de amostras e a dimensão pretendida da esfera no espaço das juntas para o cálculo da respectiva manipulabilidade, e opção relativa à grelha de pontos num dos espaços.	133
Figura 5.6	- Resultados gráficos correspondentes do cálculo da manipulabilidade para o robô	134

	RR, com diferentes comprimentos dos elos.	
Figura 5.7	- Janelas correspondentes ao robô RR, com a configuração pretendida, a seleção dos parâmetros do robô, número de pontos, número de amostras e a dimensão pretendida da esfera no espaço das juntas para o cálculo da respectiva manipulabilidade, e opção relativa à grelha de pontos num dos espaços.	134
Figura 5.8	- Resultados gráficos correspondentes do cálculo da manipulabilidade para dois robôs RR, distanciados de um metro, com a grelha no espaço das juntas.	135
Figura 5.9	- Resultados gráficos correspondentes do cálculo da manipulabilidade para dois robôs RR, a uma distância de um metro, com a grelha no espaço operacional.	135
Figura 5.10	- A figura apresenta a manipulabilidade do robô RR, com os parâmetros $l_1=1$ e $l_2=0,8$ , pelo método de Yoshikawa.	136
Figura 5.11	- Manipulabilidade do robô RR com $l_1=1$ e $l_2=0,8$ , obtida pelo método numérico para mil pontos distintos no espaço operacional do robô.	137
Figura 5.12	- Curvas de nível sobrepondo os resultados de Yoshikawa/Método numérico.	137
Figura 5.13	- Movimentos aleatórios do robô no espaço das juntas $\{O, q_1, q_2\}$ dentro de uma determinada esfera e respectivo mapeamento para o espaço operacional $\{O, x, y\}$ , robô com o cotovelo em baixo e a grelha de pontos no espaço das juntas $\{O, q_1, q_2\}$ .	138
Figura 5.14	- Movimentos aleatórios do robô no espaço das juntas $\{O, q_1, q_2\}$ dentro de uma determinada esfera e respectivo mapeamento para o espaço operacional $\{O, x, y\}$ , robô com o cotovelo em baixo, e a grelha de pontos no espaço operacional $\{O, x, y\}$ .	138
Figura 5.15	- Gráfico tridimensional e curvas de nível do robô RR com os parâmetros $l_1=1,5$ e $l_2=0,5$ para uma amostras de mil pontos no espaço operacional do robô.	139
Figura 5.16	- Gráfico tridimensional e curvas de nível do robô RR com os parâmetros $l_1=0,5$ e $l_2=1,5$ para uma amostras de mil pontos no espaço operacional do robô.	139
Figura 5.17	- Gráfico tridimensional e curvas de nível para o robô RR com os parâmetros $l_1=1$ e $l_2=1$ para uma amostra de mil pontos no espaço operacional do robô.	140
Figura 5.18	- Mapa de elipses de manipulabilidade para um robô RR com $l_1=1$ e $l_2=1$ com limitações nas juntas, para mil pontos distintos no espaço operacional do robô.	140
Figura 5.19	- Gráfico tridimensional e curvas de nível para o robô RR com os parâmetros $l_1=1$ e $l_2=$ com limitações nas juntas, para uma amostra de mil pontos no espaço operacional do robô .	140
Figura 5.20	- Dois robôs RR a trabalhar em cooperação.	141
Figura 5.21	- Medida de manipulabilidade conjunta de dois robôs RR no espaço operacional para vários valores de $b \in [0, 4[$ .	142
Figura 5.22	- Medida de manipulabilidade conjunta de dois robôs RR no espaço operacional para vários valores de $b \in [0, 4[$ .	143
Figura 5.23	- Curvas de nível da medida de manipulabilidade de dois robôs RR do espaço operacional de $b \in [0, 4[$ .	144
Figura 5.24	- Curvas de nível da medida de manipulabilidade de dois robôs RR do espaço operacional de $b \in [0, 4[$ .	145
Figura 5.25	- Elipse calculada através do mapeamento para dois braços.	145
Figura 5.26	Medida de manipulabilidade do espaço operacional a 3D de $b \in [0, 4[$ .	146
Figura 5.27	Medida de manipulabilidade do espaço operacional a 3D de $b \in [0, 4[$ .	147
Figura 5.28	- Curvas de nível da medida de manipulabilidade de dois robôs RR do espaço operacional de $b \in [0, 4[$ .	147
Figura 5.29	- Curvas de nível da medida de manipulabilidade de dois robôs RR do espaço operacional de $b \in [0, 4[$ .	148
Figura 5.30	- Medida de manipulabilidade conjunta de dois robôs RR no espaço operacional para vários valores de $b \in [0, 4[$ .	149
Figura 5.31	- Medida de manipulabilidade conjunta de dois robôs RR no espaço operacional para vários valores de $b \in [0, 4[$ .	150
Figura 5.32	- Curvas de nível da medida de manipulabilidade de dois robôs RR do espaço operacional de $b \in [0, 4[$ .	151
Figura 5.33	- Curvas de nível da medida de manipulabilidade de dois robôs RR do espaço operacional de $b \in [0, 4[$ .	152

Figura 5.34	– Manipulabilidade média para a base considerada.	153
Figura 5.35	– Manipulabilidade máxima.	154
Figura 5.36	– Volume médio de manipulabilidade.	154

# Lista de Tabelas

Tabela 4.1	- Parâmetros do robô RR com juntas ideais.	78
Tabela 4.2	- Parâmetros dos controladores lineares para $f_n= 10\text{Hz}$ .	80
Tabela 4.3	- Parâmetros dos controladores lineares para $f_n= 100\text{Hz}$ .	81
Tabela 4.4	- Parâmetros dos controladores não lineares para $f_n= 10\text{Hz}$ .	85
Tabela 4.5	- Parâmetros dos controladores não lineares para $f_n= 100\text{Hz}$ .	85
Tabela 4.6	- Parâmetros das folgas do robô.	104
Tabela 4.7	- Parâmetros das folgas do robô.	104
Tabela 4.8	- Parâmetros da flexibilidade nas juntas do robô.	104
Tabela 4.9	- Parâmetros da flexibilidade nas juntas do robô.	104
Tabela 4.10	- Parâmetros do atrito não linear das juntas do robô.	104
Tabela 4.11	- Parâmetros da saturação dos actuadores do robô.	104
Tabela 4.12	- Parâmetros do robô RP.	113
Tabela 4.13	- Parâmetros do controlador VSS-FOM-PDD2.	116
Tabela 4.14	- Parâmetros do controlador VSS-SOM-PID.	116
Tabela 4.15	- Parâmetros do controlador de força do tipo VSS-FOM-PDD2.	118
Tabela 4.16	- Parâmetros da superfície de restrição.	118
Tabela 4.17	- Parâmetros da superfície de restrição.	118
Tabela 4.18	- Controlador de posição do tipo VSS-FOM-PDD2.	120
Tabela 4.19	- Controlador de posição do tipo VSS-FOM-PID.	120
Tabela 4.20	- Controlador de força do tipo VSS-FOM-PDD2.	120
Tabela 4.21	- Parâmetros do robô com flexibilidade nas juntas.	122
Tabela 4.22	- Parâmetros do sub-controlador de posição.	122
Tabela 4.23	- Parâmetros do sub-controlador de força.	122
Tabela 4.24	- Parâmetros do robô com atrito nas juntas.	123
Tabela 4.25	- Parâmetros do sub-controlador de posição.	123
Tabela 4.26	- Parâmetros do sub-controlador de força.	123
Tabela 4.27	- Parâmetros do robô com folgas nas juntas.	124
Tabela 4.28	- Parâmetros do sub-controlador de posição.	124
Tabela 4.29	- Parâmetros do sub-controlador de força.	124



# Simbologia

$A_i$  - Constante de atrito no instante inicial

$B_i$  - Atrito viscoso.

$B_m$  - Factor de amortecimento e a rigidez do motor da transmissão

$B$ - Amortecimento

$b$ - Distância entre robôs

$C(q, \dot{q})$  - **Vector,  $n \times 1$  dimensional, dos binários/forças coriolis/centrípetos**

$F$ - Força de contacto

$G_x$ - Vector  $n \times 1$  dimensional, dos binários/forças gravitacionais.

$J(q)$ - Matriz simétrica das inércias, de dimensão  $n \times n$

$J_c^{-T}$  - Vector  $n \times n$  do jacobiano transposto inverso

$J_c^{-1}(q)$  - Vector  $n \times n$  do jacobiano inverso

- Jacobiano

$J$

$J(q)$ - Matriz simétrica das inércias, de dimensão  $n \times n$

$J_{ig}$  - Inércia após a engrenagem

$J_{im}$  - Inércia do motor

$J_{im}$  - Inércia do motor

**RR**- Robô com dois eixos rotacionais

**RP**- Robô com um eixo rotacional e um eixo prismático

$l_i$ - Comprimento dos elos

$P$ - É a energia potencial

$q_i$ - Posição do eixo

$\dot{q}_i$  - Velocidade no eixo

$\ddot{q}_i$  - Aceleração no eixo

$q_{im}$  - Posição do motor.

$\dot{q}'_i$  - Velocidade das inércias das juntas depois da colisão.

$\dot{q}'_{im}$  - Velocidade do motor depois da colisão.

$\varepsilon$ - Constante de *Newton* que define a elasticidade do impacto

$K$ - É a energia cinética

$K_m$  - Factor de rigidez do motor e da transmissão

$K_1$ - Representa o ganho de velocidade  
 $K_2$ - Representa o ganho de posição  
 $K$ - Rigidez, respectivamente.  
 $K_i$  - Atrito de Coulomb  
 $t$  - Valor da variável tempo  
 $T$ - Representa o vector  $n \times 1$  dos binários dos actuadores  
 $T_{atrito}$ - Binário de atrito  
 $T_{max}$ - Binário máximo  
 $T_{min}$ - Binário mínimo  
 $T_{min}$ - Binário mínimo  
**PID**- Proporcional Integral Derivativo  
 $u$ - Representa o erro  
**CEV**- Controlador de estrutura variável  
**MPO**- Modelo de primeira ordem  
**MSO**- Modelo de segunda ordem  
 $M_x$ - Matriz das inércias de dimensão  $n \times n$ ,  
 $M$ - Massa  
 $\sigma$ - Superfície de deslizamento  
 $\xi$ - Coeficiente de amortecimento  
 $w_n$ - Frequência natural não amortecida  
 $F$ - Vector  $n \times 1$  de força nos actuadores que actuam no robô  
 $x$ - Vector  $n \times 1$  que representa a posição e orientação do manipulador no espaço cartesiano  
 $V_x$ - Vector  $n \times 1$  dimensional, dos binários/forças coriolis/centrípetos  
 $\gamma$ - Vector  $n \times 1$  dos binários dos actuadores.  
 $\mu$ - Manipulabilidade

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES

*Simulação Dinâmica e Controlo de Robôs  
Industriais*

Tese de Mestrado

Nuno Miguel Fonseca Ferreira

Licenciado em Engenharia Electrotécnica e de Computadores

Porto  
Setembro 1999

Dissertação submetida ao Departamento de Engenharia Electrotécnica e de Computadores da Universidade do Porto satisfazendo parcialmente os requisitos para a obtenção do grau de mestre.

*Trabalho desenvolvido sob a orientação do Professor Doutor José António Tenreiro Machado – Professor Coordenador do Departamento de Engenharia. Electrotécnica do Instituto Superior de Engenharia do Porto e sob a co-orientação do Professor Doutor Martins de Carvalho – Professor Catedrático do Departamento de Engenharia Electrotécnica e de Computadores da Universidade do Porto.*

Dedico esta dissertação  
aos meus Pais  
às minhas irmãs Catarina e Joana  
e à minha namorada Anabela

# Agradecimentos

Desejo dar os meus agradecimentos:

- Ao Prof. Dr. Tenreiro Machado, que me disponibilizou o seu apoio, orientação, e material necessário para efectuar este trabalho;
- À minha família, em especial aos meus pais e irmãs, o constante apoio e encorajamento;
- À minha namorada, pelo carinho, pela compreensão e paciência durante todo este tempo;
- Ao ISEC, onde eu trabalho, que me deu o suporte financeiro para efectuar esta tese;
- A todo o pessoal do ISEC.



## Resumo

Neste trabalho é apresentado o programa *RobLib*. O programa foi desenvolvido no ambiente Windows e consiste numa biblioteca de robôs industriais. As funcionalidades do programa consistem em efectuar o planeamento de trajectórias, quer no espaço das juntas quer no espaço cartesiano, utilizando uma interpolação linear e efectuar a simulação de vários fenómenos dinâmicos nas juntas dos robôs, assim como a simulação de diversas estratégias de controlo de posição e de força.

Através da integração de uma base de dados no programa de simulação RobLib, pode-se comparar as simulações dos diversos algoritmos, assim como os vários tipos de fenómenos nas juntas.

O trabalho é complementado com um estudo sobre a manipulabilidade de um braço robô e de dois braços robôs a trabalhar em cooperação, na manipulação de pequenos objectos.

## **Abstract**

In this work is presented the *RobLib* software which was developed in the Windows environment. This software consists of a library for industrial robots. This software allows trajectory planning algorithms both in the joint or cartesian space using a linear interpolation procedure, allows the simulation of dynamical phenomena in the robot joints and the simulation of several algorithms of control involving position and force.

Through the integration of a database in the *RobLib* software we can compare the diferents control algorithms, as well the diferents dynamical phenomena in the joints of the robots.

This work is complemented with a study about the manipulability for a single arm manipulators and two robotic arms working in cooperation, in the manipulation of small objects.

## Résumé

Dans ce travail, le programme RobLib est présenté. Le programme a été développé dans l'ambiance Windows et consiste d'une bibliothèque de robots industriels. Les fonctionnalités du programme consistent en effectuer la planification de trajectoires, soit dans l'espace des joints, soit dans l'espace cartésien, tout en utilisant une interpolation linéaire et effectuer la simulation de plusieurs phénomènes dynamiques, comme la simulation de divers algorithmes de contrôle de position/force.

À travers l'intégration d'une datatabase au programme de simulation *RobLib*, on peut comparer les simulations de plusieurs algorithmes, comme les divers phénomènes dynamiques aux jointes

En complément, le travail est ajouté d'une étude sur la manipulabilité d'un bras robot et de deux bras robot en travaillant en coopération, à la manipulation de dimensions réduites.



# CAPÍTULO 1

## 1.1 Introdução

A automação e a robótica são áreas de investigação muito activas no desenvolvimento de sistemas que permitem reduzir ou eliminar a necessidade de intervenção humana em ambientes potencialmente perigosos, ou mesmo inacessíveis, ao homem. As aplicações práticas mais numerosas verificam-se ao nível dos robôs manipuladores industriais, com especial destaque para a indústria automóvel, nomeadamente com a utilização dos robôs para a pintura, para a montagem e para soldadura. Contudo, as numerosas aplicações já desenvolvidas revelam, sem margem para dúvidas, as fortes limitações ainda existentes e, conseqüentemente, os numerosos aspectos ainda inexplorados que se torna necessário estudar para se conseguirem manipuladores mais eficientes. Nesta perspectiva, desenvolveu-se o programa *RobLib* no ambiente *Windows*, como uma ferramenta útil para a análise dinâmica dos robôs e para o estudo de vários algoritmos de controlo de manipuladores robóticos.

## 1.2 Objectivos

Os objectivos deste trabalho consistem em :

Fazer uma análise dinâmica de robô industriais. Todos os sistemas estudados têm como base o robô com duas juntas, isto é, com dois graus de liberdade. Esta estratégia

tem sido seguida por muitos investigadores, pois permite uma fácil implementação e interpretação, sem contudo perder generalidade, já que, para os fins em vista, se verificam todos os fenómenos cinemáticos e dinâmicos que aparecem em robôs manipuladores com um maior número de graus de liberdade.

Desenvolver um programa intitulado *RobLib* (*Robot Library*), que permite a simulação de robôs e de algoritmos de controlo. O programa disponibiliza robôs de duas juntas (com estruturas RR e RP) ideais ou com fenómenos dinâmicos nas juntas, assim como um conjunto de algoritmos de controlo de posição e de força.

Desenvolver uma biblioteca de manipuladores robóticos, de modo a efectuar o estudo dinâmico dos robôs e de diversos algoritmos de controlo. Através da integração de uma base de dados no programa de simulação *RobLib*, pode-se comparar as simulações dos diversos algoritmos, assim como os vários tipos de fenómenos nas juntas.

Criar uma ferramenta numérica que permita o cálculo e a visualização da manipulabilidade de sistemas robóticos com um ou com dois braços no seu espaço de trabalho.

## **1.3 Estrutura da dissertação**

A presente dissertação é composta por seis capítulos, incluindo este, onde é descrito o trabalho efectuado, bem como o conteúdo dos capítulos que se seguem.

No capítulo 2 é feito um estudo da modelização de robôs manipuladores de dois eixos, a partir da análise matricial e simbólica. Desta forma, estabelecem-se as equações da cinemática directa de posição, de velocidade e de aceleração. De uma forma idêntica, obtêm-se as equações relativas à cinemática inversa de posição, de velocidade e de aceleração. Com base na cinemática apresenta-se o planeamento de trajectórias quer no espaço das juntas quer no espaço cartesiano. De seguida, desenvolveu-se a modelização das dinâmicas directa e inversa e dos fenómenos dinâmicos nas juntas dos robôs. Por último são apresentados diversos algoritmos de controlo de posição e de força.

No capítulo 3 é apresentado o programa desenvolvido, intitulado *RobLib*, que permite a simulação de robôs e o estudo do desempenho de algoritmos de controlo. As suas funcionalidades são ilustradas utilizando robôs de dois eixos. Neste simulador foi integrado uma base de dados que permite armazenar as simulações. Deste modo torna-se possível efectuar comparações aos diversos tipos de algoritmos de controlo, assim como aos diversos tipos de fenómenos dinâmicos nas juntas

No capítulo 4 apresentam-se os resultados obtidos nas simulações com o programa, *RobLib*. Desta forma, faz-se um estudo comparativo dos diversos algoritmos de controlo apresentados no capítulo 2, assim como da influência dos diversos tipos de fenómenos dinâmicos nas juntas dos robôs. Apresentam-se também diversas simulações do algoritmo controlo híbrido de modo a analisar os efeitos dos seus subcontroladores e estudar várias opções de controlo. Apresentam-se também os efeitos de diferentes modelos da restrição do ambiente.

No capítulo 5 apresenta-se um estudo paramétrico de modo a testar a medida da manipulabilidade para o robô RR, e, para dois robôs RR a trabalhar em cooperação. São apresentadas interpretações desta medida, utilizando técnicas gráficas, mostrando-se o espaço de trabalho e a sua manipulabilidade.

Por último, no capítulo 6 são apresentadas as conclusões que decorrem do trabalho e são indicadas as perspectivas de desenvolvimento futuro.



## CAPÍTULO 2

### 2.1 Introdução

Um manipulador robótico pode ser modelizado como uma cadeia articulada de elos rígidos interligados por juntas movidas por actuadores, juntas essas que podem ser rotacionais ou prismáticas. Uma das extremidades da cadeia está ligada a uma base de suporte, enquanto que a outra é livre e possui um órgão terminal para a manipulação de objectos e a realização de tarefas.

O movimento combinado das juntas resulta no movimento dos elos que posicionam o órgão terminal. Nesta perspectiva, a modelização requer o estabelecimento da cinemática e da dinâmica. A cinemática trata do estudo analítico da geometria do movimento de um braço robótico. A dinâmica relaciona as posições, velocidades e acelerações que ocorrem numa trajectória com as forças/binários fornecidos pelos actuadores. Ambos os níveis da modelização têm, a importância na simulação, na análise e na concepção mecânica dos manipuladores. Todavia, os robôs industriais apresentam fenómenos complexos que tornam difícil a sua análise. Desta forma, neste trabalho apresentam-se alguns aspectos relativos à sua modelização, tais como, atritos, folgas nas engrenagens e flexibilidades nas juntas. São também modelizados limitações dos actuadores e dos sensores.

Uma vez desenvolvido um modelo é necessário considerar o controlo do robô nomeadamente, as variáveis de posição/velocidade e as de força.

O controlo de posição tem o objectivo de assegurar que uma sequência de movimentos planeados seja executada correctamente, face a eventuais erros resultantes

das limitações computacionais e efeitos mecânicos. As estratégias de controlo de posição são adequadas para tarefas onde o manipulador não interaccua significativamente com os objectos na região de trabalho, tais como, a transferência de materiais, a pintura ou a soldadura por pontos.

O controlo de posição/força é necessário em trabalhos que envolvam o controlo de posição segundo certas direcções no espaço de trabalho e um controlo de força de contacto entre o órgão terminal do robô e a superfície de restrição nas restantes direcções. As estratégias de controlo de posição/força são adequadas para tarefas tais como a montagem, o polimento e a rebarbagem, que envolvem contacto. Nestes casos deve adoptar-se uma estratégia de controlo das forças de interacção entre o manipulador e o ambiente.

Este capítulo, está estruturado em oito secções. Assim, na segunda secção, faz-se uma apresentação dos robôs a estudar e mostra-se a sua região de trabalho. Na terceira secção apresenta-se a cinemática directa e inversa de posição, velocidade e aceleração. De seguida, na quarta secção mostra-se a implementação de movimentos rectilíneos utilizando a interpolação linear entre dois pontos com diferentes perfis de velocidade. Na quinta secção desenvolveu-se as dinâmicas directa e inversa. Na sexta secção, abordam-se os fenómenos das junta. A partir dos modelos nas secções sete e oito faz-se uma análise do controlo de robôs, respectivamente, ao nível das variáveis posição, velocidade e posição/força.

## 2.2 Manipuladores Robóticos

Os manipuladores mecânicos são constituídos por diversos elos, supostos rígidos, interligados por articulações rotacionais ( R ) e lineares ou prismáticas ( P ).

O número, o tipo de articulações e a estrutura mecânica adoptada para cada manipulador dão origem a características e desempenhos distintos. As características do manipulador robótico vem influenciadas pelo tipo de estrutura do braço. Deste modo, neste estudo, considerou-se um robô do tipo **RR**, isto é, com dois eixos rotacionais, (ou **2R**) e o robô do tipo **RP**, com um eixo rotacional e um eixo prismático.

Considera-se então um braço robô **RR** ideal (*i.e.* sem fenómenos “adicionais” nos eixos), esquematizado na figura 2.1, que contém duas juntas rotativas, cuja posição é definida pelos ângulos  $q_1$  e  $q_2$ .

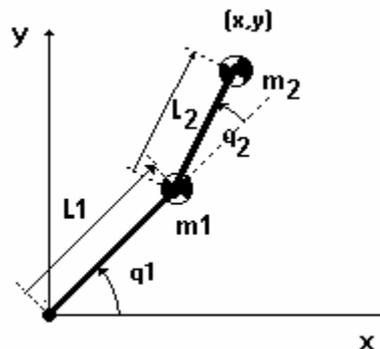


Figura 2.1 – Robô RR.

Faz-se também o estudo de um robô **RP** ideal (*i.e.* sem fenómenos “adicionais” nos eixos), representado na figura 2.2, que contém uma junta rotativa e uma junta prismática, cuja posição é definida pelo ângulo  $\theta$  e pelo deslocamento  $r$ .

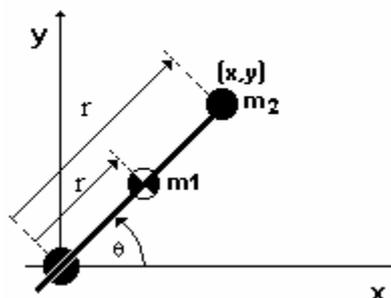


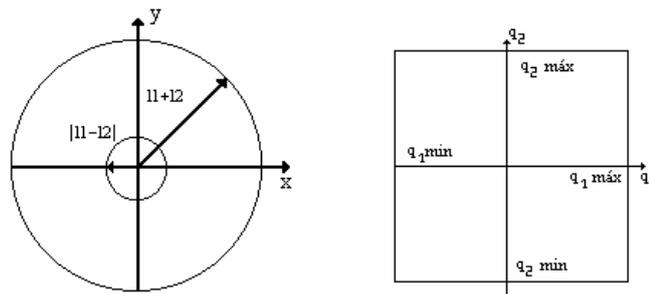
Figura 2.2 – Robô RP.

### 2.2.1 Região de trabalho “WorkSpace”

A região de trabalho corresponde à zona onde o robô pode manipular as suas ferramentas de trabalho. A área de trabalho do robô é determinada pelas seguintes características [1]:

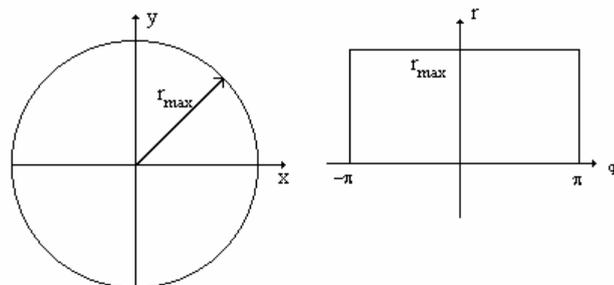
- estrutura cinemática do robô;
- comprimento dos seus elos;
- limitações dos movimentos das juntas.

Para um robô RR com comprimentos dos braços iguais a  $l_1 = l_2 = 1$ , a região de trabalho é uma circunferência de raio  $2 \cdot 1$ , como se vê na figura 2.3. Todavia, se os comprimentos dos elos forem diferentes e se os ângulos variarem  $2\pi$  rad, a área de trabalho fica limitada por dois círculos com raios  $l_1 + l_2$  e  $|l_1 - l_2|$ , como se observa na figura 2.3.



**Figura 2.3** –Área de trabalho do robô RR com  $q_i \text{ min} = -\pi$  e  $q_i \text{ max} = \pi$ .

Para um robô RP com extensão máxima do braço igual a  $r_{\text{max}}$ , e a variação angular de  $2\pi$  rad, a região de trabalho também é uma circunferência de raio  $r_{\text{max}}$ .



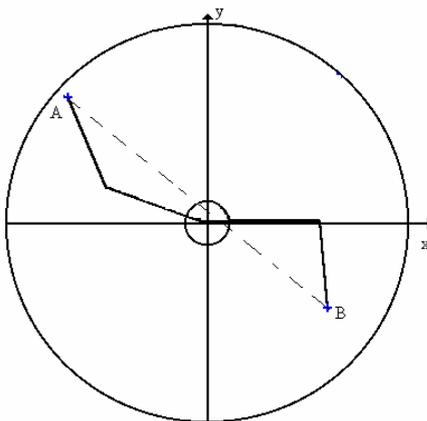
**Figura 2.4** –Área de trabalho do robô RP.

## 2.2.2 Problemas geométricos relacionados com o espaço de trabalho

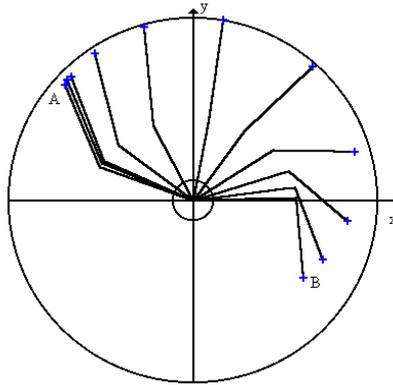
Até agora apresentou-se a região de trabalho do robô. No entanto, o planeamento de trajectórias trazem alguns problemas que se encontram relacionados com o espaço de trabalho [2].

As trajectórias no espaço cartesiano apresentam alguns problemas nomeadamente quando os pontos intermédios não são alcançáveis, quando ocorrem valores elevados das velocidades nas juntas, perto das singularidades do robô, e quando as trajectórias apresentam diferentes soluções.

Apesar das posições inicial e final do manipulador estarem dentro do espaço de trabalho, é possível que nem todos os pontos da trajectória rectilínea planeada sejam alcançáveis. Considere por exemplo o robô RR e seu espaço de trabalho associado que se encontra na figura 2.5. Neste caso, o elo 2 é menor que o elo 1, o que faz com que exista um espaço na região de trabalho que o robô não consiga alcançar (*i.e.* um “buraco” no meio com um raio de  $|l_1 - l_2|$ ). Estão desenhados no espaço de trabalho os pontos de partida e de chegada, respectivamente, A e B. O movimento rectilíneo de A para B não seria um problema no espaço das juntas (figura 2.6) contudo, no caso da trajectória planeada no espaço cartesiano, existem pontos intermédios ao longo da trajectória que não são alcançáveis.



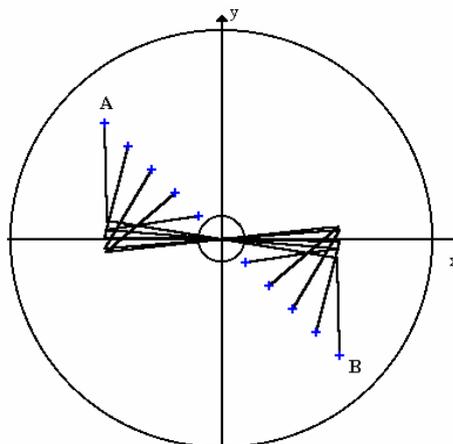
**Figura 2.5** – Pontos não alcançáveis na trajectória.



**Figura 2.6** – Planeamento da trajectória nas juntas.

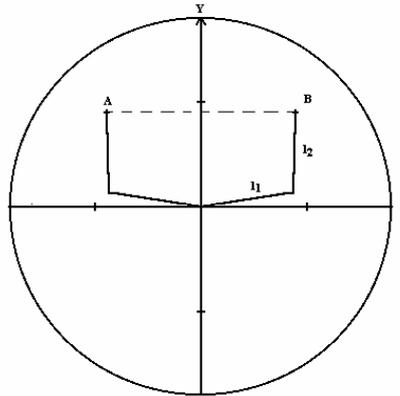
Existem locais no espaço de trabalho do manipulador onde é impossível promover a velocidade desejada do manipulador no espaço cartesiano através de valores finitos nas juntas. Se o manipulador seguir uma trajectória rectilínea no espaço cartesiano e se aproximar de uma configuração singular do mecanismo, uma ou mais velocidades nas juntas podem evoluir até ao infinito, o que torna impossível concretizar a trajectória desejada.

Considere-se um robô RR com elos iguais que se movem na trajectória do ponto A para o ponto B como se pode observar na figura 2.7, onde vários pontos intermédios do manipulador foram desenhados para ajudar a visualizar o movimento. Todos os pontos que se encontram na trajectória são alcançáveis mas, quando o robô passa no ponto  $(x, y) = (0,0)$ , surge a situação ilustrada na figura 2.7, *i.e.* mudança da estrutura do robô.



**Figura 2.7** – Singularidades do robô RR.

Outro tipo de problemas que podem surgir é se o ponto a atingir não poder ser alcançado com a mesma uma configuração idêntica à da posição inicial como se apresenta na figura 2.8 para um robô RR de elos iguais.



**Figura 2.8** – Trajectórias com diferentes soluções.

Os problemas referidos surgem, somente, com trajectórias especificadas no espaço cartesiano. Assim, a maioria dos sistemas de controlo dos manipuladores industriais suportam o planeamento de trajectória quer no espaço das juntas quer no espaço cartesiano.

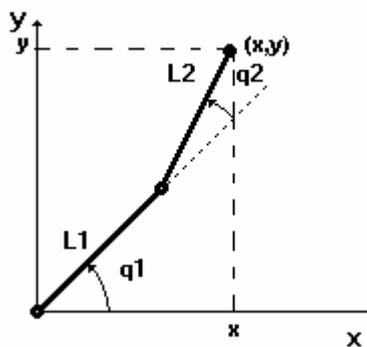
### 2.3 Introdução à cinemática

O estudo e o controlo dos manipuladores passa pela compreensão da cinemática.

Esta etapa da modelização trata as relações geométricas que envolvem as coordenadas na região de trabalho  $\{x, y\}$  e as variáveis nas articulações  $\{q_1, q_2\}$ . Assim, consoante o sentido da transformação de coordenadas surgem duas possibilidades: a cinemática directa  $\{q_1, q_2\} \rightarrow \{x, y\}$  e a cinemática inversa  $\{x, y\} \rightarrow \{q_1, q_2\}$ . Em qualquer dos casos pode-se estabelecer uma relação diferencial (em ordem ao tempo) de onde resulta a cinemática diferencial.

As figura 2.9 e 2.10 ilustram as cinemáticas directa e inversa para o robô RR.

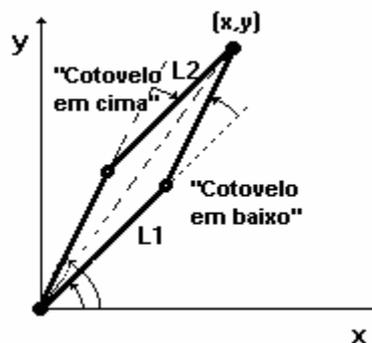
Nestas figuras surgem duas propriedades importantes: para a cinemática directa só existe uma solução, enquanto que na cinemática inversa podem existir várias soluções, se estas estiverem dentro da região de trabalho do braço. Por exemplo, na figura 2.10 há duas soluções  $\{q_1, q_2\}$ , “cotovelo em cima” e “cotovelo em baixo”, quando a “mão” está numa determinada coordenada  $\{x, y\}$ .



**Figura 2.9** – Cinemática directa  $\{q_1, q_2\} \rightarrow \{x, y\}$  do robô RR.

Relativamente às cinemáticas directa e inversa, estas podem ser resolvidas por diversos métodos, nomeadamente, utilizando os ângulos de Euler, os ângulos de Bryant, os ângulos náuticos, os parâmetros de Euler ou as transformações homogêneas. A todos estes métodos está subjacente um conjunto de expressões que não se apresentam por saírem fora do âmbito deste estudo[2][3][4].

Assim, tendo em vista os objectivos do presente trabalho, que consiste no desenvolvimento e análise dinâmica de vários algoritmos de controlo, é unicamente desenvolvida uma análise matricial e simbólica da cinemática dos robôs com dois eixos num sistema de coordenadas  $O_{XY}$ .



**Figura 2.10** – Cinemática inversa  $\{x, y\} \rightarrow \{q_1, q_2\}$  do robô RR.

A análise diferencial será também tratada de modo a obter-se a relação entre as posições, as velocidades e as acelerações do manipulador.

### 2.3.1 Cinemática directa do robô RR

Definindo  $\mathbf{P}_i'$  como sendo o vector que vai da base do elo  $i$  até a sua extremidade, resulta :

$$\mathbf{P}_1' = l_1 \begin{bmatrix} \cos q_1 \\ \sin q_1 \end{bmatrix} \quad (1)$$

$$\mathbf{P}_2' = l_2 \begin{bmatrix} \cos (q_1 + q_2) \\ \sin (q_1 + q_2) \end{bmatrix} \quad (2)$$

Definindo  $\mathbf{P}_i$  como sendo o vector que vai da base do sistema de eixos  $O_{xy}$  até á extremidade do elo  $i$ , tem-se  $\mathbf{P}_1 = \mathbf{P}_1'$  e  $\mathbf{P}_2 = \mathbf{P}_1' + \mathbf{P}_2'$ . Assim, a cinemática directa do manipulador é dada por  $\mathbf{P}_2 = (x, y)$  ou, na forma matricial:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} l_1 \cdot \cos q_1 + l_2 \cdot \cos (q_1 + q_2) \\ l_1 \cdot \sin q_1 + l_2 \cdot \sin (q_1 + q_2) \end{bmatrix} \quad (3)$$

Uma vez calculada a derivada da expressão (3) em ordem ao tempo, chega-se à relação entre a velocidade das juntas e a velocidade cartesiana do manipulador. Assim, resulta a expressão:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \mathbf{J} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \quad (4.a)$$

$$\mathbf{J} = \begin{bmatrix} -l_1 \cdot \sin q_1 - l_2 \cdot \sin (q_1 + q_2) & -l_2 \cdot \sin (q_1 + q_2) \\ l_1 \cdot \cos q_1 + l_2 \cdot \cos (q_1 + q_2) & l_2 \cdot \cos (q_1 + q_2) \end{bmatrix} \quad (4.b)$$

onde a matriz  $\mathbf{J}$  toma o nome de jacobiano da relação (3).

Ao derivar a expressão (4), obtém-se as equações da aceleração:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} -l_1 \cdot \sin q_1 & -l_2 \cdot \sin (q_1 + q_2) \\ l_1 \cdot \cos q_1 & l_2 \cdot \cos (q_1 + q_2) \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_1 + \ddot{q}_2 \end{bmatrix} - \begin{bmatrix} l_1 \cdot \sin q_1 & l_2 \cdot \sin (q_1 + q_2) \\ l_1 \cdot \cos q_1 & l_2 \cdot \cos (q_1 + q_2) \end{bmatrix} \begin{bmatrix} \dot{q}_1^2 \\ (\dot{q}_1 + \dot{q}_2)^2 \end{bmatrix} \quad (5)$$

As expressões (4) e (5) constituem a cinemática diferencial directa de primeira e de segunda ordem do robô RR.

### 2.3.2 Cinemática directa do robô RP

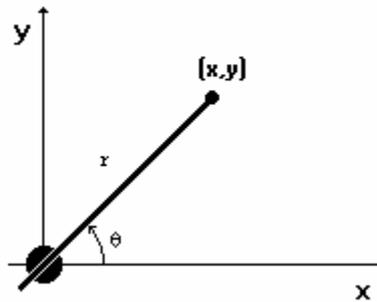


Figura 2.11– Cinemática do Robô RP.

Para um robô ser do tipo **RP** (figura 2.11) a cinemática directa, na forma matricial; consiste na seguinte relação.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} r \cos(\theta) \\ r \sin(\theta) \end{bmatrix} \quad (6)$$

Calculando a derivada de (6) em ordem ao tempo, chega-se à relação entre a velocidade das juntas e a velocidade cartesiana do manipulador. Assim, resulta a expressão seguinte:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \mathbf{J} \begin{bmatrix} \dot{r} \\ \dot{\theta} \end{bmatrix} \quad (7.a)$$

$$\mathbf{J} = \begin{bmatrix} \cos(\theta) & -r \sin(\theta) \\ \sin(\theta) & r \cos(\theta) \end{bmatrix} \quad (7.b)$$

onde **J** é o jacobiano correspondente a (6).

Ao derivar a expressão da velocidade (7), obtém-se as equações cinemáticas da aceleração:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -r \sin(\theta) \\ \sin(\theta) & r \cos(\theta) \end{bmatrix} \begin{bmatrix} \ddot{r} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} -2 \sin(\theta) & -r \cos(\theta) \\ 2 \cos(\theta) & -r \sin(\theta) \end{bmatrix} \begin{bmatrix} \dot{r} \dot{\theta} \\ \dot{\theta}^2 \end{bmatrix} \quad (8)$$

As expressões (7) e (8) representam a cinemática diferencial directa de primeira e de segunda ordem do robô RP.

### 2.3.3 Cinemática inversa do robô RR

Como já foi dito, a cinemática inversa envolve o cálculo dos ângulos das juntas para uma determinada posição do manipulador. Assim, poderão existir, zero, uma ou várias soluções. Caso o ponto desejado  $(x, y)$  estar fora da área de trabalho do braço robô, não é possível encontrar uma solução mas no caso do ponto  $(x, y)$  estar dentro do espaço de trabalho podem existir uma ou mais soluções. Neste caso, a inversão de (3) conduz à equação:

$$\begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} \tan^{-1}\left(\frac{y}{x}\right) - \tan^{-1}\left(\frac{l_2 \cdot \text{sen } q_2}{l_1 + l_2 \cdot \text{cos } q_2}\right) \\ \cos^{-1}\left(\frac{x^2 + y^2 - l_1^2 - l_2^2}{2 \cdot l_1 \cdot l_2}\right) \end{bmatrix} \quad (9)$$

A partir da forma matricial (9) conclui-se que existem duas soluções, dependendo do quadrante que se escolhe para  $q_2$ . Como  $\cos(\pi - q_2) = -\cos(q_2)$ , se o ângulo na junta  $q_2 \in [-\pi, 0]$ , então obtém-se a solução do tipo “cotovelo em cima”, enquanto que se o ângulo  $q_2 \in [0, \pi]$ , resulta uma solução do tipo “cotovelo em baixo”. Para cada um dos casos obtém-se um valor distinto de  $q_1$ .

Observando a relação matricial (9), e no caso de  $x = y = 0$  e  $l_1 = l_2$ , obtém-se uma solução indeterminada. Por outras palavras, no caso de se querer posicionar o manipulador na origem e  $l_1 = l_2$ , existem infinitas soluções, pois qualquer ângulo  $q_1$  é admissível.

As velocidades dos ângulos das juntas a partir das velocidades cartesianas, são obtidas encontrando o Jacobiano inverso ( $\mathbf{J}^{-1}$ ) da matriz (4):

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \quad (10.a)$$

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \frac{1}{l_1 l_2 \text{sen } q_2} \underbrace{\begin{bmatrix} l_2 \text{cos}(q_1 + q_2) & l_2 \text{sen}(q_1 + q_2) \\ -l_1 \text{cos } q_1 - l_2 \text{cos}(q_1 + q_2) & -l_1 \text{sen } q_1 - l_2 \text{sen}(q_1 + q_2) \end{bmatrix}}_{\mathbf{J}^{-1}} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \quad (10.b)$$

As relações para a aceleração são conseguidas através da inversão da matriz (5):

$$\begin{bmatrix} \ddot{q}_1 \\ (\ddot{q}_1 + \ddot{q}_2) \end{bmatrix} = \frac{1}{l_1 l_2 \sin q_2} \begin{bmatrix} l_2 \cdot \cos(q_1 + q_2) & l_2 \cdot \sin(q_1 + q_2) \\ -l_1 \cdot \cos q_1 & -l_1 \cdot \sin q_1 \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} + \frac{1}{l_1 l_2 \sin q_2} \begin{bmatrix} l_1 l_2 \cdot \cos q_2 & l_2^2 \\ -l_1^2 & -l_1 l_2 \cdot \cos q_1 \end{bmatrix} \begin{bmatrix} \dot{q}_1^2 \\ (\dot{q}_1 + \dot{q}_2)^2 \end{bmatrix} \quad (11)$$

As expressões (10) e (11) representam a cinemática diferencial inversa, de primeira e de segunda ordem para o robô RR.

### 2.3.4 Cinemática inversa do robô RP

A cinemática inversa, tal como foi referido anteriormente, envolve o cálculo dos ângulos das juntas para uma determinada posição  $(x, y)$  do manipulador. Assim, poderão existir, ou não, soluções. No caso do ponto desejado  $(x, y)$  estar fora da área de trabalho do braço robô, não é possível encontrar uma solução. No caso do ponto  $(x, y)$  estar dentro do espaço de trabalho a solução é encontrada através das expressões seguintes:

$$\begin{bmatrix} r \\ \theta \end{bmatrix} = \begin{bmatrix} \sqrt{x^2 + y^2} \\ \text{arctg}\left(\frac{y}{x}\right) \end{bmatrix} \quad (12)$$

As velocidades nas juntas são obtidas a partir das velocidades cartesianas, através do Jacobiano inverso ( $\mathbf{J}^{-1}$ ) da matriz (7):

$$\begin{bmatrix} \dot{r} \\ \dot{\theta} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \quad (13.a)$$

$$\mathbf{J}^{-1} = \begin{bmatrix} \frac{x}{r} & \frac{y}{r^2} \\ -\frac{y}{r^2} & \frac{x}{r^2} \end{bmatrix} \quad (13.b)$$

As relações para a aceleração são conseguidas através da inversão de (8):

$$\begin{bmatrix} \ddot{r} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{x}{r} & \frac{y}{r} \\ -\frac{y}{r^2} & \frac{x}{r^2} \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} + \begin{bmatrix} \frac{1}{r} & 0 \\ 0 & -\frac{2}{r} \end{bmatrix} \begin{bmatrix} \dot{\theta}^2 \\ \dot{r}\dot{\theta} \end{bmatrix} \quad (14)$$

As expressões (13) e (14) constituem a cinemática diferencial inversa, de primeira e de segunda ordem, para o robô RP.

## 2.4 Planeamento de trajectórias

O planeamento de trajectórias consiste na especificação do caminho que o órgão terminal do robô deve percorrer no espaço e no tempo. Por outras palavras, não são somente os pontos por onde o robô deve passar que devem ser especificados, mas também os instantes em que são alcançados esses pontos. A definição da trajectória pode ser feita quer no espaço cartesiano quer no espaço das juntas. As trajectórias definidas no espaço cartesiano encontram-se mais adaptadas para uma fácil visualização mas, por outro lado, acarretam uma carga computacional mais elevada quando se faz apelo à cinemática inversa [4]. A definição das trajectórias no espaço das juntas coloca uma carga computacional mais reduzida mas, em contrapartida, torna-se difícil antever qual a trajectória executada pelo robô no espaço operacional.

O planeamento de trajectórias pode ser abordado de duas formas distintas:

- através da especificação de um conjunto de pontos pelos quais o órgão terminal do robô deve passar, em determinados instantes, complementadas com restrições de posição, velocidade e aceleração que têm de ser cumpridas.
- através da especificação do tipo de movimento a ser efectuado pelo órgão terminal, nomeadamente, através de trajectórias do tipo rectilíneo.

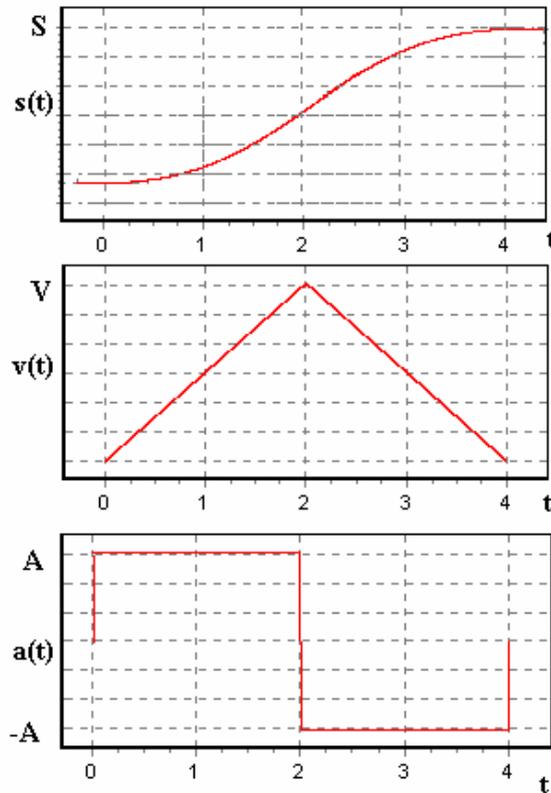
Neste trabalho, as equações da trajectória são definidas através de expressões para a posição  $s(t)$ , a velocidade  $v(t)$  e a aceleração  $a(t)$ , as equações são as seguintes:

$$s(t) = \begin{cases} \frac{1}{2}At^2 & , 0 < \frac{T}{2} \\ A(T \cdot t - 0.5 \cdot t^2 - t_0) & , \frac{T}{2} < t < T \end{cases} \quad (15)$$

$$v(t) = \begin{cases} At & , 0 < \frac{T}{2} \\ A(T - t) & , \frac{T}{2} < t < T \end{cases} \quad (16)$$

$$a(t) = \begin{cases} A & , 0 < \frac{T}{2} \\ -A & , \frac{T}{2} < t < T \end{cases} \quad (17)$$

O diagrama temporal das variáveis  $s(t)$ ,  $v(t)$  e  $a(t)$  são apresentadas na figura 2.12.



**Figura 2.12** – Diagrama temporal da trajectória para aceleração a degrau.

Alternativamente, quando se adopta uma evolução temporal (*i.e.* mais suave) da aceleração sinusoidal nas juntas resultam as seguintes equações:

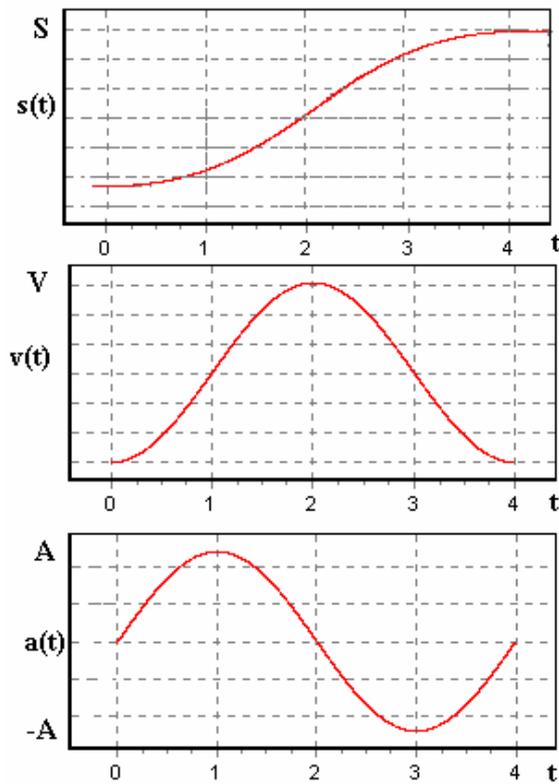
$$\omega_0 = \frac{\pi}{T} \quad (18)$$

$$s(t) = \frac{A}{\omega_0} \left[ t - \frac{1}{\omega} \text{sen}(\omega_0 t) \right] \quad (19)$$

$$v(t) = \frac{A}{\omega_0} \left[ 1 - \frac{\cos(\omega_0 t)}{\omega_0} \right] \quad (20)$$

$$a(t) = A \cdot \text{sen}(\omega_0 t) \quad (22)$$

O diagrama temporal das variáveis  $s(t)$ ,  $v(t)$  e  $a(t)$  são apresentadas na figura 2.13.



**Figura 2.13** – Diagrama temporal da trajectória para aceleração sinusoidal.

### 2.4.1 Planeamento de Trajectórias no Espaço Cartesiano.

Para uma trajectória rectilínea no espaço cartesiano a transformação a fazer é do tipo:

$$\{s(t), v(t), a(t)\} \rightarrow \{x(t), y(t)\} \quad (23)$$

A obtenção desta transformação é feita através das seguintes expressões:

$$x_{\text{inicial}} \neq x_{\text{final}} \begin{cases} \Delta x = x_{\text{inicial}} - x_{\text{final}} \\ \Delta y = y_{\text{inicial}} - y_{\text{final}} \end{cases} \quad (24)$$

$$x_{\text{inicial}} = x_{\text{final}} \begin{cases} \Delta x = 0 \\ \Delta y = y_{\text{inicial}} - y_{\text{final}} \end{cases} \quad (25)$$

$$\alpha = \arctg\left(\frac{\Delta y}{\Delta x}\right) \quad (26)$$

$$x(t) = x_{\text{inicial}} + s(t) \cdot \cos \alpha \quad (27)$$

$$y(t) = y_{\text{inicial}} + s(t) \cdot \sen \alpha \quad (28)$$

$$\frac{dx(t)}{dt} = v(t) \cdot \cos \alpha \quad (29)$$

$$\frac{dy(t)}{dt} = v(t) \cdot \sen \alpha \quad (30)$$

$$\frac{d^2x(t)}{dt^2} = a(t) \cdot \cos \alpha \quad (31)$$

$$\frac{d^2y(t)}{dt^2} = a(t) \cdot \sen \alpha \quad (32)$$

Se pretendermos as posições, velocidades e acelerações nas juntas têm que se calcular através da cinemática inversa.

### 2.4.2 Planeamento de Trajectórias no Espaço das Juntas

Para uma trajectória rectilínea no espaço das juntas a transformação a fazer é do tipo:

$$\{s(t), v(t), a(t)\} \rightarrow \{q_1(t), q_2(t)\} \quad (33)$$

A obtenção desta transformação é feita através das seguintes expressões:

$$q_{1 \text{ inicial}} \neq q_{2 \text{ final}} \begin{cases} \Delta q_1 = q_{1 \text{ inicial}} - q_{1 \text{ final}} \\ \Delta q_2 = q_{2 \text{ inicial}} - q_{2 \text{ final}} \end{cases} \quad (34)$$

$$q_{1 \text{ inicial}} = q_{2 \text{ final}} \begin{cases} \Delta q_1 = 0 \\ \Delta q_2 = q_{2 \text{ inicial}} - q_{2 \text{ final}} \end{cases} \quad (35)$$

$$\alpha = \arctg\left(\frac{\Delta q_2}{\Delta q_1}\right) \quad (36)$$

$$q_1(t) = q_{1 \text{ inicial}} + s(t) \cdot \cos \alpha \quad (37)$$

$$q_2(t) = q_{2 \text{ inicial}} + s(t) \cdot \sen \alpha \quad (38)$$

$$\frac{dq_1(t)}{dt} = v(t) \cdot \cos \alpha \quad (39)$$

$$\frac{dq_2(t)}{dt} = v(t) \cdot \sen \alpha \quad (40)$$

$$\frac{d^2 q_1(t)}{dt^2} = a(t) \cdot \cos \alpha \quad (41)$$

$$\frac{d^2 q_2(t)}{dt^2} = a(t) \cdot \sen \alpha \quad (42)$$

Se pretendermos as posições, velocidades e acelerações no espaço cartesiano têm que se calcular através da cinemática directa, como se viu anteriormente.

## 2.5 Dinâmica do Robô

A dinâmica relaciona as posições, velocidades e acelerações, que ocorrem numa trajectória do robô, com as correspondentes forças/binários requeridos aos actuadores[5].

A dinâmica segue as leis da física clássica e pode ser expressa através de uma relação diferencial:

$$\{q_i(t), \dot{q}_i(t), \ddot{q}_i(t)\} \xrightarrow{\text{Dinâmica Inversa}} \{T(t)\} \quad (43)$$

$$\{T(t), q_i(t=0), \dot{q}_i(t=0)\} \xrightarrow{\text{Dinâmica Directa}} \{\ddot{q}_i(t), \dot{q}_i(t), q_i(t)\} \quad (44)$$

### 2.5.1 Dinâmica Inversa

Na forma simbólica para um robô com  $n$  eixos, a dinâmica inversa pode ser expressa por um conjunto de equações diferenciais não lineares da forma:

$$\mathbf{T} = \mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) \quad (45)$$

onde  $\mathbf{T}$  representa o vector  $n \times 1$  dos binários dos actuadores,  $\mathbf{J}(\mathbf{q})$  é a matriz simétrica das inércias, de dimensão  $n \times n$ ,  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$  é o vector,  $n \times 1$  dimensional, dos binários/forças coriolis/centrípetos e  $\mathbf{G}(\mathbf{q})$  é o vector,  $n \times 1$  dimensional, dos binários/forças gravitacionais.

Existem vários métodos para a obtenção das equações da dinâmica inversa:

- método de Newton-Euler que se obtém extraindo relações vectoriais de equilíbrio de forças/binários nos diversos elos.
- método de Lagrange que se obtém a partir das energias cinética e potencial.

O primeiro método conduz a expressões mais eficientes sob o ponto de vista computacional. O segundo método possibilita uma obtenção mais simples e sistemática.

Desta forma, o estudo dinâmico de um robô manipulador pode-se obter facilmente a partir do seguinte lagrangeano:

$$\mathbf{L} = \mathbf{K} - \mathbf{P} \quad (46)$$

$$\mathbf{T} = \frac{d}{dt} \left( \frac{d\mathbf{L}}{d\dot{\mathbf{q}}} \right) - \frac{d\mathbf{L}}{d\mathbf{q}} \quad (47)$$

onde  $\mathbf{K}$  é a energia cinética,  $\mathbf{P}$  é a energia potencial,  $\mathbf{T}$  é o binário /força generalizado,  $\mathbf{q}$  é a coordenada generalizada e  $\mathbf{t}$  é o valor da variável tempo.

### 2.5.2 Dinâmica Directa

A dinâmica directa corresponde à integração da expressão (45), isto é:

$$\ddot{\mathbf{q}}(\mathbf{t}) = \mathbf{J}(\mathbf{q})^{-1} [\mathbf{T} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{G}(\mathbf{q})] \quad (48)$$

$$\dot{q}(t) = \int_0^t \ddot{q} dt + \dot{q}(0) \quad (49)$$

$$q(t) = \int_0^t \dot{q} dt + q(0) \quad (50)$$

Normalmente adopta-se um método numérico de integração que seja computacionalmente eficiente. Neste sentido é de referir o algoritmo de Runge-Kutta [6].

### 2.5.3 Dinâmica Inversa do Robô RR

Para o caso do robô RR obtêm-se as seguintes expressões para a dinâmica inversa:

$$J(q) = \begin{bmatrix} (m_1 + m_2)r_1^2 + m_2r_2^2 + & m_2r_2^2 + \\ 2m_2r_1r_2C_2 + J_{1m} + J_{1g} & m_2r_1r_2C_2 \\ m_2r_2^2 + m_2r_1r_2C_2 & m_2r_2^2 + \\ & J_{2m} + J_{2g} \end{bmatrix} \quad (51)$$

$$C(q, \dot{q}) = \begin{bmatrix} -m_2r_1r_2S_2\dot{q}_2^2 - 2m_2r_1r_2S_2\dot{q}_1\dot{q}_2 \\ m_2r_1r_2S_2\dot{q}_1^2 \end{bmatrix} \quad (52)$$

$$G(q) = \begin{bmatrix} g(m_1r_1C_1 + m_2r_1C_1 + m_2r_2C_{12}) \\ gm_2r_2C_{12} \end{bmatrix} \quad (53)$$

onde  $C_1 = \cos(q_1)$ ,  $C_2 = \cos(q_2)$ ,  $S_{12} = \sin(q_1 + q_2)$  e  $C_{12} = \cos(q_1 + q_2)$ .

### 2.5.4 Dinâmica Inversa do Robô RP

No caso do robô RP obtêm-se as seguintes expressões para a dinâmica inversa:

$$J(q) = \begin{bmatrix} m_1r^2 + m_2q_2^2 + & 0 \\ J_{1m} + J_{1g} & \\ 0 & m_2 + \\ & J_{2m} + J_{2g} \end{bmatrix} \quad (54)$$

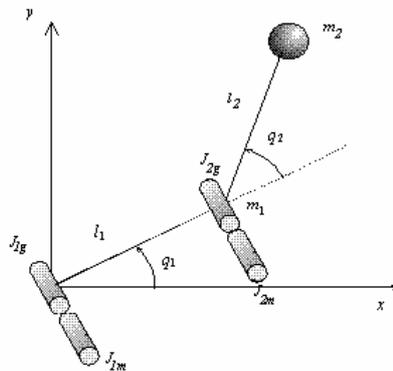
$$C(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} 2m_2q_2\dot{q}_1\dot{q}_2 \\ m_2q_2\dot{q}_1 \end{bmatrix} \quad (55)$$

$$G(\mathbf{q}) = \begin{bmatrix} gC_1(m_2r + m_2q_2) \\ gm_2S_1 \end{bmatrix} \quad (56)$$

onde  $C_1 = \cos(q_1)$ ,  $C_2 = \cos(q_2)$ ,  $S_{12} = \sin(q_1 + q_2)$  e  $C_{12} = \cos(q_1 + q_2)$ .

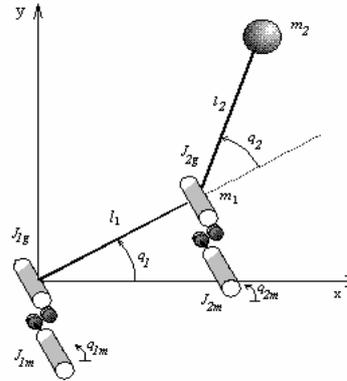
## 2.6 Fenómenos Dinâmicos nas Juntas dos Robôs

Nesta secção aborda-se a existência de efeitos dinâmicos nos eixos e no ambiente de trabalho considerando a modelização do robô não apenas como sendo ideal (figura 2.14) mas também, consideram-se a transmissão nas juntas contendo folgas (figura 2.15), flexibilidade (figura 2.16) e com atrito não linear (figura 2.17). Além disso, estudam-se também os casos de saturação nos actuadores dos eixos e a quantificação dos sensores de posição.



**Figura 2.14** – Robô com juntas ideais.

### 2.6.1 Juntas dos Robôs com Folgas



**Figura 2.15** - Robô com folgas nas engrenagens.

$$\mathbf{T} = \mathbf{J}_m \ddot{\mathbf{q}}_m + \mathbf{B}_m \dot{\mathbf{q}}_m \quad (57)$$

onde os parâmetros  $J_{ig}$ ,  $J_{im}$ ,  $q_{im}$  respectivamente a inércia após a engrenagem, inércia do motor e a posição do motor.

Frequentemente, num robô existem engrenagens sujeitas à ocorrência de folgas, isto é, à existência de intervalos nos quais a transmissão de movimento entre dois componentes consecutivos é interrompida (figura 2.14). No entanto, devido aos princípios da conservação de energia e da quantidade de movimento, pode-se determinar o efeito das folgas e avaliar as suas consequências.

O estudo das folgas em engrenagens de manipuladores é de grande importância, pois estas afectam a exactidão do controlo (seja de posição, seja de força), e levam à ocorrência de instabilidades.

Desta forma, podem-se modelizar estes fenómenos da seguinte forma.

Quando ocorre um impacto entre as inércias, pelo princípio da conservação do momento resulta:

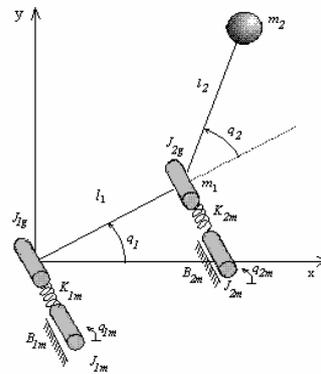
$$\dot{q}'_i = \frac{\dot{q}_i (J_{ii} - J_{im}) + \dot{q}_{im} J_{im} (1 + \epsilon)}{J_{ii} + J_{im}} \quad (58)$$

$$\dot{q}'_{im} = \frac{\dot{q}_i J_i (1 + \epsilon) + \dot{q}_{im} (J_{im} - \epsilon J_{ii})}{J_{ii} + J_{im}} \quad (59)$$

onde  $0 < \varepsilon < 1$  (constante de *Newton*) define a elasticidade do impacto tal que ( $\varepsilon = 0$  corresponde a impacto não elástico,  $\varepsilon = 1$  a impacto elástico) e  $\dot{q}'_i$  e  $\dot{q}'_{im}$  são as velocidades das inércias das juntas e do motor depois da colisão.

### 2.6.2 Juntas dos Robôs com Flexibilidade

Nos robôs industriais o accionamento não é directo (devido à existência de mecanismos de transmissão) pelo que se torna importante estudar os efeitos das flexibilidades nas juntas.



**Figura 2.16** – Robô com flexibilidade nas juntas.

Este fenómeno é usualmente modelizado por uma mola que interliga o eixo do motor com o eixo da junta do robô, resultando as expressões:

$$T = J_m \ddot{q}_m + B_m \dot{q}_m + K_m (q_m - q) \quad (60)$$

$$K_m (q_m - q) = J(q) \ddot{q} + C(q, \dot{q}) + G(q) \quad (61)$$

onde os parâmetros  $J_m$ ,  $B_m$ ,  $K_m$  são a inércia, o factor de amortecimento e a rigidez do motor e da transmissão, respectivamente[7][8][9].

### 2.6.3 Juntas dos Robôs com Atrito

A modelização dos atritos revela algumas dificuldades devido ao facto de estes serem dependentes de muitos factores, nomeadamente dos materiais em contacto, da temperatura, da humidade do ambiente de trabalho e, por outro lado, de variarem com o tipo de sistema, com o desgaste, etc.

A fim de evitar uma grande complexidade, têm-se adoptado modelos simples nomeadamente com a formulação de uma força em função da velocidade relativa entre as duas superfícies em contacto.

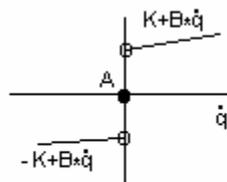


Figura 2.17 – Modelização do atrito.

Assim, o binário de atrito é calculado da seguinte forma:

$$T_{\text{atrito}} = \begin{cases} K_i + B_i \dot{q}_i & , \quad q_i > 0 \\ A_i & , \quad q_i = 0 \\ -K_i + B_i \dot{q}_i & , \quad q_i < 0 \end{cases} \quad (62)$$

onde os parâmetros,  $B_i$ ,  $K_i$  são o atrito viscoso, o atrito de Coulomb e  $A_i$  constante de atrito no instante inicial.

### 2.6.4 Juntas dos Robôs com Limitações dos Binários

A modelização da saturação nos actuadores das juntas é também estudada pois, frequentemente, os controladores requerem esforços (*i.e.* binários dos actuadores) impraticáveis.

Os actuadores dos robôs industriais apresentam limitações pelo que é também importante verificar o comportamento do sistema na ocorrência destas situações.

O modelo correspondente vem:

$$\begin{cases} \mathbf{T} > \mathbf{T}_{\max} & , \mathbf{T} = \mathbf{T}_{\max} \\ \mathbf{T}_{\min} < \mathbf{T} < \mathbf{T}_{\max} & , \mathbf{T} = \mathbf{T} \\ \mathbf{T} < \mathbf{T}_{\min} & , \mathbf{T} = \mathbf{T}_{\min} \end{cases} \quad (63)$$

onde  $\mathbf{T}_{\max}, \mathbf{T}_{\min}$ , representa o binário máximo e mínimo do actuador.

### 2.6.5 Juntas dos Robôs com Limitações dos Sensores

Tendo em vista que a resolução dos sensores depende da sua constituição física é importante quantificar as saídas de modo a modelizar a totalidade do sistema. A quantificação dos sensores acoplados nas juntas vai criar certos efeitos na medição e na estimação das variáveis e, dessa forma, vai alterar a exactidão do controlo. Assim pode modelizar-se a quantificação dos sensores das juntas através do nível de quantificação desejado (*i.e.* número de bits a truncar).

## 2.7 Análise Dinâmica e Controlo de Robôs

O objectivo de um algoritmo de controlo é assegurar que uma sequência de movimentos planeados seja executada correctamente, face aos erros imprevisíveis resultantes das limitações da precisão computacional e de efeitos mecânicos, tais como, o atrito ou as folgas.

A formulação básica de um algoritmo de controlo consiste na medição da posição do braço manipulador durante o movimento, na comparação com a posição desejada, e conseqüentemente, no desenvolvimento de uma actuação com vista a eliminar o eventual erro. No controlo de movimento de robôs há a considerar dois aspectos: os modelos dinâmicos do manipulador e as leis de controlo.

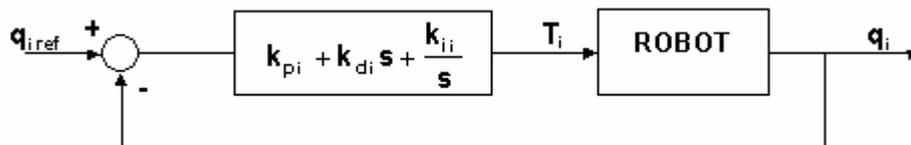
O primeiro aspecto passa pelo estudo da dinâmica do braço do robô, a qual é importante para a simulação dos movimentos. Assim, testam-se diferentes estratégias de controlo sem que isso implique custos e eventuais problemas mecânicos que podem ocorrer com robôs reais. É ainda de salientar que o estudo da dinâmica é também importante para a análise do manipulador na sua concepção mecânica.

O segundo aspecto envolve o estudo do algoritmo de controlo. Geralmente, o movimento de um braço robô é realizado em duas fases distintas. A primeira fase, em que o braço se move de uma posição inicial para a proximidade do ponto desejado é possível um controlo menos apurado. A segunda fase na qual o braço interage dinamicamente com o objecto, utilizando a informação dos sensores para completar a tarefa, o que requer um controlo “fino” do movimento.

Nesta perspectiva, o desenvolvimento de algoritmos visa o controlo das variáveis posição, velocidade e força/binário de acordo com as tarefas a executar. Desta forma, são apresentadas várias alternativas de controlo, nomeadamente: controladores lineares *versus* não-lineares, bem como controladores baseados em modelos. Uma outra forma de classificar os algoritmos tem a ver com o espaço onde é feito o controlo, ou seja no espaço cartesiano *versus* no espaço das juntas.

### 2.7.1 Controladores PID

Pode ver-se a estrutura básica deste tipo de controladores lineares no diagrama de blocos da figura 2.18. O algoritmo é idêntico para cada junta do manipulador.



**Figura 2.18** – Controladores lineares do tipo PID.

O controlo proporcional (P) consiste numa constante multiplicativa  $K_p$  que actua sobre o erro. Por outras palavras, a acção de comando gerado pelo controlador é proporcional à diferença entre o valor de referência desejado e o valor da saída.

A introdução do termo integrador (I) num controlador tem por finalidade de eliminar o erro em regime permanente para valores de referência constante mas, em contrapartida, aumenta o tempo de estabelecimento e diminui a estabilidade.

Um termo derivativo (D) é responsável pela diminuição da sobre-elevação (*overshoot*) e do tempo de estabelecimento, melhorando a estabilidade [10].

Assim as equações do sistema são:

$$\mathbf{T} = \mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) \quad (64)$$

$$T_i = (K_{i_p} + \frac{K_{i_i}}{s} + K_{i_d} \cdot s) \cdot e(t) \quad (65)$$

$$e_i(t) = q_{i_{ref}} - q_i \quad (66)$$

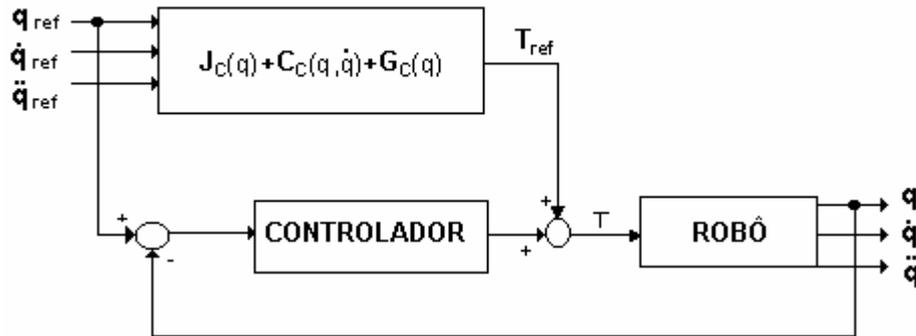
Onde T é o vector dos binários a aplicar nos actuadores e  $e(t)$  é a diferença entre a posição de referência e a posição actual.

### 2.7.2 Controladores baseados em modelos

Como a dinâmica dos manipuladores é caracterizada por fortes não-linearidades, torna-se difícil fazer um controlo eficiente através de algoritmos lineares do tipo PID. Deste modo, os métodos do controlo clássico tornam-se inadequados na presença de fortes não-linearidades. Assim, criaram-se algoritmos de compensação nomeadamente controladores baseados em modelos do sistema. Nesta perspectiva, são de referir o controlador de compensação por acção “antecipativa” (*Feedforward*) e o controlador por binário calculado (*Computed Torque*)[5][11][12].

O controlador com compensação “Feedforward” prevê as saídas desejadas para os actuadores através do modelo da dinâmica do manipulador, baseado nas posições, velocidades e acelerações desejadas. Deste modo, pode-se calcular, antecipadamente, os binários requeridos aos actuadores. Além disso, este cálculo pode ser feito “*off-line*”, ou seja, pode ser executado na fase de planeamento da trajectória desejada.

Obviamente, a precisão do modelo da dinâmica do manipulador influencia o desempenho do algoritmo de controlo.



**Figura 2.19** - O controlador do tipo compensação por acção “antecipativa”.

A figura 2.19 ilustra os principais blocos do sistema. A variável  $T_{ref}$  é o binário calculado para um sistema ideal e uma frequência do controlador elevada. Se os binários inerciais, coriolis/centrípetos e gravitacionais do robô forem iguais aos do modelo então tem-se uma modelização perfeita e a saída  $q(t)$  acompanha a referência  $q_{ref}(t)$ .

O controlador que está inserido no anel de controlo serve para compensar qualquer desvio, no caso de a modelização não ser perfeita, criando um binário que somado ao binário de referência (calculado a partir do modelo do manipulador) gera o binário requerido aos actuadores. Por outro lado, o controlador existente nestes algoritmos pode ser do tipo linear bem como do tipo não linear.

As equações do sistema são:

$$T = J(q)\ddot{q} + C(q, \dot{q}) + G(q) \quad (67)$$

$$T_{ref} = J(q_{ref})\ddot{q}_{ref} + C(q_{ref}, \dot{q}_{ref}) + G(q_{ref}) \quad (68)$$

$$T = T_{ref} + T_{controlador} \quad (69)$$

O controlo pelo método binário calculado (figura 2.20) é uma forma de cancelamento de não linearidades, pois se a dinâmica do modelo for idêntica à do robô então as perturbações não lineares provocadas pela dinâmica, são canceladas.

Em princípio, este tipo de controlo é mais eficiente que o controlador “feedforward”, pois a compensação, é feita adoptando os valores reais das variáveis e não os valores de referência. Assim, o controlador do binário calculado processa a dinâmica em tempo real, utilizando os valores da posição e da velocidade das juntas. Se

o modelo for exacto, então cada elo do robô pode ser desacoplado e o erro da trajectória tende para zero, de acordo com uma equação diferencial linear de segunda ordem.

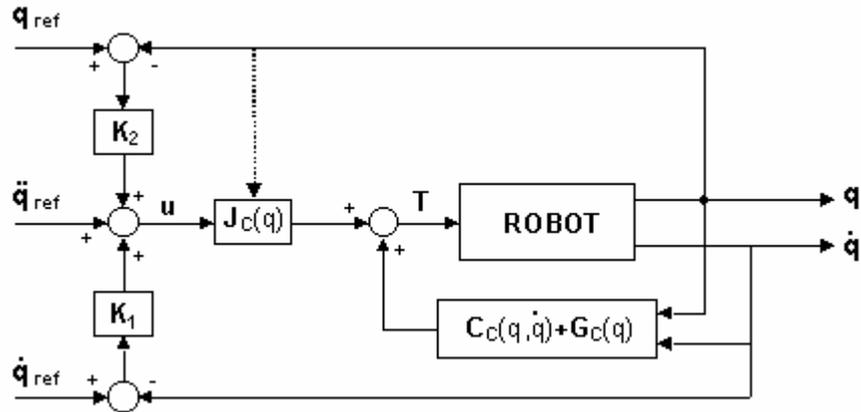


Figura 2.20 - O controlo do binário calculado.

As equações do sistema são:

$$T = J(q)\ddot{q} + C(q, \dot{q}) + G(q) \quad (70)$$

$$T = J_c(q)u(t) + C_c(q, \dot{q}) + G_c(q) \quad (71)$$

$$u = \ddot{q}_{ref} + K_1(\dot{q}_{ref} - \dot{q}) + K_2(q_{ref} - q) \quad (72)$$

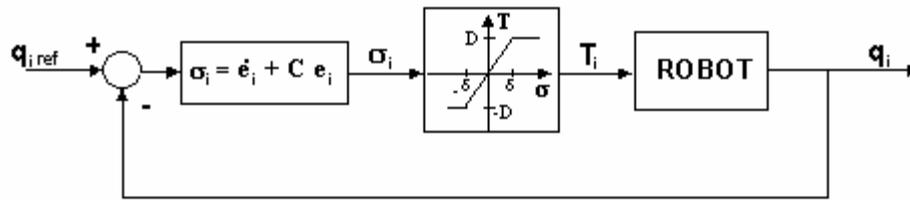
$$K_1 = \begin{bmatrix} 2 \cdot \xi \omega_{n_1} & 0 \\ 0 & 2 \cdot \xi \omega_{n_2} \end{bmatrix} \quad (73)$$

$$K_2 = \begin{bmatrix} \omega_{n_1}^2 & 0 \\ 0 & \omega_{n_2}^2 \end{bmatrix} \quad (74)$$

onde  $K_1$  representa o ganho de velocidade e o  $K_2$  representa um ganho de posição e  $u$  representa o erro.

### 2.7.3 Controladores Não Lineares

Os controladores de estrutura variável CEVs (*Variable Structure Controllers*) são algoritmos não lineares conforme representado no diagrama de blocos da figura 2.21.



**Figura 2.21** – Controlador de estrutura variável.

Sistemas de controlo com estrutura variável demonstram uma elevada robustez, e uma baixa carga computacional e não requerem o conhecimento do modelo do sistema a controlar [13][14][15].

Os elementos principais de um CEV são a superfície de deslizamento  $\sigma$  (*sliding surface*) e a acção de controlo (*control effort*).

Para um CEV que adopte um modelo de primeira ordem (MPO) na superfície de deslizamento, existem duas possibilidades para realizar a correspondente equação diferencial, que são a utilização do sinal de erro e de sua derivada (*i.e.* proporcional e derivativo, MPO-PD ) ou a utilização do sinal do erro e do seu integral (*i.e.* proporcional e integral, MPO-PI).

Assim, para o caso do algoritmo CEV-MPO-PD resulta:

$$\sigma = \dot{q}_e + c \cdot q_e \quad (75)$$

Para o caso do algoritmo CEV-MPO-PI resulta:

$$\sigma = q_e + c \cdot \int q_e \cdot dt \quad (76)$$

onde o parâmetro  $c$  é o valor próprio,  $q_e$  é a entrada do CEV e  $\sigma$  é a curva de deslizamento.

Alternativamente pode adoptar-se um CEV com um modelo de segunda ordem (MSO). Neste caso existem, também, várias possibilidades para estabelecer a correspondente equação diferencial, nomeadamente, utilizar o sinal do erro e as suas primeira e segunda derivadas (MSO-PDD<sup>2</sup>) ou recorrer ao erro, ao seu integral e à sua derivada (MSO-PID).

Assim, para o algoritmo CEV-MSO-PDD<sup>2</sup> resulta:

$$\sigma = \ddot{q}_e + 2\xi_i \omega_{ni} \dot{q}_e + \omega_{ni}^2 q_e \quad (77)$$

Para o algoritmo CEV-MSO-PID resulta:

$$\sigma = \dot{q}_e + 2\xi_i \omega_{ni} q_e + \omega_{ni}^2 \int q_e dt \quad (78)$$

onde o  $\xi$  é coeficiente de amortecimento e  $\omega_n$  é a frequência natural não amortecida.

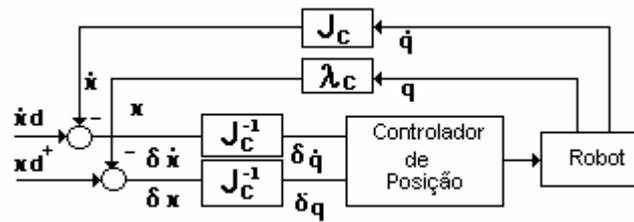
O segundo bloco é constituído pela acção de controlo. Uma expressão usada frequentemente por este bloco é dada por:

$$T_i = \begin{cases} T_{\max} & , T \geq \frac{T_{\max}}{K_i} \\ K_i \cdot T & , |T| < \frac{T_{\max}}{K_i} \\ -T_{\max} & , T \leq -\frac{T_{\max}}{K_i} \end{cases} \quad (79)$$

Neste caso, evita-se uma acção “intempestiva” do tipo relé, já que existe uma banda “suave” com um ganho proporcional  $K_i$ .

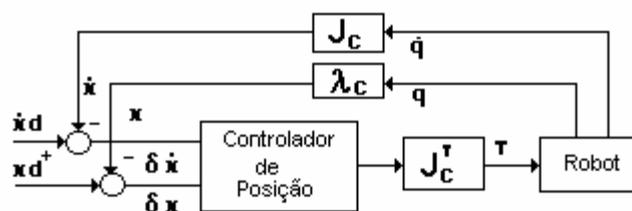
### 2.7.4 Controlo Baseado no Espaço Cartesiano

Todos os controladores descritos anteriormente apresentam um algoritmo concebido no espaço das juntas do manipulador. Uma outra alternativa consiste em criar um algoritmo de controlo onde o processo de conversão da trajectória de referência é substituído por uma conversão de coordenadas dentro da malha de realimentação do sistema. Deste modo, pode-se comparar a trajectória desejada com a trajectória real do manipulador no espaço cartesiano.



**Figura 2.22** - Controlador do inverso do Jacobiano.

A alteração do algoritmo de controlo acarreta cálculos suplementares na malha de realimentação, devido às transformações cinemáticas. Assim, a mudança da estratégia de controlo do espaço das juntas para o espaço cartesiano pode ter vantagens e inconvenientes, visto que, para uma estratégia baseada no controlo do espaço cartesiano, os algoritmos são implementados com uma frequência de amostragem mais baixa do que os sistemas de controlo baseado nas juntas. De facto, em geral, a diminuição da frequência de amostragem, acarreta uma degradação da estabilidade do sistema. Para um algoritmo de controlo no espaço cartesiano da comparação da posição nos eixos cartesianos obtêm-se o erro  $\delta x$  e a sua derivada  $\delta \dot{x}$ . Estes erros, que se presumem serem pequenos, podem ser mapeados em pequenos incrementos no espaço das juntas respectivamente  $\delta q_i$ ,  $\delta \dot{q}_i$  ( figura 2.22) através do Jacobiano inverso. Os erros no espaço das juntas são multiplicados por ganhos, de modo a obter os binários necessários à correcção da trajectória. Este tipo de controlador é designado por “controlador do inverso do Jacobiano”.



**Figura 2.23** - Controlador do Jacobiano transposto.

O diagrama da figura 2.23 apresenta outra alternativa de controlo no espaço cartesiano. O vector dos erros no espaço cartesiano  $\delta x$ ,  $\delta \dot{x}$  dá origem a um vector de força  $F_i$  que, por seu lado, é mapeado pelo Jacobiano transposto, de modo a obterem-se

os binários  $T_i$  nas juntas para a correcção da trajectória. Este algoritmo designa-se por “controlador do Jacobiano transposto”.

Note-se que todos os controladores de posição/velocidade apresentados podem adoptar um algoritmo quer do tipo linear quer do tipo não linear.

Para o robô RR as equações do sistema são:

$$\lambda_c = \begin{bmatrix} l_1 C_1 + l_2 C_{12} \\ l_1 S_1 + l_2 S_{12} \end{bmatrix} \quad (80)$$

$$\mathbf{J}_c^T = \begin{bmatrix} -l_1 S_1 - l_2 S_{12} & -l_2 S_{12} \\ l_1 C_1 + l_2 C_{12} & l_2 C_{12} \end{bmatrix} \quad (81)$$

$$\mathbf{J}_c^T = \begin{bmatrix} -l_1 S_1 - l_2 S_{12} & l_1 C_1 + l_2 C_{12} \\ -l_2 S_{12} & l_2 C_{12} \end{bmatrix} \quad (82)$$

$$\mathbf{J}_c^{-1} = \frac{1}{l_1 l_2 S_2} \begin{bmatrix} l_2 C_{12} & l_2 S_{12} \\ -l_1 C_1 - l_2 C_{12} & -l_1 S_1 - l_2 S_{12} \end{bmatrix} \quad (83)$$

As equações do robô RP são:

$$\lambda_c = \begin{bmatrix} r C_1 \\ r S_1 \end{bmatrix} \quad (84)$$

$$\mathbf{J}_c = \begin{bmatrix} C_1 & -r S_1 \\ S_1 & r C_1 \end{bmatrix} \quad (85)$$

$$\mathbf{J}_c^T = \begin{bmatrix} C_1 & S_1 \\ -r S_1 & r C_1 \end{bmatrix} \quad (86)$$

$$\mathbf{J}_c^{-1} = \begin{bmatrix} \frac{x}{r} & \frac{y}{r^2} \\ -\frac{y}{r^2} & \frac{x}{r^2} \end{bmatrix} \quad (87)$$

onde  $C_1 = \cos(q_1)$ ,  $C_2 = \cos(q_2)$ ,  $S_{12} = \sin(q_1 + q_2)$  e  $C_{12} = \cos(q_1 + q_2)$

Tal como nos controladores que adoptam o espaço das juntas, estes algoritmos podem utilizar técnicas de controlo baseadas na linearização do sistema, de modo a eliminar todas as não linearidades impostas pela dinâmica, resultando controladores baseados em modelos. Assim, podem escrever-se as equações dinâmicas do movimento do manipulador em termos das variáveis cartesianas:

$$\mathbf{F} = \mathbf{M}_x(\mathbf{q})\ddot{\mathbf{x}} + \mathbf{V}_x(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}_x(\mathbf{q}) \quad (88)$$

onde  $\mathbf{F}$  é o vector  $n \times 1$  de força nos actuadores que actuam no robô,  $\mathbf{x}$  é o vector  $n \times 1$  que representa a posição e orientação do manipulador no espaço cartesiano,  $\mathbf{M}_x$ , é a matriz das inércias de dimensão  $n \times n$ ,  $\mathbf{V}_x$  é o vector  $n \times 1$  dimensional, dos binários/forças coriolis/centrípetos e  $\mathbf{G}_x$  é o vector  $n \times 1$  dimensional, dos binários/forças gravitacionais.

$$\mathbf{M}_x(\mathbf{q}) = \mathbf{J}_c^{-T}(\mathbf{q})\mathbf{J}(\mathbf{q})\mathbf{J}_c^{-1}(\mathbf{q}) \quad (89)$$

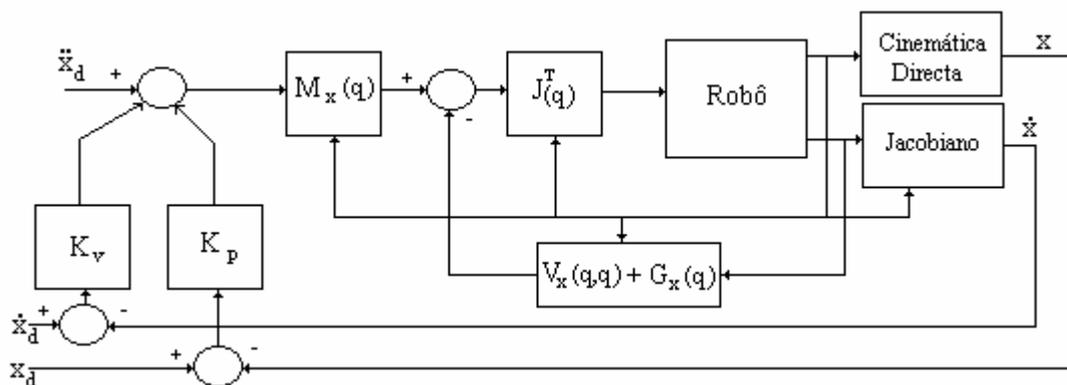
$$\mathbf{V}_x(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{J}_c^{-T}(\mathbf{q})\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{J}_c^{-1}(\mathbf{q}) - \mathbf{M}_x(\mathbf{q})\dot{\mathbf{J}}_c(\mathbf{q})\mathbf{J}_c^{-1}(\mathbf{q}) \quad (90)$$

$$\mathbf{G}_x(\mathbf{q}) = \mathbf{J}_c^{-T}(\mathbf{q})\mathbf{G}(\mathbf{q}) \quad (91)$$

$$\boldsymbol{\gamma} = \mathbf{J}^T(\mathbf{q}) \cdot \mathbf{F} \quad (92)$$

onde  $\mathbf{J}_c^{-T}$  é o vector  $n \times n$  do jacobiano transposto inverso,  $\mathbf{J}(\mathbf{q})$  é a matriz simétrica inercial de dimensão  $n \times n$ ,  $\mathbf{J}_c^{-1}(\mathbf{q})$  é o vector  $n \times n$  do jacobiano inverso,  $\boldsymbol{\gamma}$  é o vector  $n \times 1$  dos binários dos actuadores.

Na figura 2.24 mostra-se um dos possíveis diagramas de controlo utilizando o modelo do manipulador

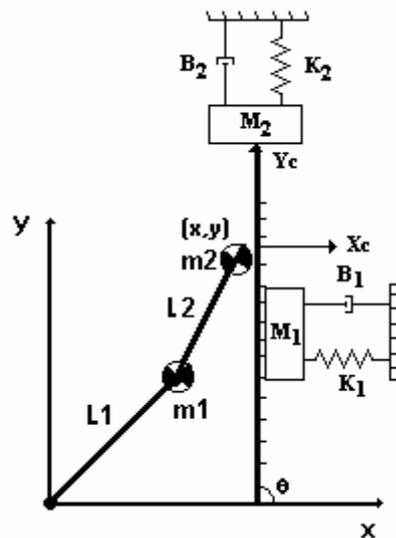


**Figura 2.24** – Controlador baseado no modelo do robô.

## 2.8 Controlo Híbrido Posição/Força de Robôs

Este tipo de controladores adopta um algoritmo que envolve um controlo de posição, segundo certas direcções do espaço de trabalho; e um controlo da força de contacto (entre o órgão terminal do robô e a superfície de restrição) nas restantes direcções.

Como exemplos de aplicação deste algoritmo de controlo pode referir-se a rebarbagem, a soldadura por pontos, a moagem, o seguimento de contornos de peças em processos de fabrico e outros.

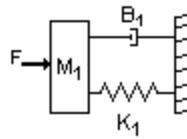


**Figura 2.25** - Robô em interação com o ambiente com inclinação 90°.

Na figura 2.25 apresenta-se o modelo dinâmico do robô e da superfície de restrição do ambiente com um ângulo  $\theta = 90^\circ$ , esta é modelizada, (no contacto), por um sistema: mola - massa - amortecedor. O contacto com superfície de restrição é modelizado como um sistema dinâmico linear através de três parâmetros de acordo com a equação diferencial:

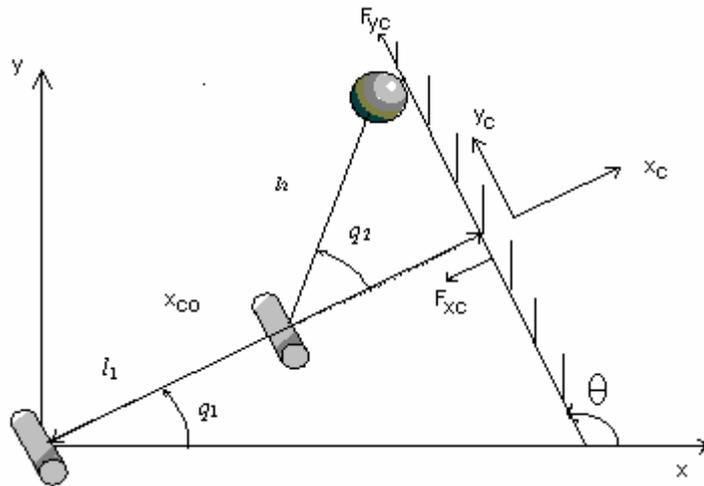
$$F = M\ddot{x} + B\dot{x} + Kx \quad (93)$$

onde  $F$  é a força de contacto e  $M$ ,  $B$  e  $K$  são a massa, amortecimento e rigidez, respectivamente.



**Figura 2.26** – Modelo do ambiente.

O sistema de coordenadas cartesiano  $\{x_c, y_c\}$  espaço associado ao contacto na mão do robô utilizado no espaço das tarefas está representado na figura 2.27.



**Figura 2.27** – Sistema de coordenadas do robô em interação com o ambiente.

Nesta ordem de ideias, o sistema total composto pelo robô e pelo ambiente é descrito pelas equações:

$$F_{x_c} = M \cdot \ddot{x}_c + B \cdot \dot{x}_c + K \cdot (x_c - x_{co}) \quad (94)$$

$$x_{co} = r_1 \text{sen}(\theta - q_{10}) + r_2 \text{sen}(\theta - q_{10} - q_{20}) \quad (95)$$

$$x_c = r_1 \text{sen}(\theta - q_1) + r_2 \text{sen}(\theta - q_1 - q_2) \quad (96)$$

$$\dot{x}_c = -r_1 \dot{q}_1 \cos(\theta - q_1) - r_2 (\dot{q}_1 + \dot{q}_2) \cos(\theta - q_1 - q_2) \quad (97)$$

$$\ddot{x}_c = -r_1 \ddot{q}_1 \cos(\theta - q_1) - r_2 (\ddot{q}_1 + \ddot{q}_2) \cos(\theta - q_1 - q_2) - r_1 \dot{q}_1^2 \text{sen}(\theta - q_1) - r_2 (\dot{q}_1 + \dot{q}_2)^2 \text{sen}(\theta - q_1 - q_2) \quad (98)$$

$$F_{y_c} = M \cdot \ddot{y}_c + B \cdot \dot{y}_c + K \cdot (y_c - y_{co}) \quad (99)$$

$$y_{co} = r_1 \cos(\theta - q_{10}) + r_2 \cos(\theta - q_{10} - q_{20}) \quad (100)$$

$$y_c = r_1 \cos(\theta - q_1) + r_2 \cos(\theta - q_1 - q_2) \quad (101)$$

$$\dot{y}_c = r_1 \dot{q}_1 \sin(\theta - q_1) + r_2 (\dot{q}_1 + \dot{q}_2) \sin(\theta - q_1 - q_2) \quad (102)$$

$$\begin{aligned} \ddot{y}_c = & r_1 \ddot{q}_1 \sin(\theta - q_1) + r_2 (\ddot{q}_1 + \ddot{q}_2) \sin(\theta - q_1 - q_2) \\ & - r_1 \dot{q}_1^2 \cos(\theta - q_1) - r_2 (\dot{q}_1 + \dot{q}_2)^2 \cos(\theta - q_1 - q_2) \end{aligned} \quad (103)$$

$$F_y = -F_{x_c} \sin(\theta) + F_{y_c} \cos(\theta) \quad (104)$$

$$F_x = F_{x_c} \cos(\theta) - F_{y_c} \sin(\theta) \quad (105)$$

A variável  $q_{i0}$  representa o valor de  $q_i$  ( $i=1,2$ ) para  $t = 0$  s supondo que o sistema está em repouso (isto é, que o robô está em contacto com a superfície de restrição e que  $F = 0$ ). A figura 2.28 apresenta o diagrama de blocos do algoritmo de CH.

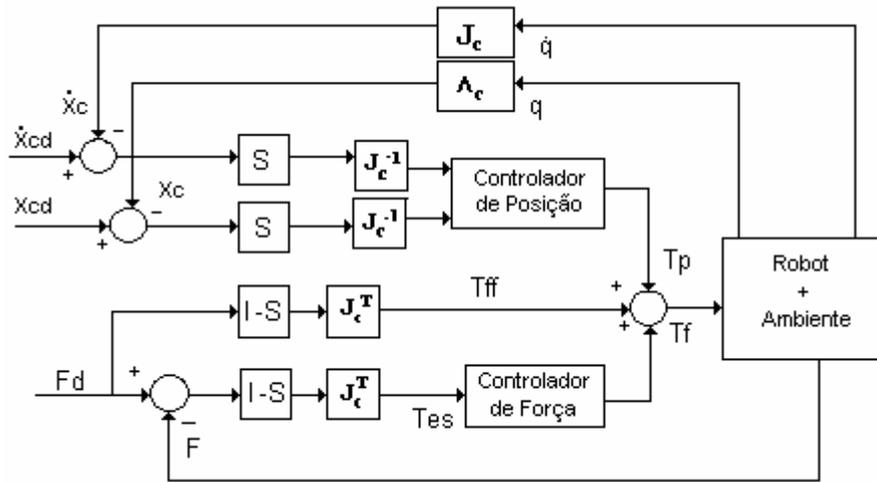


Figura 2.28 –Controlador híbrido.

Para os subcontroladores de posição e de força podem ser adoptados controladores lineares bem como controladores não lineares. Para um controlo da força na direcção  $x_c$  e de posição na direcção de  $y_c$ , resultam as seguintes expressões:

$$s = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \quad (106)$$

$$J_c^{-1}(q) = \frac{1}{j_{c11}j_{c22} - j_{c12}j_{c21}} \begin{bmatrix} j_{c22} & -j_{c12} \\ -j_{c21} & j_{c11} \end{bmatrix} \quad (107)$$

$$J_c(q) = \begin{bmatrix} -r_1 \cos(\theta - q_1) - r_2 \cos(\theta - q_1 - q_2) & -r_2 \cos(\theta - q_1 - q_2) \\ r_1 \sin(\theta - q_1) + r_2 \sin(\theta - q_1 - q_2) & r_2 \sin(\theta - q_1 - q_2) \end{bmatrix} \quad (108)$$

$$\det[J_c(q)] = r_1 r_2 \sin(q_2) \quad (109)$$

$$\lambda_c = \begin{cases} x_c = r_1 \sin(\theta - q_1) + r_2 \sin(\theta - q_1 - q_2) \\ y_c = r_1 \cos(\theta - q_1) + r_2 \cos(\theta - q_1 - q_2) \end{cases} \quad (110)$$

$$\mathbf{J}_c^T(\mathbf{q})(\mathbf{I} - \mathbf{S}) = \begin{bmatrix} -r_1 \cos(\theta - q_1) - r_2 \cos(\theta - q_1 - q_2) & 0 \\ -r_2 \cos(\theta - q_1 - q_2) & 0 \end{bmatrix} \quad (111)$$

$$\mathbf{J}_c^{-1}(\mathbf{q})(\mathbf{S}) = \frac{1}{r_1 r_2 \sin(q_2)} \begin{bmatrix} 0 & -r_2 \cos(\theta - q_1 - q_2) \\ 0 & -r_1 \cos(\theta - q_1) - r_2 \cos(\theta - q_1 - q_2) \end{bmatrix} \quad (112)$$

Alternativamente, para um controlo da força no plano  $\mathbf{y}_c$  e a posição na direcção de  $\mathbf{x}_c$ , resultam as seguintes expressões:

$$\mathbf{s} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad (113)$$

$$\mathbf{J}_c^T(\mathbf{q})(\mathbf{I} - \mathbf{S}) = \begin{bmatrix} -r_1 \cos(\theta - q_1) - r_2 \cos(\theta - q_1 - q_2) & 0 \\ -r_2 \cos(\theta - q_1 - q_2) & 0 \end{bmatrix} \quad (114)$$

$$\mathbf{J}_c^{-1}(\mathbf{q})(\mathbf{S}) = \frac{1}{r_1 r_2 \sin(q_2)} \begin{bmatrix} 0 & -r_2 \cos(\theta - q_1 - q_2) \\ 0 & -r_1 \cos(\theta - q_1) - r_2 \cos(\theta - q_1 - q_2) \end{bmatrix} \quad (115)$$

$$\det[\mathbf{J}_c(\mathbf{q})] = r_1 r_2 \sin(q_2) \quad (116)$$

$$\lambda_c = \begin{cases} x_c = r_1 \sin(\theta - q_1) + r_2 \sin(\theta - q_1 - q_2) \\ y_c = r_1 \cos(\theta - q_1) + r_2 \cos(\theta - q_1 - q_2) \end{cases} \quad (117)$$

$$\mathbf{J}_c^T(\mathbf{q})(\mathbf{I} - \mathbf{S}) = \begin{bmatrix} 0 & -r_1 \sin(\theta - q_1) - r_2 \sin(\theta - q_1 - q_2) \\ 0 & -r_2 \sin(\theta - q_1 - q_2) \end{bmatrix} \quad (118)$$

$$\mathbf{J}_c^{-1}(\mathbf{q})(\mathbf{S}) = \frac{1}{r_1 r_2 \sin(q_2)} \begin{bmatrix} -r_2 \sin(\theta - q_1 - q_2) & 0 \\ -r_1 \sin(\theta - q_1) - r_2 \sin(\theta - q_1 - q_2) & 0 \end{bmatrix} \quad (119)$$

onde a matriz  $\mathbf{I}$  é a matriz identidade ( de dimensão  $n \times n$  ) e a matriz de selecção  $\mathbf{S}$  é uma matriz diagonal ( de dimensão  $n \times n$  ) com elementos iguais a um nas direcções controladas em posição e zeros nas direcções controladas em força [8][16][17].

## 2.9 Conclusões

Neste capítulo apresentaram-se os robôs RR e RP, e o seu espaço de trabalho.

Foi também feita referência às cinemáticas directa e inversa de posição, velocidade e aceleração. Em seguida foi implementado o planeamento de movimentos rectilíneo, utilizando uma interpolação linear entre dois pontos, com a possibilidade de diferentes perfis de velocidade. Desenvolveram-se as dinâmicas directa e inversa e analisaram-se vários fenómenos dinâmicos nas juntas.

A partir dos modelos, fez-se uma análise do controlo dos robôs, respectivamente ao nível da posição da velocidade e da força. Apresentaram-se diversos métodos alternativos de controlo de posição, quer no espaço de trabalho, quer no espaço das juntas.

Para o controlo de posição/força apresentou-se o algoritmo de controlo híbrido, assim como a modelização da superfície de restrição. Desta forma, estabeleceu-se uma comparação das respostas do sistema quando se adoptam modelos do ambiente de primeira e de segunda ordem, ou quando existem fenómenos dinâmicos nas juntas.

## CAPÍTULO 3

### O Programa de Simulação *RobLib*

#### 3.1 Introdução

O programa desenvolvido, intitulado *RobLib* (*Robot Library*), permite a simulação de robôs e de algoritmos de controlo[18][19]. O programa disponibiliza, ao utilizador, robôs de duas juntas (com estruturas RR e RP), assim como um conjunto de algoritmos de controlo de posição e de força.

Para cada robô o programa apresenta as seguintes funcionalidades:

- visualização dos parâmetros cinemáticos e dinâmicos e de uma figura esquemática do robô;
- planeamento de trajectórias rectilíneas através da definição dos pontos inicial e final, no espaço cartesiano ou, alternativamente, no espaço das juntas;
- “animação” do robô de modo a obter-se uma melhor percepção da trajectória planeada;
- possibilidade de escolha de diversas estratégias e algoritmos de controlo de robôs.
- integração de uma base de dados no programa de simulação *RobLib*.
- apresentação gráfica das variáveis de entrada e de saída do robô, nomeadamente, a posição, a velocidade, a aceleração e os binários, bem como os seus erros em relação à trajectória de referência.

O programa foi desenvolvido na linguagem Delphi. A selecção da linguagem de programação baseou-se no facto de ser essencial a criação de um *interface* gráfico poderoso, de fácil manipulação, e suportado pelo ambiente Windows. O sistema de programação em Delphi permite criar aplicações atractivas, robustas e que aproveitam ao máximo o *interface* gráfico com o utilizador [20]. O programa foi desenvolvido na língua inglesa tendo em vista a utilização do programa por um vasto leque de potenciais interessados.

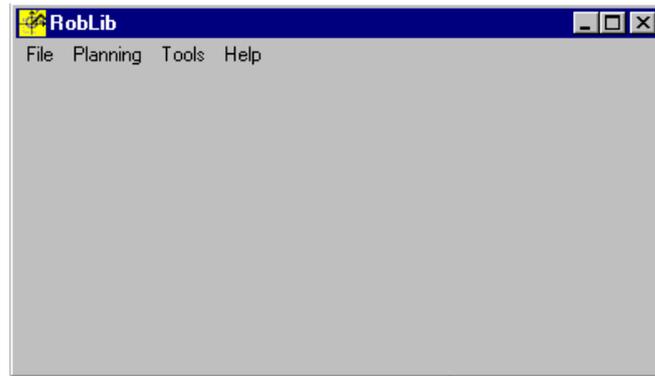
Este capítulo é constituído por seis secções onde são apresentadas as funcionalidades do programa desenvolvido. Assim, na primeira secção são apresentados os robôs disponíveis e os fenómenos dinâmicos nas juntas. Nas segunda e terceira secções descrevem-se, respectivamente, a estrutura do programa e o planeamento de trajetórias. De seguida, nas secções quatro e cinco analisam-se os algoritmos de controlo e a base de dados integrada no programa. Por último, na secção seis, mostram-se resultados obtidos em diversas simulações.

## **3.2 Robôs Disponíveis**

Os robôs a simular no programa são de duas juntas com estruturas do tipo RR ou RP. Consideram-se robôs não só com eixos ideais, mas também com fenómenos nas juntas tais como folgas, flexibilidade, atritos não-lineares, saturação nos actuadores e quantificação (*i.e.* resolução finita nos sensores).

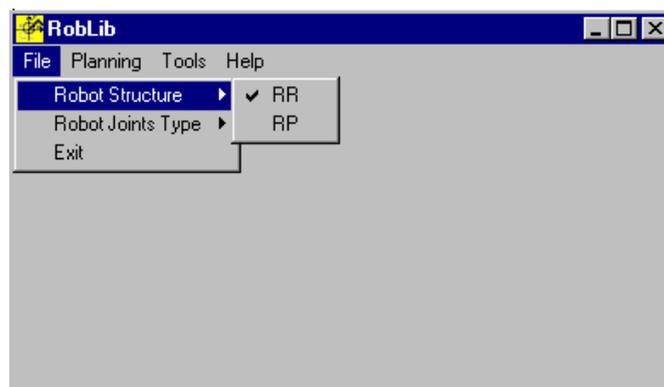
## **3.3 Estrutura Geral do RobLib**

O programa *RobLib* apresenta um esquema de funcionamento típico do ambiente Windows, utilizando janelas que são abertas em resposta a eventos desencadeados pelo utilizador. A janela de abertura do programa está ilustrada na figura 3.1 e permite o acesso imediato de todas as funcionalidades do programa.

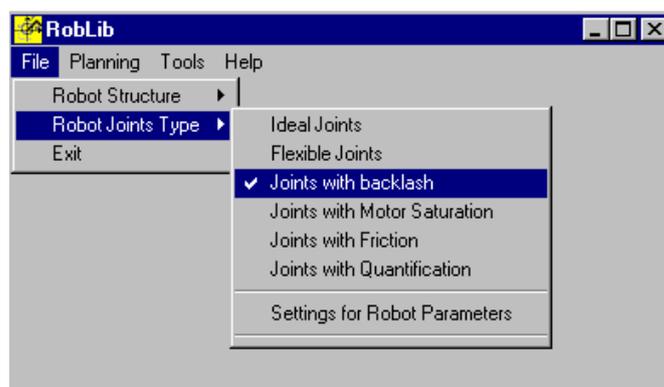


**Figura 3.1-** Janela Principal do programa *RobLib*.

A opção *File* apresenta três opções, *Exit* para sair do programa, *Robot Structure* para a configuração do robô (figura 3.2) e *Robot Joints Type*, para a selecção do tipo de fenómeno a acrescentar nas juntas do robô previamente seleccionado (figura 3.3).



**Figura 3.2 –** Opção da escolha do tipo de robô.



**Figura 3.3–** Opção da escolha do tipo de fenómenos nas juntas.

A figura 3.4 ilustra uma das possíveis estruturas das juntas do robô RR. Após a selecção de uma dada estrutura, o *RobLib* apresenta um quadro com os parâmetros do robô.

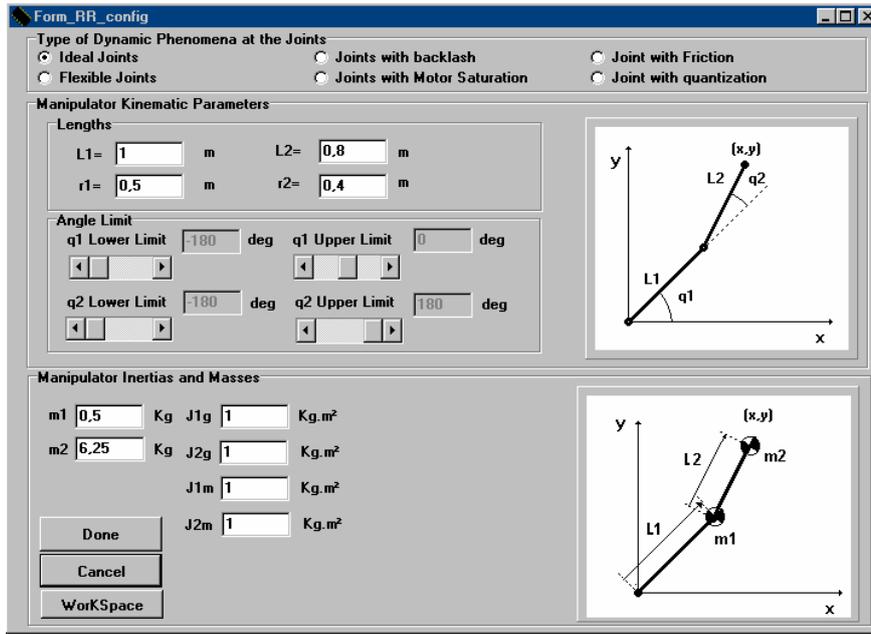


Figura 3.4 - Janela de visualização dos parâmetros do robô RR.

Na figura 3.5 tem-se a janela de parametrização, correspondente ao robô RP.

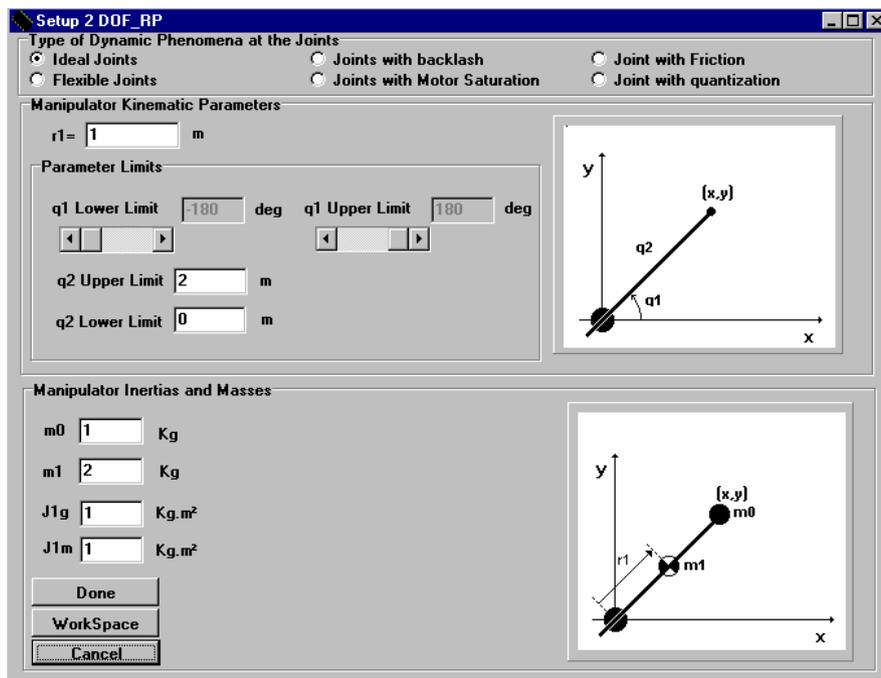
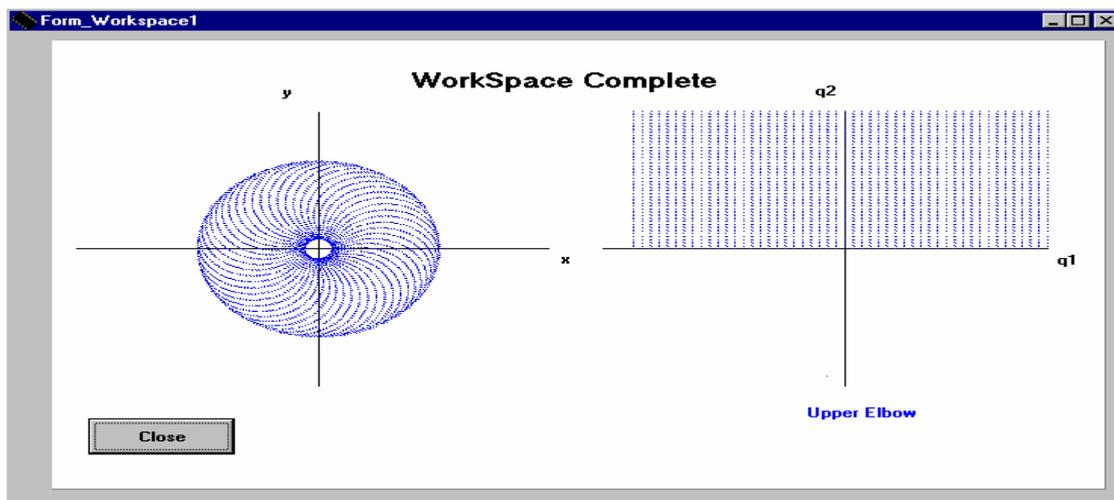


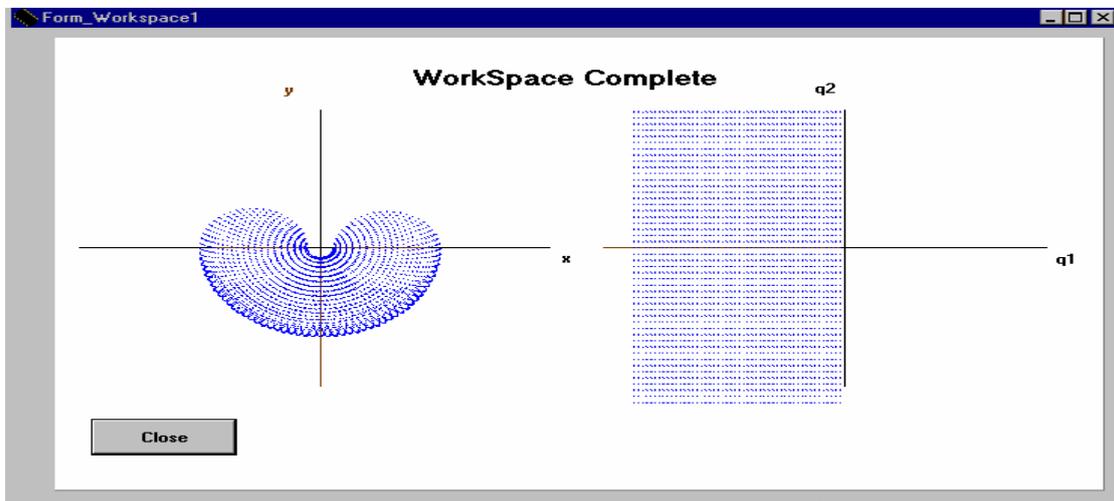
Figura 3.5 - Janela de visualização dos parâmetros do robô RP.

Nas janelas das figuras 3.4 e 3.5 podem definir-se os parâmetros das limitações das juntas, o que produz diferentes áreas de trabalho que se podem visualizar com a opção *Workspace* (figuras 3.6, 3.7 e 3.8).

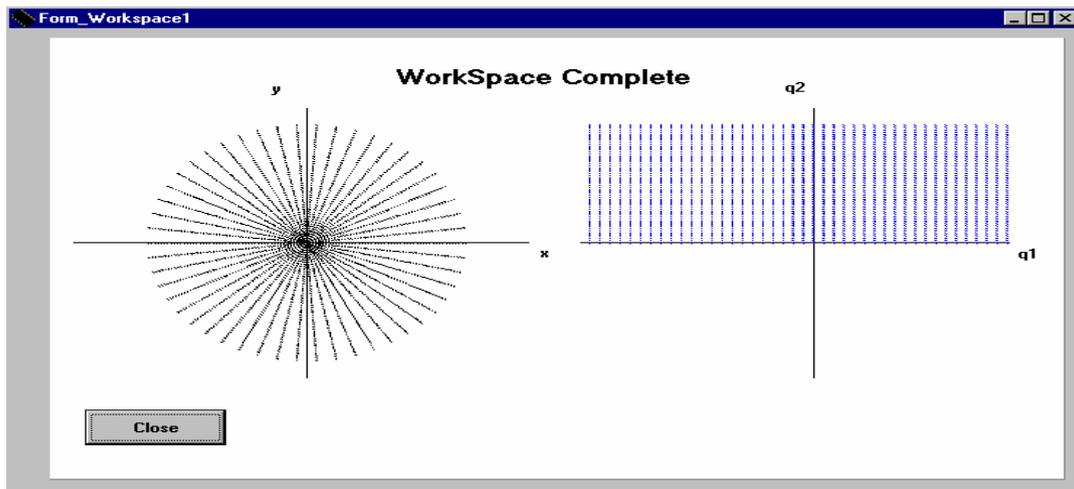
A opção *Done* aceita os dados inseridos e a opção *Cancel* mantém os dados anteriores.



**Figura 3.6** - Janela de visualização da região de trabalho do robô RR sem limitações nas juntas  $\{q_1, q_2\}$ .

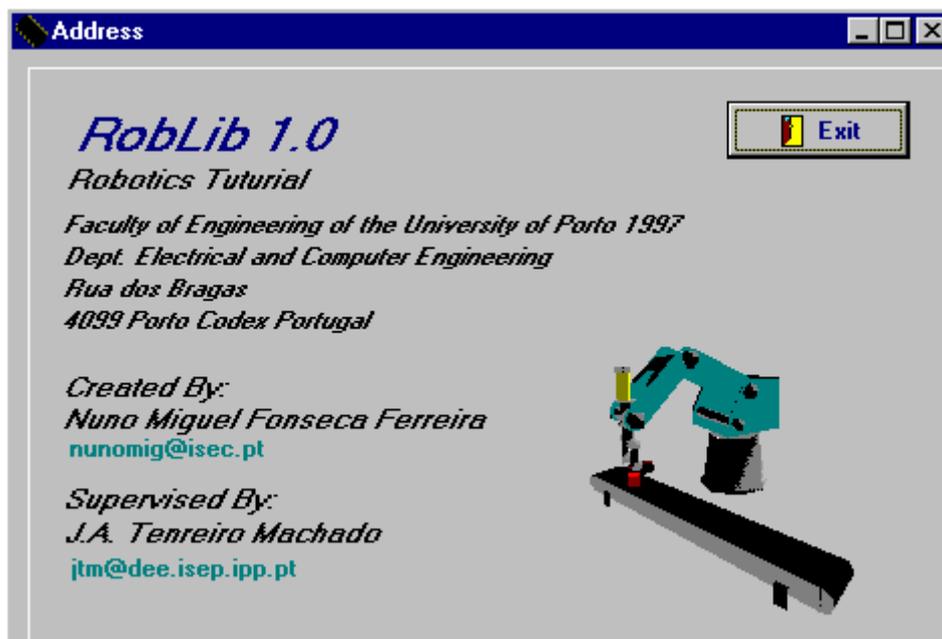


**Figura 3.7** - Janela de visualização da região de trabalho do robô RR com limitações nas juntas (*i.e.* com  $-180^\circ \leq q_1 \leq 0^\circ$ ).



**Figura 3.8** - Janela de visualização da região de trabalho do robô RP sem limitações nas juntas.

A opção *Help* permite o acesso à janela informativa do programa *RobLib* (figura 3.9) e ao manual do programa (figura 3.10).



**Figura 3.9** –Janela informativa.

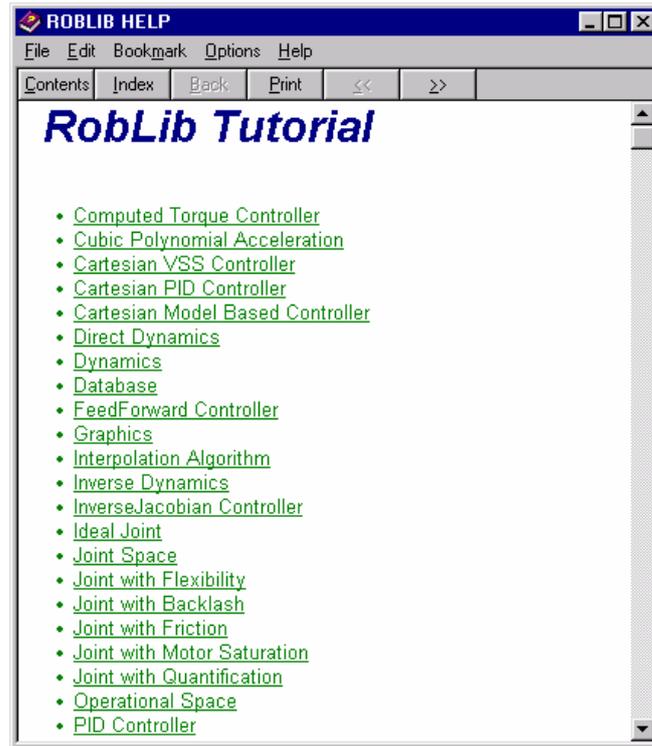


Figura 3.10 – Janela do manual do utilizador.

### 3.4 Planeamento de trajectórias de posição

A opção *Trajectory* (figura 3.11) permite efectuar o planeamento de trajectórias rectilínea a partir da definição de um ponto inicial e de um ponto final. Este planeamento pode ser feito introduzindo dados quer no espaço cartesiano quer no espaço das juntas (figura 3.12).

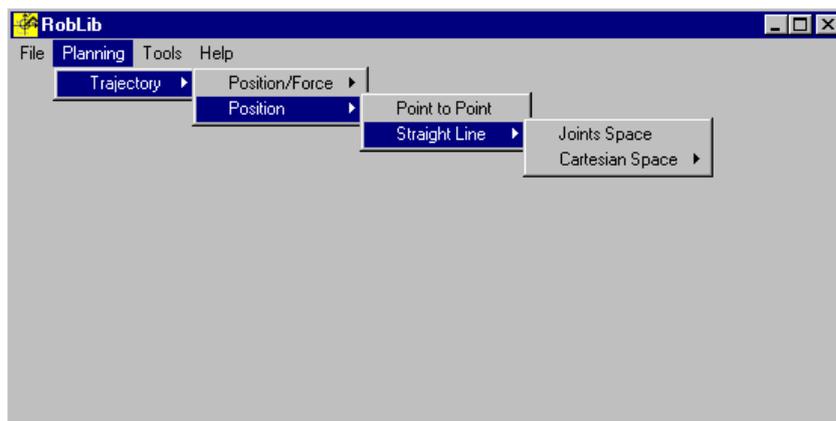


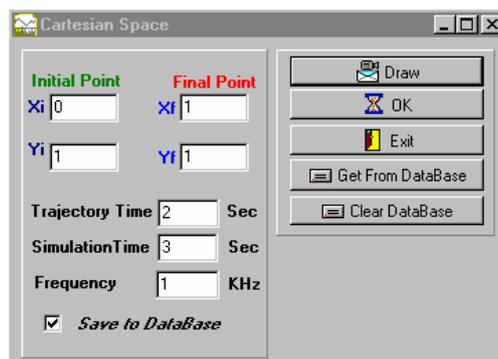
Figura 3.11 – Opção Trajectory para escolha do tipo de trajectória.

Quando a introdução dos dados é feita no espaço cartesiano, se os valores inseridos pelo utilizador ultrapassarem os limites do espaço de trabalho então surgirá um aviso que indica essa situação. Os parâmetros essenciais para o planeamento de uma trajectória com o *RobLib* são: o tempo de simulação, o tempo para efectuar a trajectória, os pontos inicial e final e a frequência pretendida no planeamento (figura 3.12). O tempo de simulação permite que o *RobLib* continue a executar a simulação após ter executado a trajectória no tempo definido de controlo para se observar os comportamentos dos diversos algoritmos.

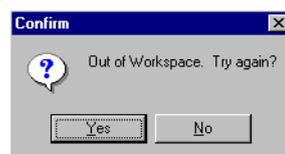
A introdução dos valores das juntas é feito a partir de uma janela semelhante à do caso anterior (figura 3.12). Quando os dados inseridos pelo utilizador violarem os limites das juntas do robô, surge também um aviso indicando essa situação (figura 3.13).

A opção *Go To DataBase* permite o acesso à base de dados, indicando ao utilizador as trajectórias já armazenadas para o tipo de planeamento seleccionado. Desta forma, o utilizador tem a possibilidade de obter uma trajectória planeada anteriormente.

A opção *Clear From DataBase* permite apagar trajectórias armazenadas abrindo a janela da base de dados. A gravação de novas trajectórias na base de dados é realizada seleccionando a “caixa” *Save to DataBase*. Assim, após a simulação da trajectória grava-se a informação correspondente na base de dados.



**Figura 3.12** - Janela de inserção dos parâmetros da trajectória (no espaço operacional).



**Figura 3.13** - Janela de aviso dos limites.

A opção *Draw* abre uma janela (figuras 3.14 e figura 3.15, consoante a estrutura do robô) de modo a que o utilizador possa visualizar o movimento que foi planeado.

Para as opções existentes na janela da figura 3.12, relacionadas com a base de dados (*i.e.* procurar ou apagar informação), surge a janela da figura 3.16 que apresenta o conteúdo da base de dados. Para obter uma trajetória simulada deve-se fazer um duplo *Click* no botão esquerdo do rato, na trajetória pretendida. Se a opção for “apagar” o procedimento é similar.

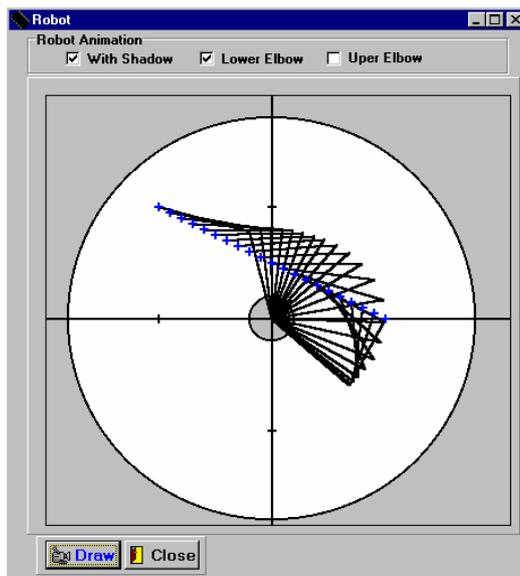


Figura 3.14- Janela do robô RR em movimento.

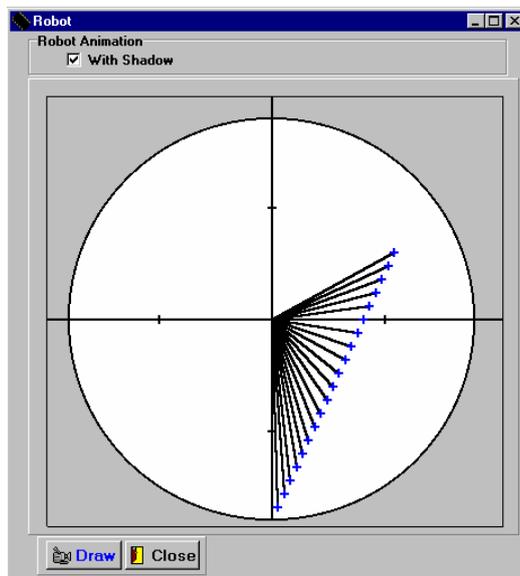
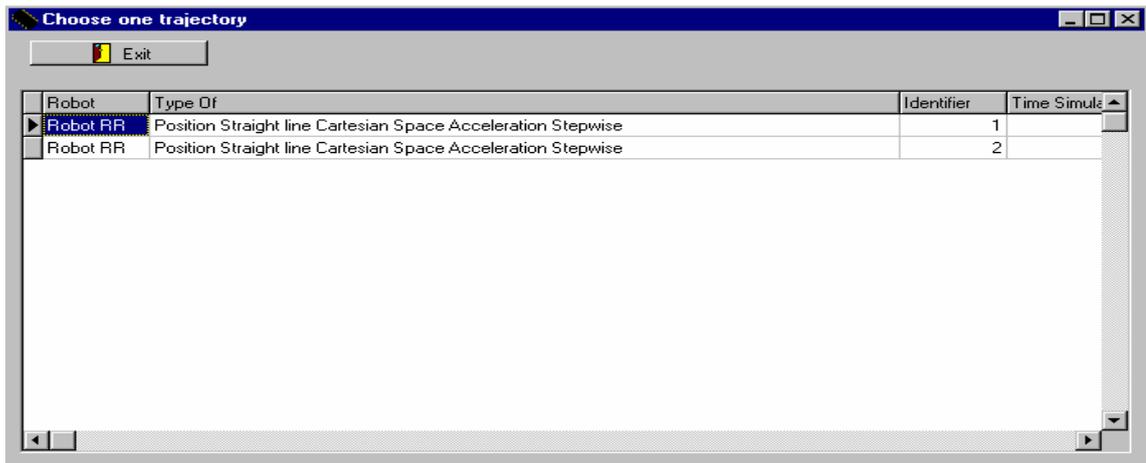
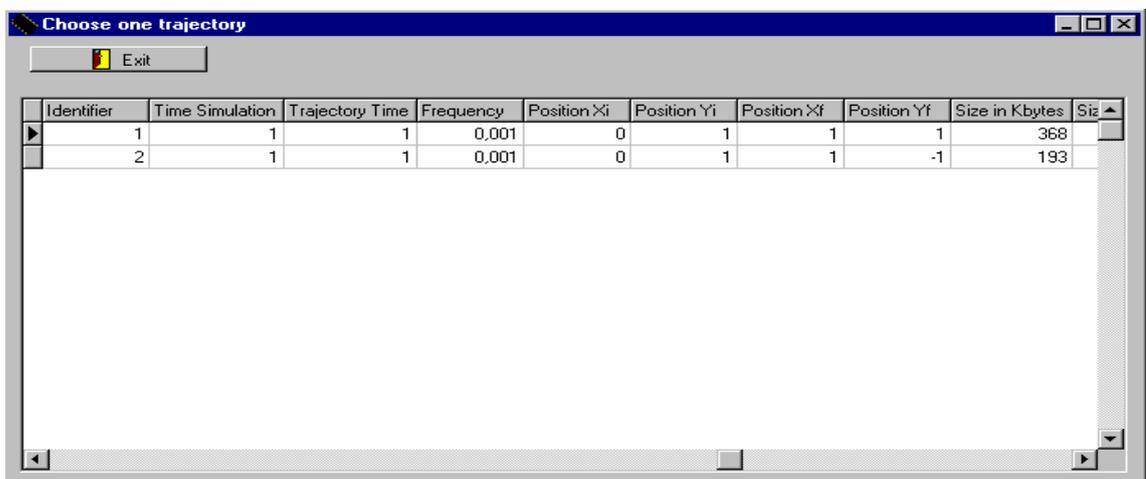


Figura 3.15- Janela do robô RP em movimento.



**Figura 3.16** - Janela da base de dados das trajectórias.



**Figura 3.17** - Janela da base de dados das trajectórias.

Apresentam-se nas figuras 3.16 e 3.17 os campos existentes na selecção da trajectória de posição armazenada na base de dados.

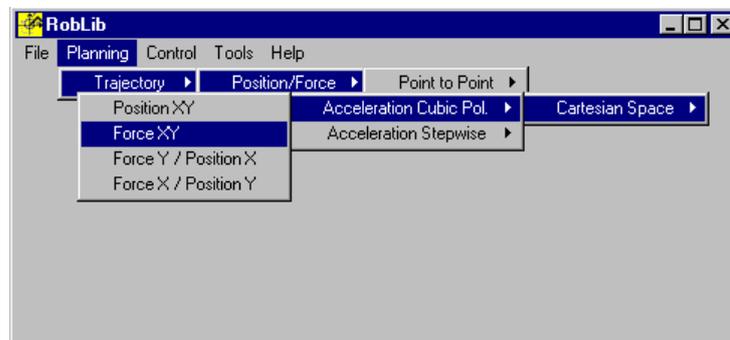
Após obtida uma trajectória (através do planeamento ou através da base de dados) fica acessível a opção *Control* que dá acesso aos diversos algoritmos de controlo.

### 3.5 O planeamento da trajectória de posição e força

O planeamento de uma trajectória, envolvendo as variáveis posição e força, apresenta algumas diferenças relativamente ao planeamento com, apenas, a posição.

Para tal, faz-se a separação dos dois tipos de planeamentos na opção *Planning*.

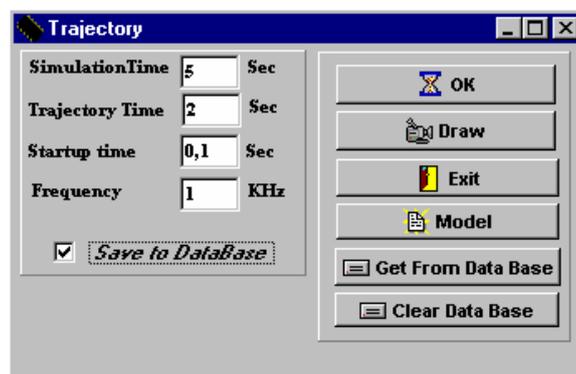
No planeamento de força modeliza-se o ambiente (*i.e.* a superfície de contacto onde se pretende exercer a força), bem como a “combinação” das variáveis conforme se ilustra na figura 3.18. Por outras palavras, o planeamento pode ser de posição, ou de força em ambos os eixos ou então de posição num eixo e força no outro eixo).



**Figura 3.18** – Janela de opção de planeamento de trajectória de posição e de força.

Qualquer que seja a opção de planeamento seleccionada, abre-se uma janela de introdução dos dados para o cálculo da respectiva trajectória (figura 3.19). Esta janela é idêntica à janela de introdução dos dados do planeamento de posição, acrescentando o parâmetro da introdução do início do tempo para o controlo de força.

Se os valores inseridos, pelo utilizador, ultrapassarem os limites do espaço de trabalho, surgirá um aviso similar ao existente no planeamento de posição.



**Figura 3.19** – Janela de introdução dos dados.

O botão *Model* permite a alteração do modelo do ambiente através da janela ilustrada na figura 3.20. O modelo do ambiente é modelizado através de um sistema com mola, massa e amortecimento. Desta forma, a modelização da superfície nos eixos  $x$  e  $y$  é realizada através das constantes  $M_i$ ,  $B_i$  e  $K_i$  ( $i=1,2$ ) que representam a massa, a constante de amortecimento e a mola onde o índice  $i$  representa os dois eixos. A variável  $\theta$  permite definir a inclinação (em graus) da superfície restrição. Após as

alterações do modelo do ambiente o utilizador deve ainda indicar os pontos inicial e final da trajectória a percorrer. Caso estes passos não sejam feitos o simulador executa a simulação com os parâmetros pré-defenidos.

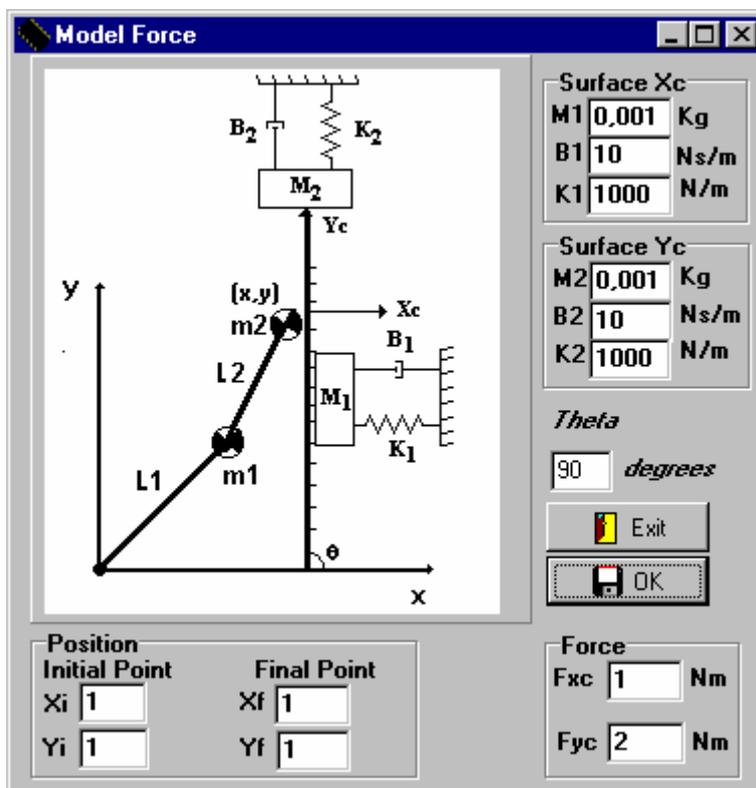
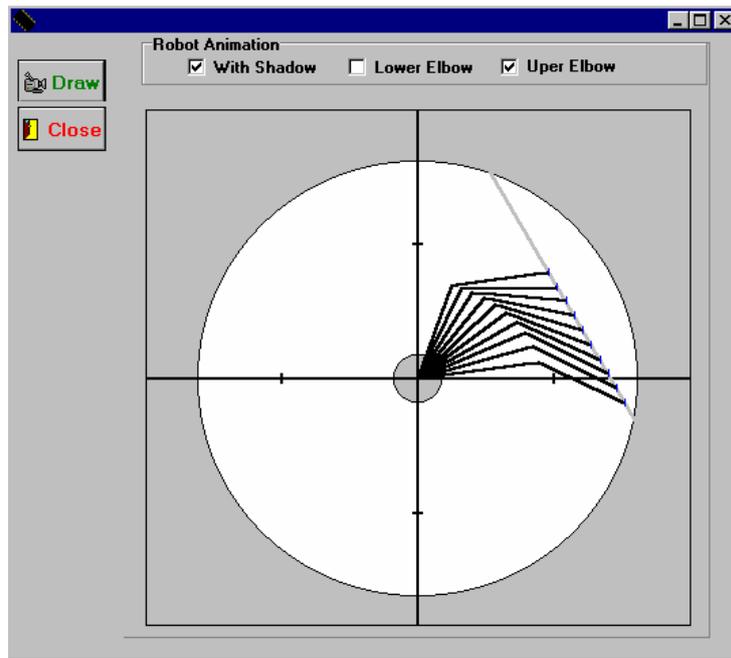


Figura 3.20 – Janela da introdução do modelo do ambiente.

A opção *OK* permite validar os dados do modelo do ambiente, e da trajectória (posição e força) desejada. A opção *Exit* permite sair desta janela sem realizar qualquer alteração dos parâmetros.

A opção *Draw* da figura 3.19 permite a visualização do movimento (desejado) do robô em interacção com o ambiente, como se apresenta na figura 3.21.



**Figura 3.21** – Movimento (desejado) do robô em interacção com o ambiente.

A opção de controlo fica acessível após o planeamento da trajectória. Esta opção depende do tipo de planeamento seleccionado. Se a trajectória planeada for de posição, apenas se tem acesso aos algoritmos de controlo de posição; Se o planeamento tiver em conta a posição e a força de contacto no ambiente, então fica acessível a opção de controlo dos algoritmos de posição e força desactivando apenas ou somente os algoritmos de controlo de posição.

### 3.6 Algoritmos de Controlo de Posição

O modo de selecção dos diversos algoritmos de controlo existentes no *RobLib* pode analisar-se na figura 3.22



**Figura 3.22** - Janelas de selecção dos controladores de posição.

As estratégias de controlo de posição apresentadas no *RobLib* estão agrupadas em algoritmos no espaço das juntas e no espaço operacional.

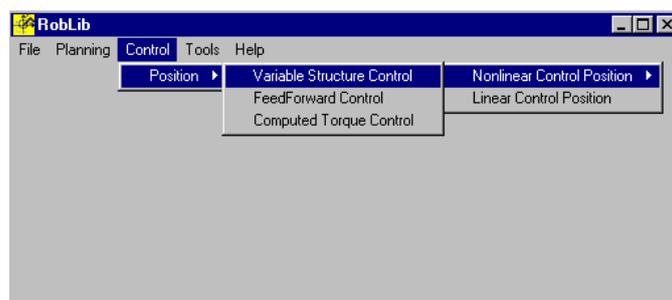
No espaço das juntas foram implementados:

- controladores não lineares, nomeadamente, os controladores de estrutura variável (VSS);
- controladores lineares (PID);
- controladores baseados em modelos, tais como: o controlador do binário calculado (*computed torque*) e os controladores do tipo por acção antecipativa (*feedforward*) com sub-controladores lineares ou não lineares;

No espaço operacional foram implementados:

- controladores não lineares de estrutura variável com o jacobiano transposto (*Transpose Jacobian Controller*);
- controladores não lineares de estrutura variável com o jacobiano inverso (*Inverse Jacobian Controller*);
- controladores com acção proporcional, integral e derivativa (PID) com o jacobiano transposto (*Transpose Jacobian Controller*);
- controladores lineares com acção proporcional, integral e derivativa (PID) com o jacobiano inverso (*Inverse Jacobian Controller*);
- controlador baseado no modelo do robô.

Os diversos algoritmos de controlo apresentam um modo idêntico de diálogo com o utilizador. Por exemplo, seleccionando um controlador do tipo não linear de estrutura variável, com um controlo no espaço das juntas (figura 3.23) abre-se uma janela correspondente a este tipo de controlador (figura 3.24).



**Figura 3.23** – Janela da opção do controlador de posição não linear do tipo CEV com um controlo no espaço das juntas.

No caso de um controlador de estrutura variável, inicialmente é activado o controlador que adopta um modelo de primeira ordem (MPO) para a superfície de deslizamento.

Para um CEV que adopte um MPO na superfície de deslizamento, existem duas possibilidades para realizar a correspondente equação diferencial, que são a utilização do sinal de erro e de sua derivada (*i.e.* proporcional e derivativo, MPO-PD ) ou a utilização do sinal do erro e do seu integral (*i.e.* proporcional e integral, MPO-PI).

O algoritmo CEV-MPO-PD vem:

$$\sigma = \dot{q}_e + c \cdot q_e \quad (120)$$

O algoritmo CEV-MPO-PI :

$$\sigma = \dot{q}_e + c \cdot \int q_e \cdot dt \quad (121)$$

Para os modelos de segunda ordem (MSO) tem-se o algoritmo:

o CEV-MSO-PDD<sup>2</sup>:

$$\sigma = \ddot{q}_e + 2\xi\omega_{ni} \dot{q}_e + \omega_{ni}^2 q_e \quad (122)$$

e o algoritmo CEV-MSO-PID:

$$\sigma = \dot{q}_e + 2\xi\omega_{ni} q_e + \omega_{ni}^2 \int q_e dt \quad (123)$$

onde o parâmetro  $c$  é o valor próprio,  $q_e$  é a entrada do CEV,  $\sigma$  é a curva de deslizamento, os parâmetros são  $\xi$  o coeficiente de amortecimento e  $\omega_n$  é a frequência natural não amortecida.

O segundo bloco é constituído pela acção de controlo. Uma expressão usada frequentemente por este bloco é dada por:

$$T_i = \begin{cases} T_{\max} & , T \geq \frac{T_{\max}}{K_i} \\ K_i \cdot T & , |T| < \frac{T_{\max}}{K_i} \\ -T_{\max} & , T \leq -\frac{T_{\max}}{K_i} \end{cases} \quad (124)$$

Neste caso, evita-se uma acção “intempestiva” do tipo relé já que existe uma banda “suave” com um ganho proporcional  $K_i$ .

A janela correspondente apresenta um diagrama de blocos do sistema, os modelos para a curva de deslizamento, as acções de controlo, os ganhos do controlador e a frequência de amostragem. A janela possibilita a alteração dos parâmetros e dos fenómenos das juntas e permite, também, aceder á base de dados.

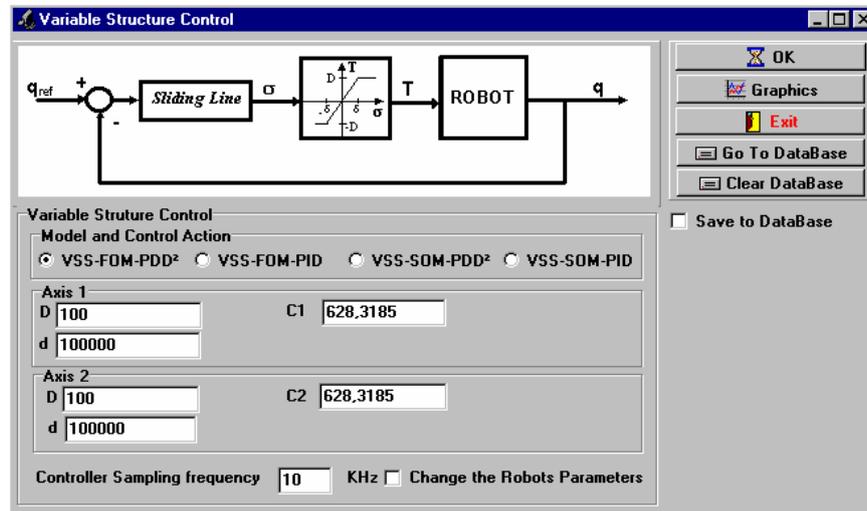


Figura 3.24 - Controladores de estrutura variável.

O botão *OK* inicia a simulação, considerando as opções seleccionadas (robô, tipo de controlo, modelo e acção de controlo) criando um ficheiro (com formato texto) que contém os valores das variáveis de posição, velocidade e aceleração (em ambos os espaços) e os binários nas juntas. Quando se processa a simulação aparece, um *status bar* permitindo uma visualização do progresso da simulação, como se ilustra na figura 3.25.

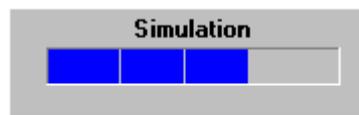


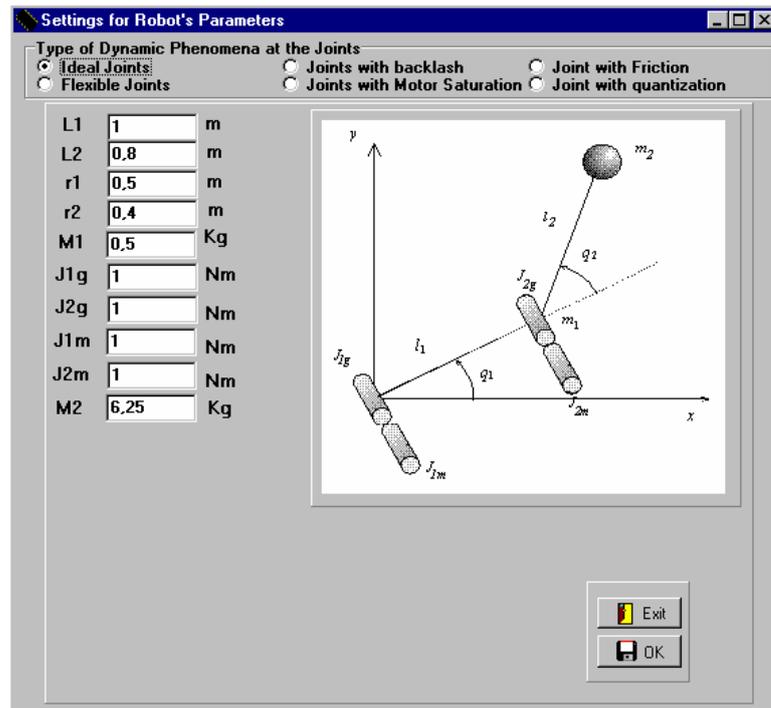
Figura 3.25 - Janela de visualização do progresso da simulação.

O botão *Exit* permite o retorno para a janela anterior e a opção *Save to Database* (se estiver seleccionada) permite guardar as simulações.

O botão *Go To Database* permite aceder á base de dados mostrando todas as simulações já realizadas com o mesmo tipo de controladores e para a trajectória seleccionada.

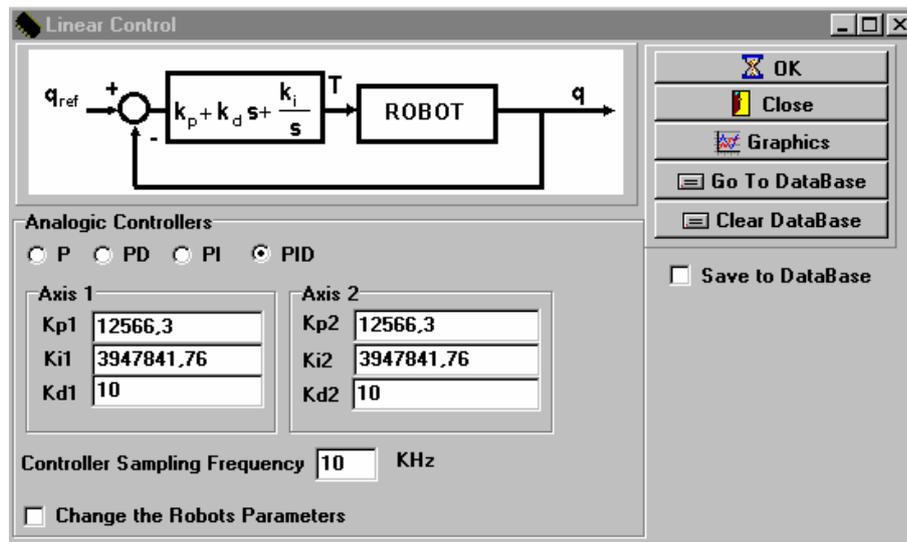
O botão *Graphics* permite visualizar todas as variáveis, mostrando a evolução temporal das variáveis e permitindo, assim, a comparação da trajetória simulada com a trajetória desejada, o que facilita a comparação com outras simulações.

A alteração dos parâmetros das juntas do robô está apresentada na figura 3.26. A janela ilustra o tipo de junta seleccionada bem como os seus parâmetros e mostra um esquema do robô, possibilitando uma maior versatilidade na selecção do tipo de fenómenos dinâmicos nas juntas.



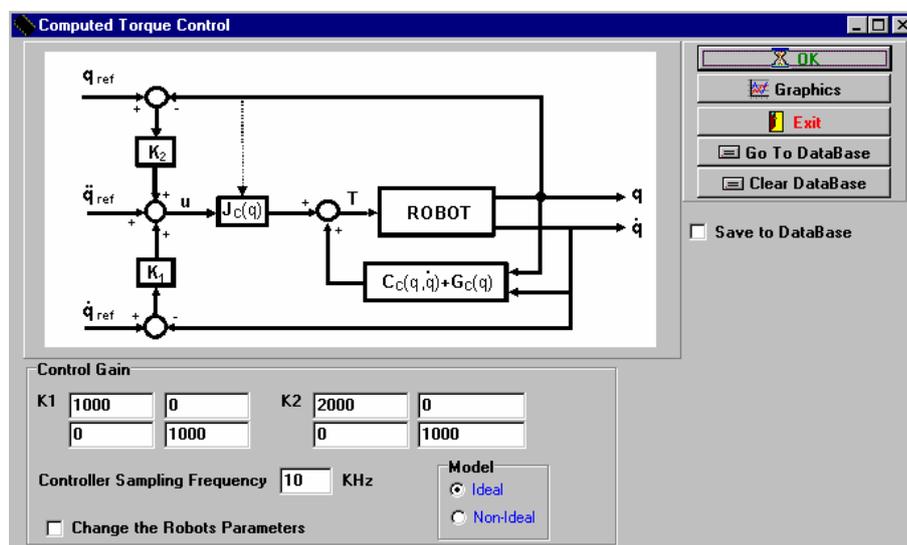
**Figura 3.26-** Janela da alteração dos parâmetros e fenómenos dinâmicos das juntas do robô.

Se a estratégia de controlo pretendida for no espaço das juntas e o controlador for linear, então a janela apresenta opções idênticas ao caso anterior, como se apresenta na figura 3.27.



**Figura 3.27** – Janela de selecção de opções para um controlador linear no espaço das juntas.

A janela da figura 3.28 apresenta um controlador baseado no modelo do robô do tipo *Computed Torque*. Se se pretender um modelo diferente do manipulador então o utilizador pode alterar os parâmetros do modelo através da opção existente na janela. Desta forma, abre-se uma janela que permite a alteração dos parâmetros do modelo de uma forma idêntica à da figura 3.26.



**Figura 3.28** - Controlador do tipo *Computed Torque*.

A janela da figura 3.29 apresenta um controlador baseado no modelo do robô do tipo *Feedforward*, onde a selecção do modelo do robô é idêntica ao controlador *Computed Torque*. Neste tipo de estratégia de controlo, o sub-controlador inserido na malha de realimentação pode ser linear (figura 3.29) ou não linear (figura 3.30).

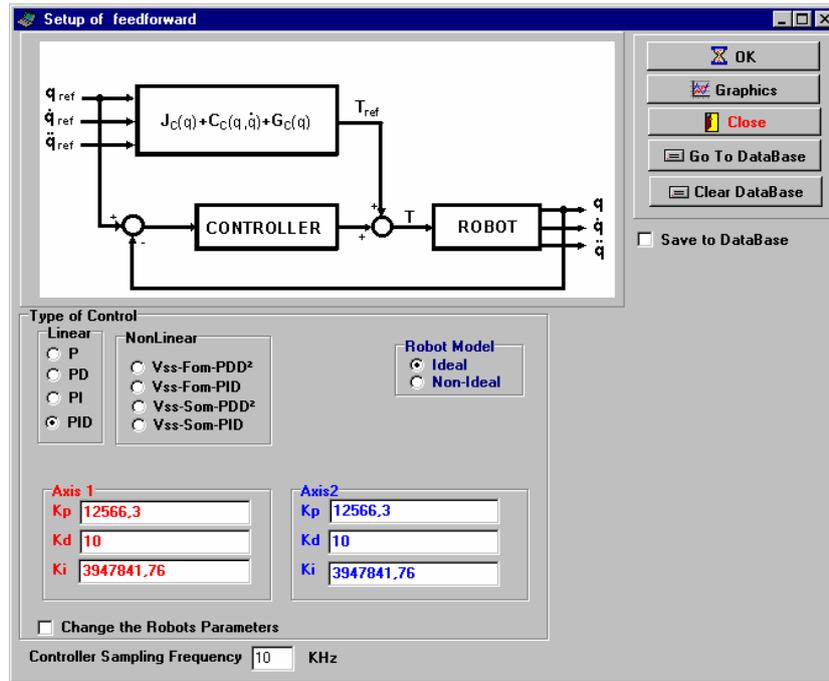


Figura 3.29 - Controlador do tipo *Feedforward* com sub-controlador do tipo PID.

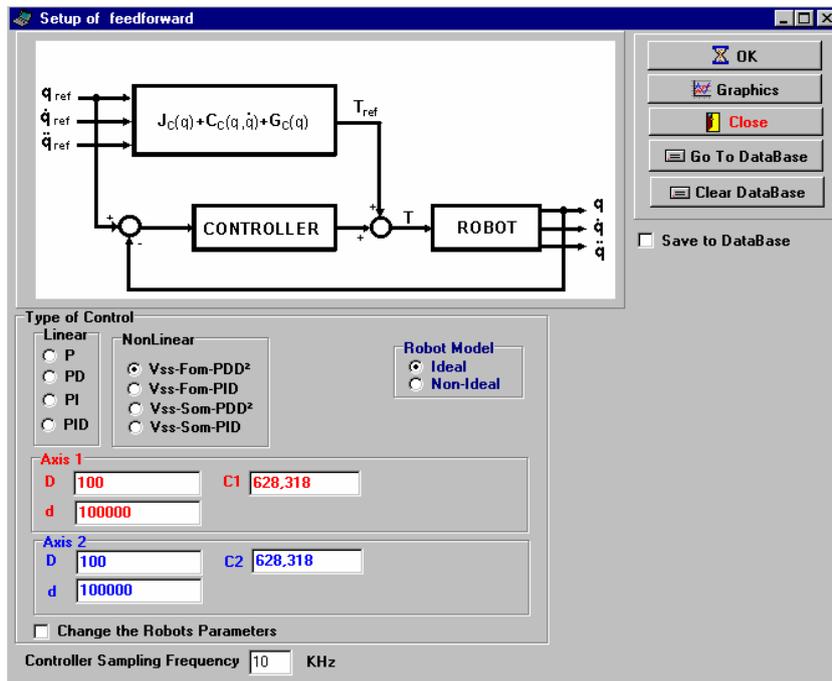
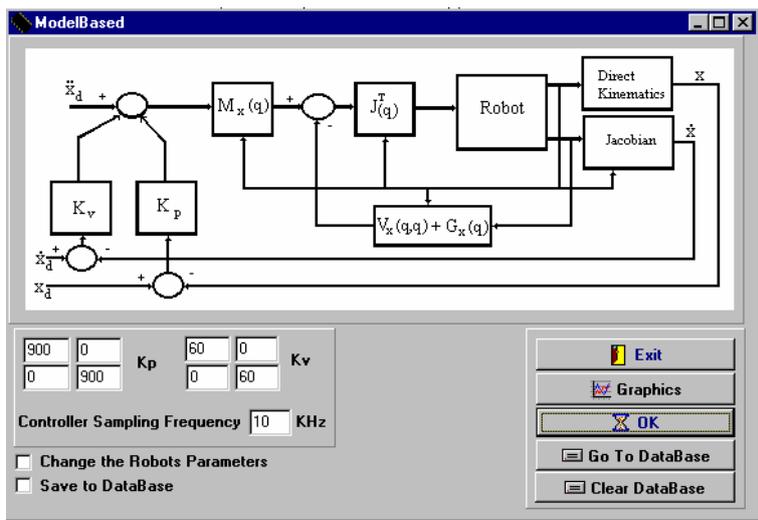


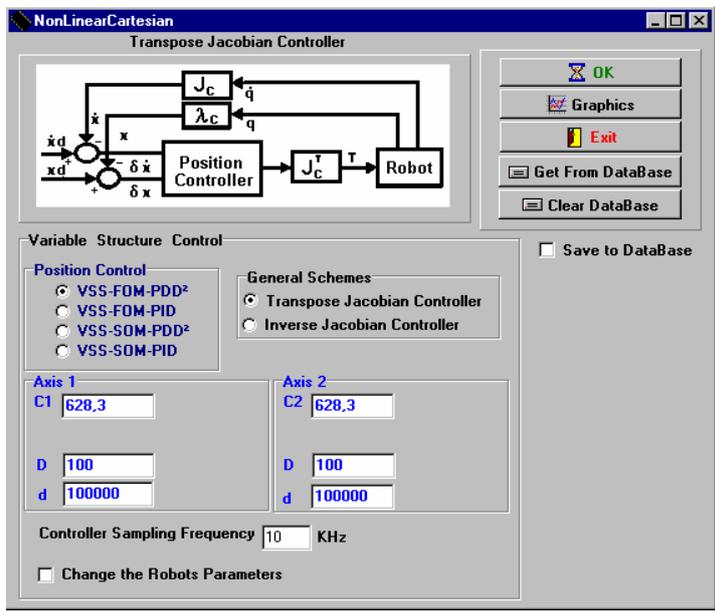
Figura 3.30 - Controlador do tipo *Feedforward* com sub-controlador do tipo VSS-FOM-PD<sup>2</sup>.

Os algoritmos seguintes optam por um controlo no espaço operacional. Por exemplo, na figura 3.31 apresenta-se um controlador baseado no modelo do robô.

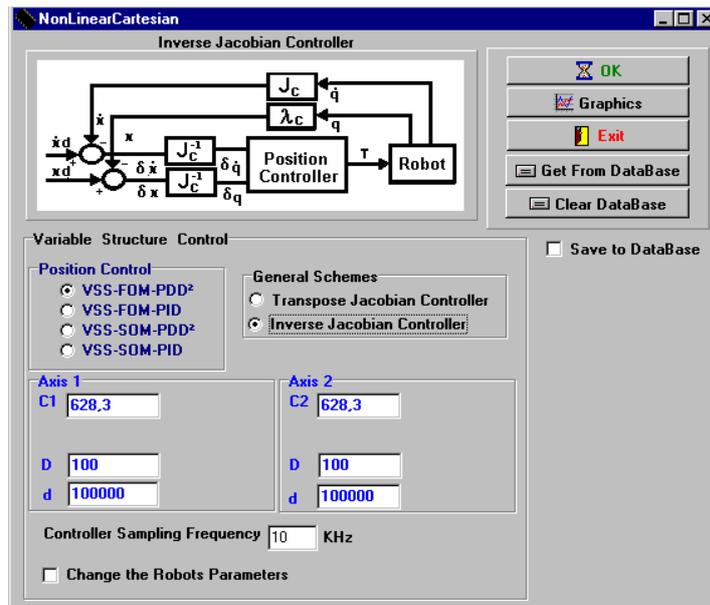


**Figura 3.31** – Controlo baseado no modelo do robô e com acção no espaço operacional.

Por seu lado nas figuras 3.32 e 3.33 apresentam-se dois esquemas de controlo no espaço operacional em que o controlador é não linear (nomeadamente com os sub-controladores de estrutura variável), respectivamente, com o jacobiano transposto e o jacobiano inverso.

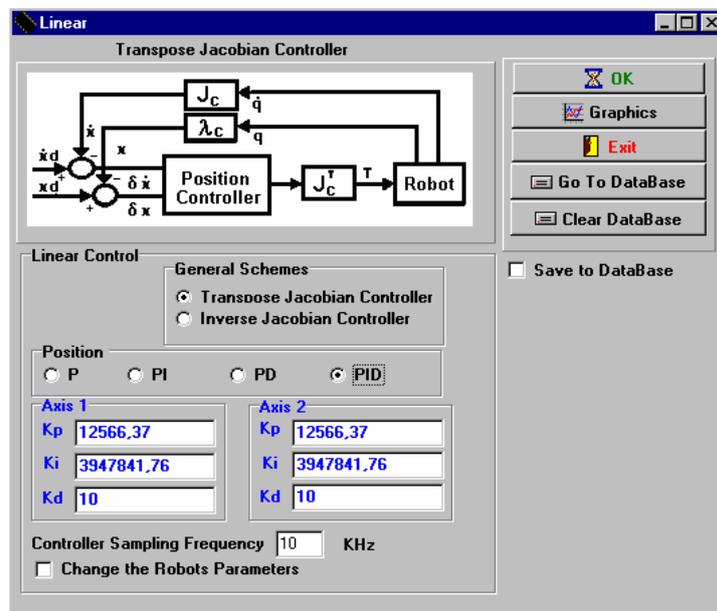


**Figura 3.32** - Controlo no espaço operacional, com sub-controladores de estrutura variável e com o jacobiano transposto.

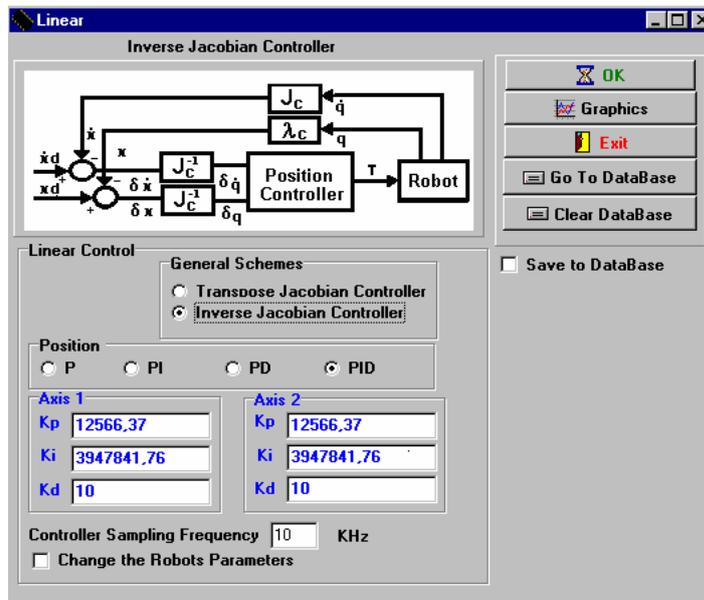


**Figura 3.33** - Controlo no espaço operacional, com sub-controladores de estrutura variável com o jacobiano inverso.

Alternativamente, as figuras 3.34 e 3.35 mostram dois esquemas de controlo similares aos anteriores com sub-controladores lineares do tipo PID. As opções existentes são idênticas às anteriores.



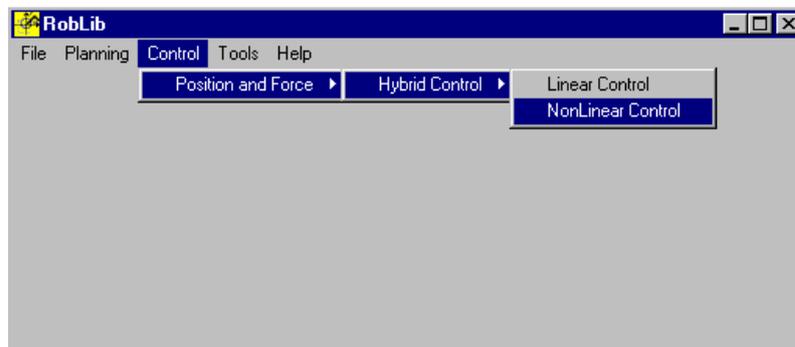
**Figura 3.34** - Controlo no espaço operacional, com sub-controladores PID com o jacobiano transposto.



**Figura 3.35** - Controlo no espaço operacional, com sub- controladores PID e com o jacobiano inverso.

### 3.7 Algoritmos de Controlo de Posição/Força

A opção de controlo de posição/força, apresenta-se na janela principal (figura 3.36). Esta opção é válida após o planeamento da trajectória e da interacção do robô com o ambiente.



**Figura 3.36** – Controlo de posição e de força.

Através da janela da figura 3.36 tem-se acesso ao controlador híbrido com duas sub-opções nomeadamente: os controladores não lineares e os controladores lineares.

Nas figuras 3.37 e 3.38 apresentam-se os controladores de posição/força, respectivamente, com características não lineares e lineares. Esta janela permite as mesmas funcionalidades existentes dos controladores de posição.

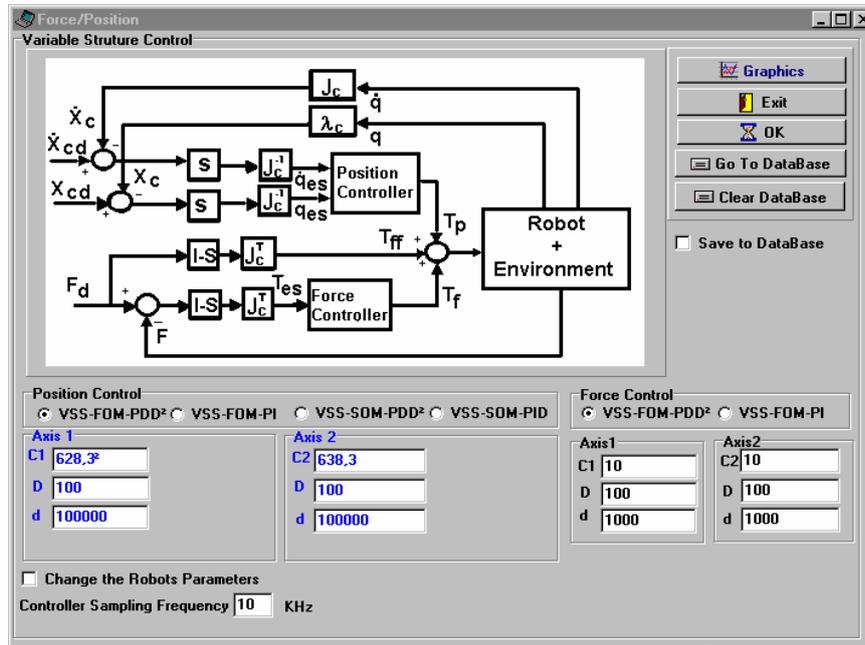


Figura 3.37 – Controlador híbrido com sub-controladores não lineares.

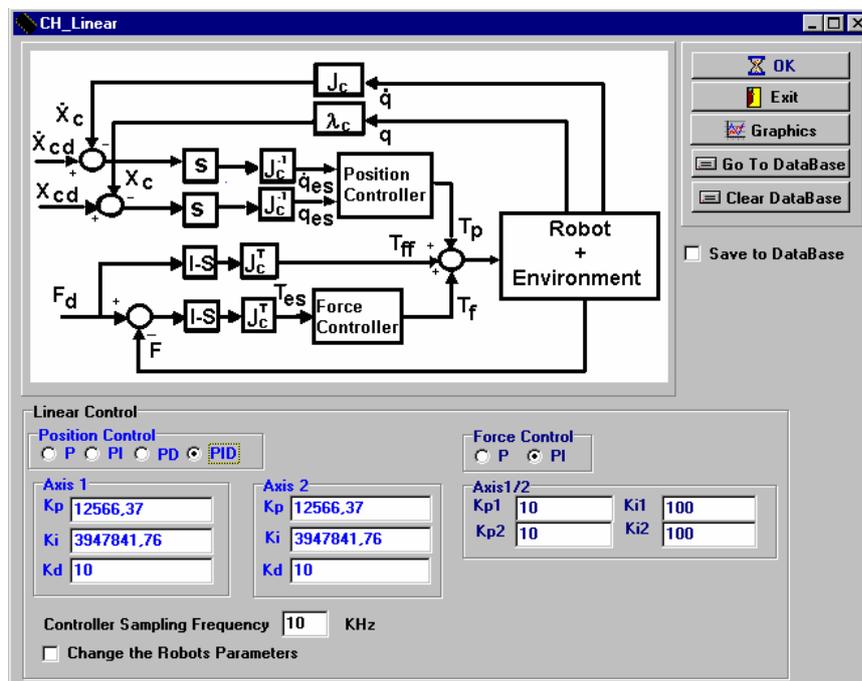


Figura 3.38 – Controlador híbrido com sub-controladores lineares.

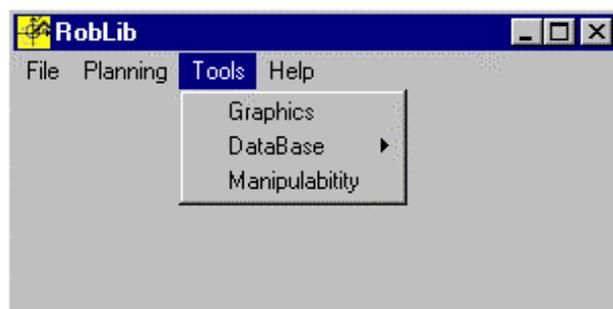
### 3.8 Base de dados

A base de dados integrada no simulador *RobLib* foi implementada em Delphi utilizando como plataforma de desenvolvimento o servidor local *Interbase*. Utiliza-se esta plataforma, porque permite incluir procedimentos *SQL* na definição da base de dados, visto que as outras plataformas, nomeadamente, o *Paradox* e o *Access*, não permitem esta possibilidade[21].

Através da opção *Tools* que se encontra na janela principal pode aceder-se às simulações já armazenadas na base de dados (opção *DataBase, Open* como ilustra a figura 3.39). O simulador *RobLib* na opção *Graphics* permite realizar consultas, somente, às simulações anteriores (figura 3.39).

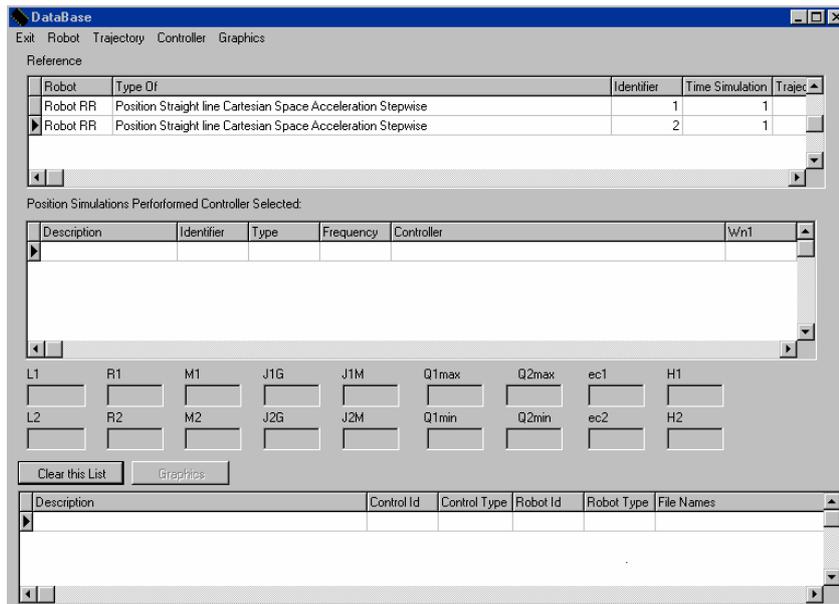
Através da opção *DataBase, Clear Everyting* o utilizador pode apagar a base de dados e todos os ficheiros criados pelas simulações.

A opção *Manipulability* será apresentada no capítulo 5 de uma forma mais pormenorizada.



**Figura 3.39**– Janela de acesso à consulta da base de dados.

Após a selecção abre-se a janela da principal da base de dados com se ilustra na figura 3.40.

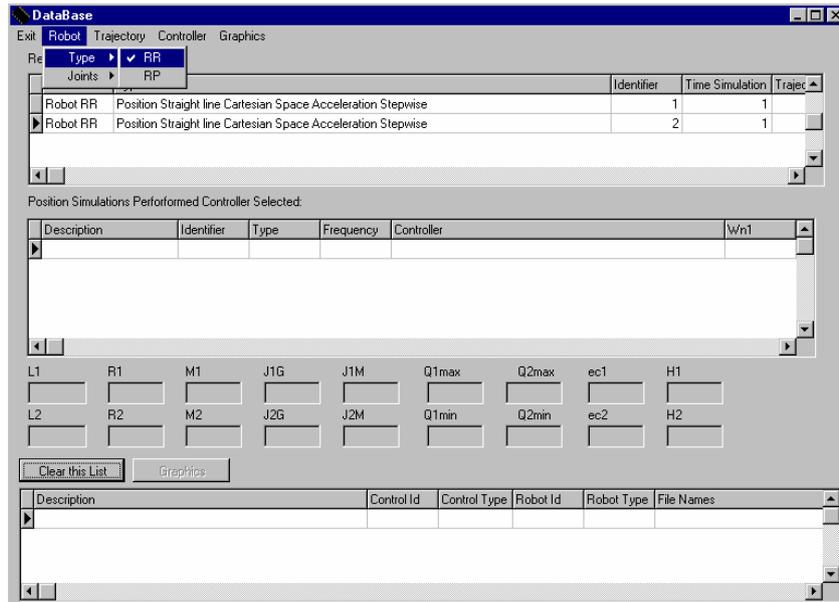


**Figura 3.40** – Janela principal da base de dados.

Na janela da base de dados (figura 3.40) as opções existentes permitem fazer pesquisas das trajetórias planeadas e saber quais foram os robôs, controladores e tipos de fenómenos nas juntas dos robots que foram simulados anteriormente para cada trajetória.

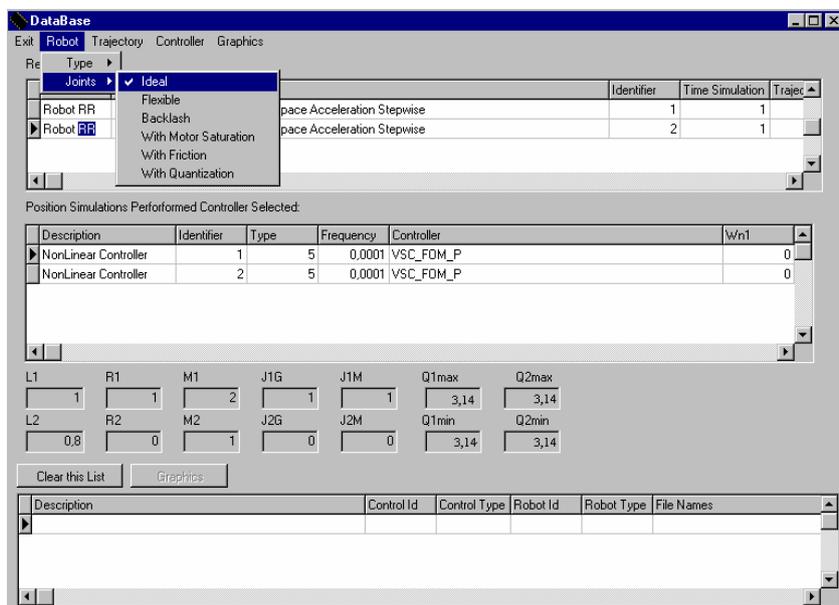
O simulador apresenta automaticamente, nas tabelas da figura 3.40, uma trajetória seleccionada, assim como um tipo de robô, um tipo de junta, um algoritmo de controlo e uma das variáveis a visualizar no gráfico.

Esta janela mostra os planeamentos na tabela 1 (*i.e. reference*, tabela superior na figura 3.40) e as simulações na tabela 2 (*i.e. Position Simulations Performed Controller Selected* tabela do meio na figura 3.40). Na tabela 1 são apresentados todos os dados necessários para saber que trajetória se está a visualizar. Da mesma forma na tabela 2 apresentam-se os dados das respectivas simulações efectuadas para a trajetória seleccionada. Os campos kbytes existentes na tabela 1 e 2 indicam o tamanho de cada ficheiro de planeamento e de simulação; os outros campos indicam as variáveis introduzidas pelo utilizador na respectiva simulação.



**Figura 3.41** – Janela com um Robô RR e uma algoritmo de controlo VSS.

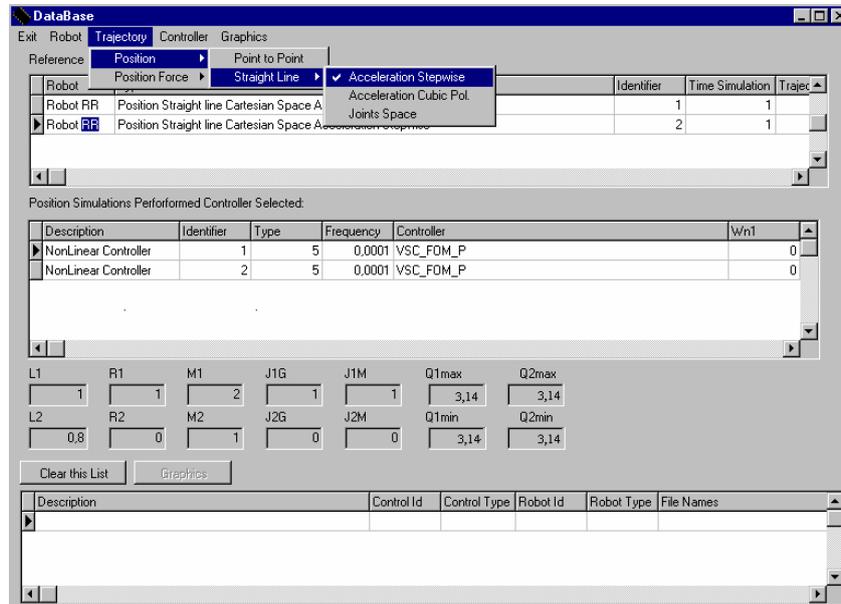
O procedimento para a utilização da base de dados do simulador passa pela selecção da trajectória planeada. Após a selecção, escolhe-se o tipo de robot, o tipo de fenómeno dinâmico nas juntas e o controlador ( figura 3.42).



**Figura 3.42** - Tipo de junta a seleccionar.

Após a selecção da trajectória, que se apresenta na tabela 1 (*Reference*), faz-se a ligação das simulações existentes à trajectória escolhida. Aqui, o utilizador deve

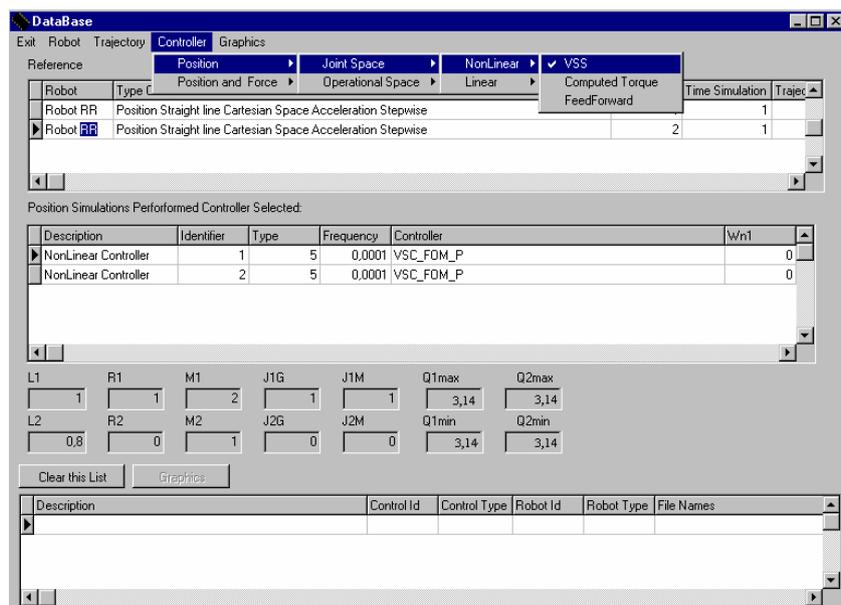
seleccionar uma das trajectórias apresentadas na tabela 1. Essa escolha deverá ser feita através do botão esquerdo do rato como é ilustrado na figura 3.43.



**Figura 3.43**– Selecção do tipo de trajectória.

As simulações existentes para o controlador seleccionado aparecem na tabela 2. Deste modo, para qualquer alteração da estratégia de controlo, aparecem na tabela 2 as simulações do novo algoritmo de controlo seleccionado.

A selecção do tipo de controlador de posição é ilustrado na figura 3.44 onde, a título de exemplo, se apresenta um controlador não linear de estrutura variável.



**Figura 3.44** - Controlador não linear de estrutura variável.

Se a opção do controlador for do tipo baseado em modelos, o modo de apresentação do modelo do robô simulado para cada trajectória é feito com um duplo *clik* no botão direito do rato na simulação seleccionada (figura 3.45), apresentando noutra janela o modelo do robô, como se observa na figura 3.46.

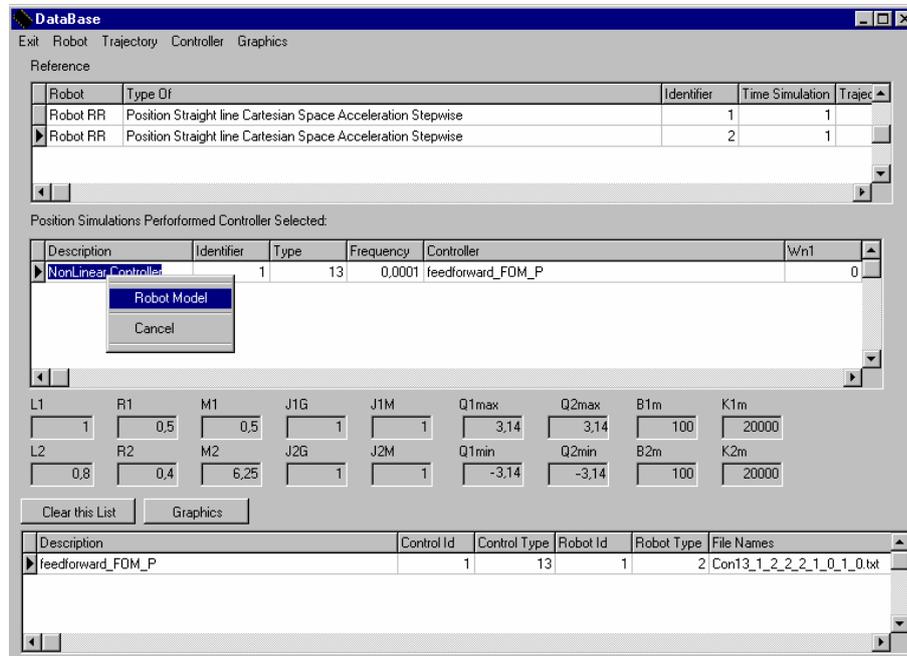


Figura 3.45 - Opção do controlador do tipo baseado em modelos.

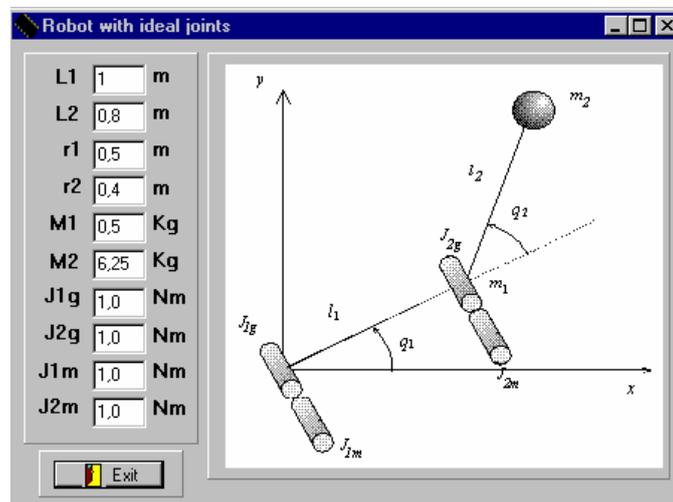
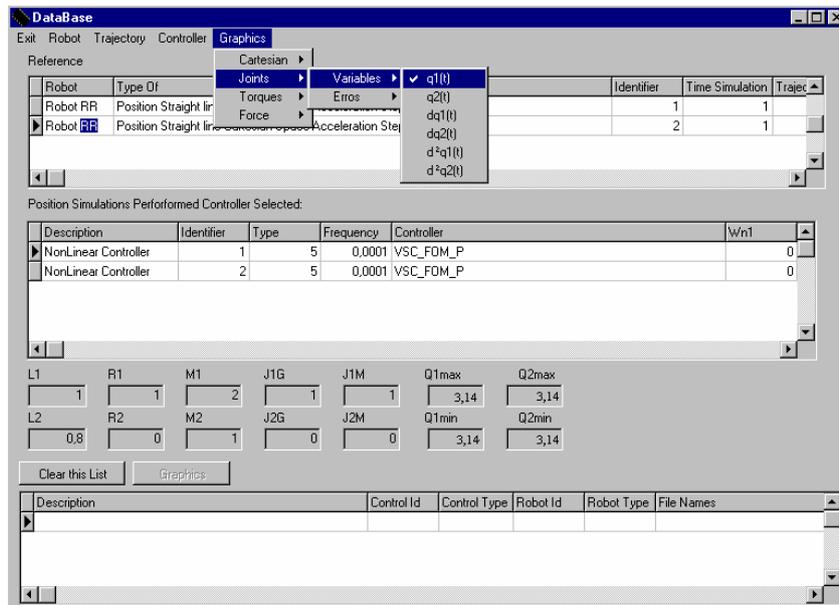


Figura 3.46 – Modelo do robô para a trajectória seleccionada.

Para visualizar graficamente uma ou mais simulações selecciona-se a lista das simulações apresentadas na tabela 2. Uma vez seleccionada uma determinada simulação esta passa a estar incluída na tabela 3. A tabela 3 lista as simulações que vão sendo

seleccionadas a partir da tabela 2. Desta forma pode-se consultar e visualizar qualquer simulação armazenada na base de dados, estabelecendo comparações entre simulações, com diversos tipos de algoritmos e os diversos tipos de fenómenos nas juntas do robot.

A escolha do tipo de variável a visualizar, assim como dos erros entre a trajectória de referência e a resultante da simulação podem observar-se na figura 3.47.



**Figura 3.47** - Tipo de variável a visualizar gráficamente.

Quando o controlador em análise é de posição/força apresenta-se uma nova tabela, paralela à tabela 2, como se mostra na figura 3.48. Assim, os controladores de posição listam-se na tabela 2 e os de força na tabela 3. Pode-se saber, se o controlador seleccionado é só de posição, só de força ou tem ambas as componentes, através de um campo na tabela 1 (*Trajectory* que indica a acção de controlo).

O modo de visualizar as simulações e de as estudar passa por aceder aos dados da tabela 3, da mesma forma que, anteriormente, se descreveu para aceder aos controladores de posição na tabela 2.

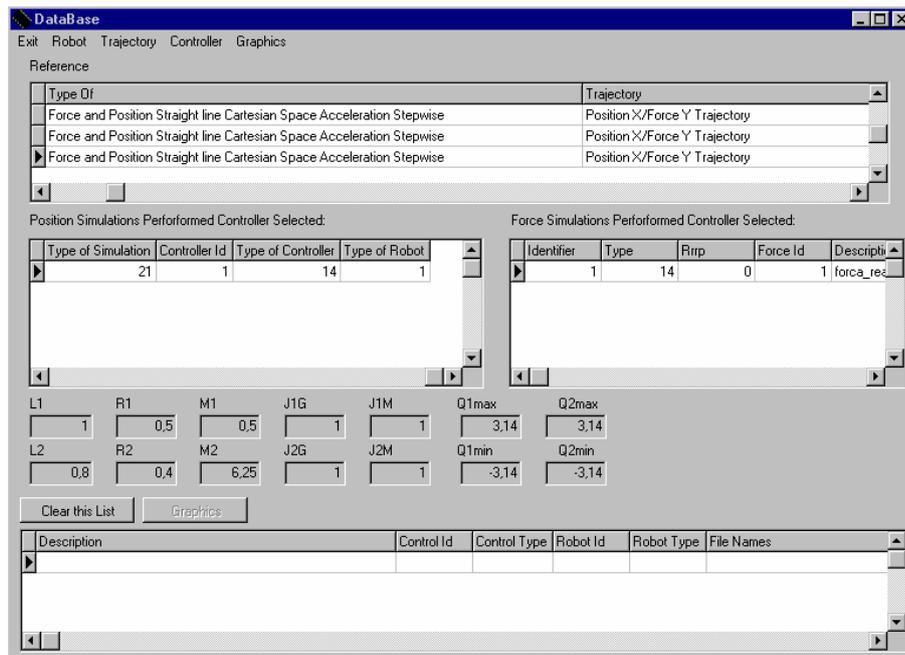


Figura 3.48 – Controladores de posição/força.

### 3.9 Gráficos do simulador

O simulador RobLib permite efectuar gráficos das diversas variáveis das trajectórias simuladas. Gráficos esses que correspondem à evolução temporal nas variáveis das juntas, da trajectória no espaço operacional, da trajectória no espaço das juntas e dos erros em relação à trajectória de referência.

O modo de visualização dos gráficos pode ser feito a partir do menu principal, (figura 3.49), mas também existe a possibilidade de ser executado com os menus dos controladores ou através da base de dados.

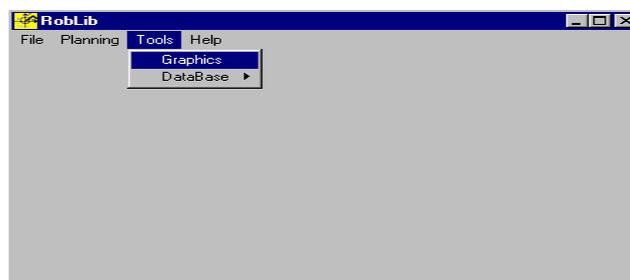
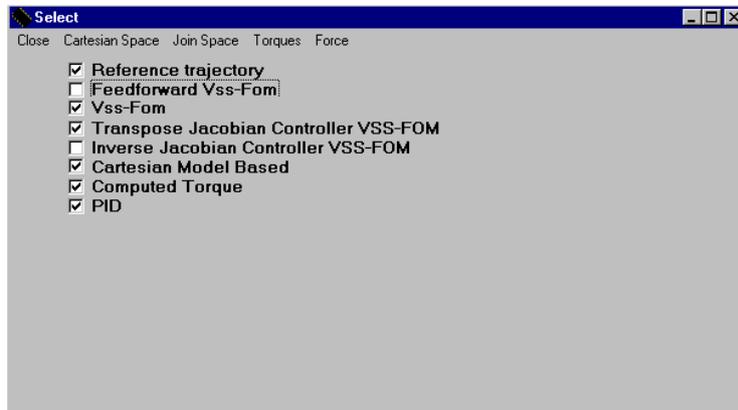


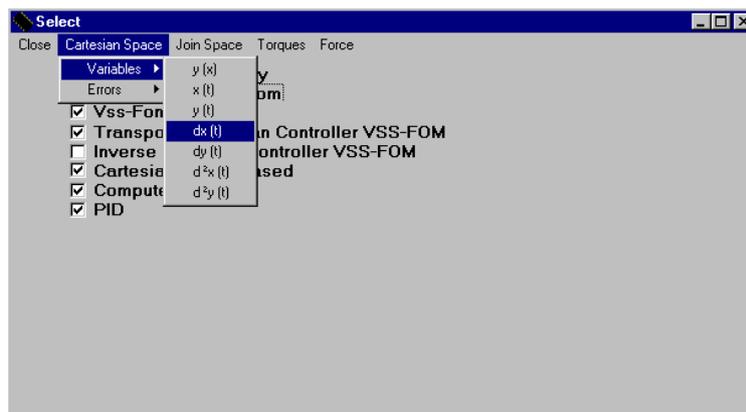
Figura 3.49 –Janela de acesso aos gráficos.

Para ilustrar as funcionalidades da função “Graphics” apresenta-se na figura 3.50 um exemplo onde se executaram simulações de vários algoritmos para uma trajectória

teste. Após cada simulação a janela actualiza-se “dinamicamente”, inserindo os nomes das simulações já executadas. As simulações e as variáveis aqui a analisar (figura 3.47), podem ser visualizadas graficamente, uma vez seleccionadas no menu adequado.

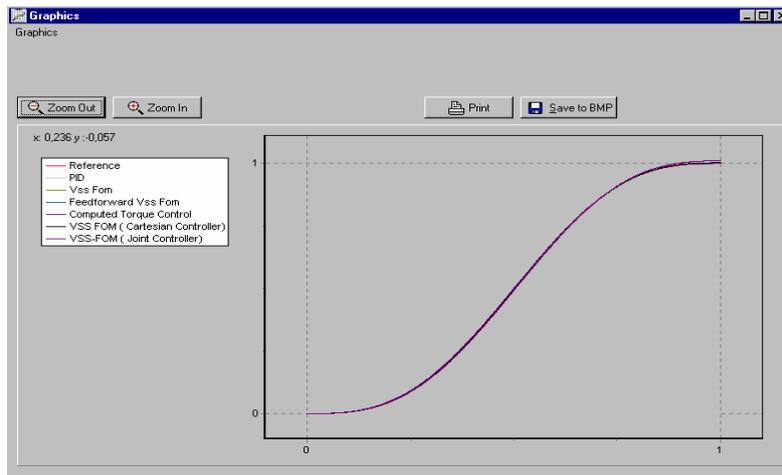


**Figura 3.50** – Janela de selecção dos gráficos.

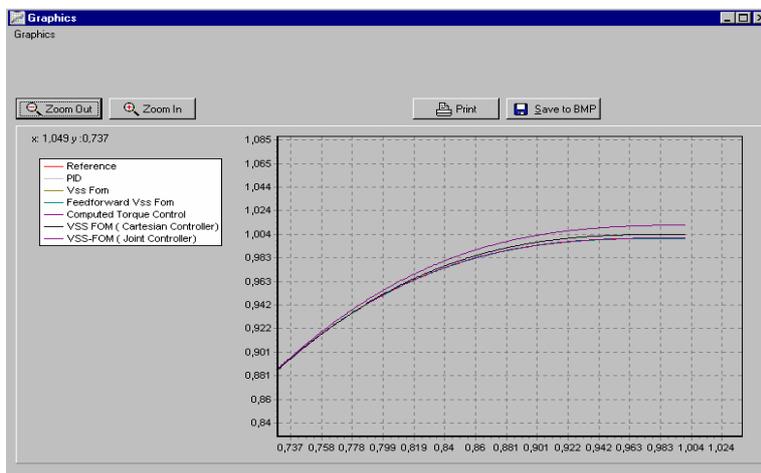


**Figura 3.51** – Janela de selecção da variável a observar.

Na figura 3.52 apresentam-se os gráficos, relativos às posições segundo o eixo  $x$ , das diversas simulações seleccionadas no menu anterior, assim como da trajectória de referência. O botão *Print* permite imprimir a janela correspondente aos gráficos da evolução temporal das variáveis e o botão *Save* abre uma janela de gravação (de ficheiros do tipo *\*.BMP*) correspondente aos gráficos visualizados. Os botões *Zoom in* e *Zoom out* permitem ampliar e diminuir o tamanho do gráfico. Existe também a possibilidade de aumentar ou diminuir uma área específica do gráfico, utilizando o rato como se ilustra na figura 3.53. Desta forma, o utilizador pode ver as diferentes evoluções da variável observada para cada simulação.



**Figura 3.52** - Gráfico da evolução temporal da variável x.



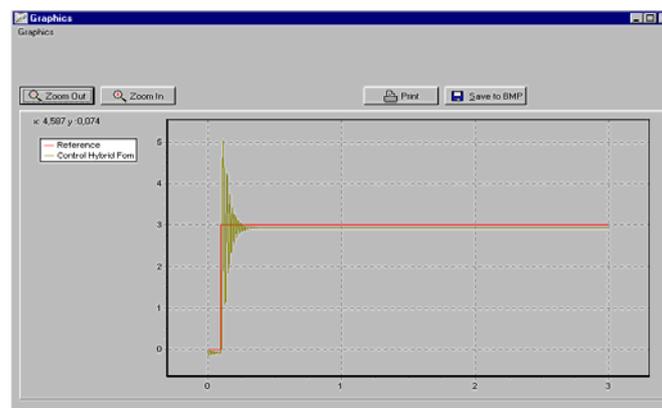
**Figura 3.53** - “Zoom” do gráfico da evolução temporal da variável x.

O modo de apresentação dos gráficos é idêntico para qualquer tipo de controlador, seja ele de posição ou de posição/força. Neste último caso apresentam--se as variáveis de força, que obviamente “não existem”(i.e. não são controlados explicitamente) nos controladores de posição. Dependendo do tipo da acção de controlo, de posição ou de força nos eixos do robô, existem variáveis que “deixam de ter significado” visto que não estão a ser controladas directamente.

Na figura 3.55 pode-se ver a representação gráfica da variável força, da simulação face à planeada do eixo 1 para o robô.



**Figura 3.54** - Selecção da simulação dos controladores de força.



**Figura 3.55** - Simulação da variável força do controlador híbrido.

### 3.10 Conclusões

Neste capítulo descreveram-se as funcionalidades do programa *RobLib*. Este programa permite a simulação de robôs de duas juntas com diferentes tipos de fenómenos dinâmicos nas juntas assim como um conjunto de algoritmos de controlo de posição, de velocidade e de força.

No *RobLib* podem-se parametrizar os robôs, visualizar a sua região de trabalho e a área de manipulação, por outro lado, o programa permite efectuar um planeamento rectilíneo quer no espaço cartesiano quer no espaço das juntas.

De seguida, apresentou-se a base de dados integrada no simulador e descreveram-se as suas funcionalidades. Assim, é possível armazenar as simulações efectuadas e comparar os diferentes algoritmos de controlo.

Além disso, o *RobLib* permite visualizar inúmeras variáveis tais como a posição, a velocidade, a aceleração, os erros de seguimento da trajectória em ambos os eixos, as forças e os binários a aplicar nos actuadores.

Deve ainda referir-se que os dados armazenados resultantes das simulações são guardados em ficheiros, o que possibilita o seu tratamento e análise através de outros programas.

## CAPITULO 4

### Análise Dinâmica

e

### Estudo de Algoritmos de Controlo

#### 4.1 Introdução

Este capítulo é dedicado à apreciação do desempenho dos controladores estudados, perante um conjunto de testes efectuados com o simulador *RobLib*. Iremos observar os resultados apresentados por cada um dos controladores e compará-los entre si. Os sinais em análise são :

- os binários nos eixos do robô e as forças aplicadas numa superfície de contacto;
- as posições, as velocidades e as acelerações no espaço das juntas e espaço cartesiano;
- os erros relativamente às posições, velocidades, acelerações, binários e forças.

Nesta perspectiva, são apresentadas várias simulações para a mesma trajectória, com vista à análise da dinâmica dos robôs, e do desempenho dos algoritmos de controlo. Assim, o sistema é simulado através do método de integração numérica de Runge-Kutta de ordem quatro a uma frequência de integração  $f_{sim}= 100\text{kHz}$ . Por outro lado, os controladores adoptam uma frequência de amostragem  $f_c= 10\text{kHz}$ .

Os erros foram encontrados pela diferença entre o sinal simulado e o sinal de referência. A medida do erro de seguimento adoptado é dado pela seguinte equação:

$$e_{pi} = \sqrt{(x_{di} - x_{ii})^2 + (y_{di} - y_{ii})^2} \quad i=1,2 \quad (125)$$

onde  $e_{pi}$  é o erro de seguimento,  $x_d$  e  $y_d$  são as posições desejadas,  $x_i$  e  $y_i$  as posições actuais da simulação para a junta  $i=1,2$ .

Neste capítulo apresentam-se somente algumas das simulações efectuadas visto ser “impossível” abranger todo tipo de situações. Desta forma, as simulações apresentadas estão divididas em seis conjuntos.

O primeiro conjunto de experiências é efectuado para o robô com estrutura RR ideal. As simulações permitem fazer um estudo comparativo dos algoritmos de controlo de posição, assim como, entre as estratégias de controlo no espaço das juntas e no espaço cartesiano.

O segundo grupo de simulações testa os diferentes tipos de planeamentos de trajectórias, tanto no espaço cartesiano como no espaço das juntas (para o mesmo ponto de partida e de chegada), assim como diferentes tipos de evolução da aceleração no tempo.

O terceiro conjunto estabelece uma comparação dos algoritmos baseados em modelos. Desta forma, testam-se os algoritmos de ‘acção antecipativa’ e do ‘binário calculado’ face à estimação inexacta de parâmetros do modelo dinâmico do robô.

O quarto conjunto de experiências analisa os fenómenos dinâmicos das juntas. Deste modo, pode apreciar-se o comportamento de cada fenómeno dinâmico e observar o seu efeito face ao caso ideal.

O quinto conjunto de simulações é feito para um robô com estrutura RP ideal do modo análogo ao efectuado anteriormente para o robô RR.

O último grupo apresenta diversas simulações do algoritmo CH de modo a analisar os efeitos dos seus subcontroladores no controlo de:

- somente de posição;
- somente de força;
- posição numa direcção e força noutra direcção.

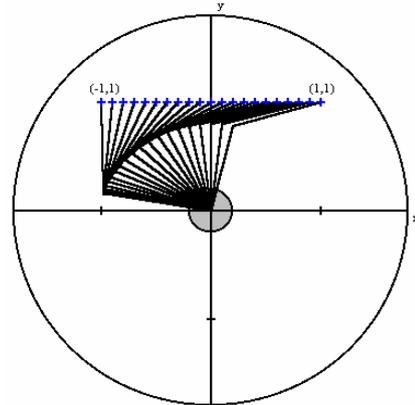
Nestes grupos são também estudados os efeitos dinâmicos da superfície de restrição.

## **4.2 Simulação de Algoritmos de Controlo**

As simulações apresentadas são efectuadas para a mesma referência com vista a comparar o desempenho dos algoritmos de controlo. Para o efeito planeou-se uma

trajectória (figura 4.1) segundo uma recta com evolução temporal de aceleração/desaceleração em degrau e com os seguintes parâmetros:

- posição inicial (-1,1);
- posição final (1,1);
- tempo de simulação 4 segundos;
- tempo de trajectória 2 segundos.

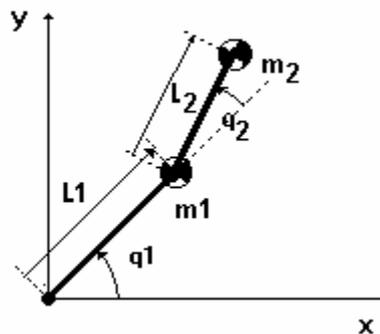


**Figura 4.1** Trajectória de referência para o robô RR.

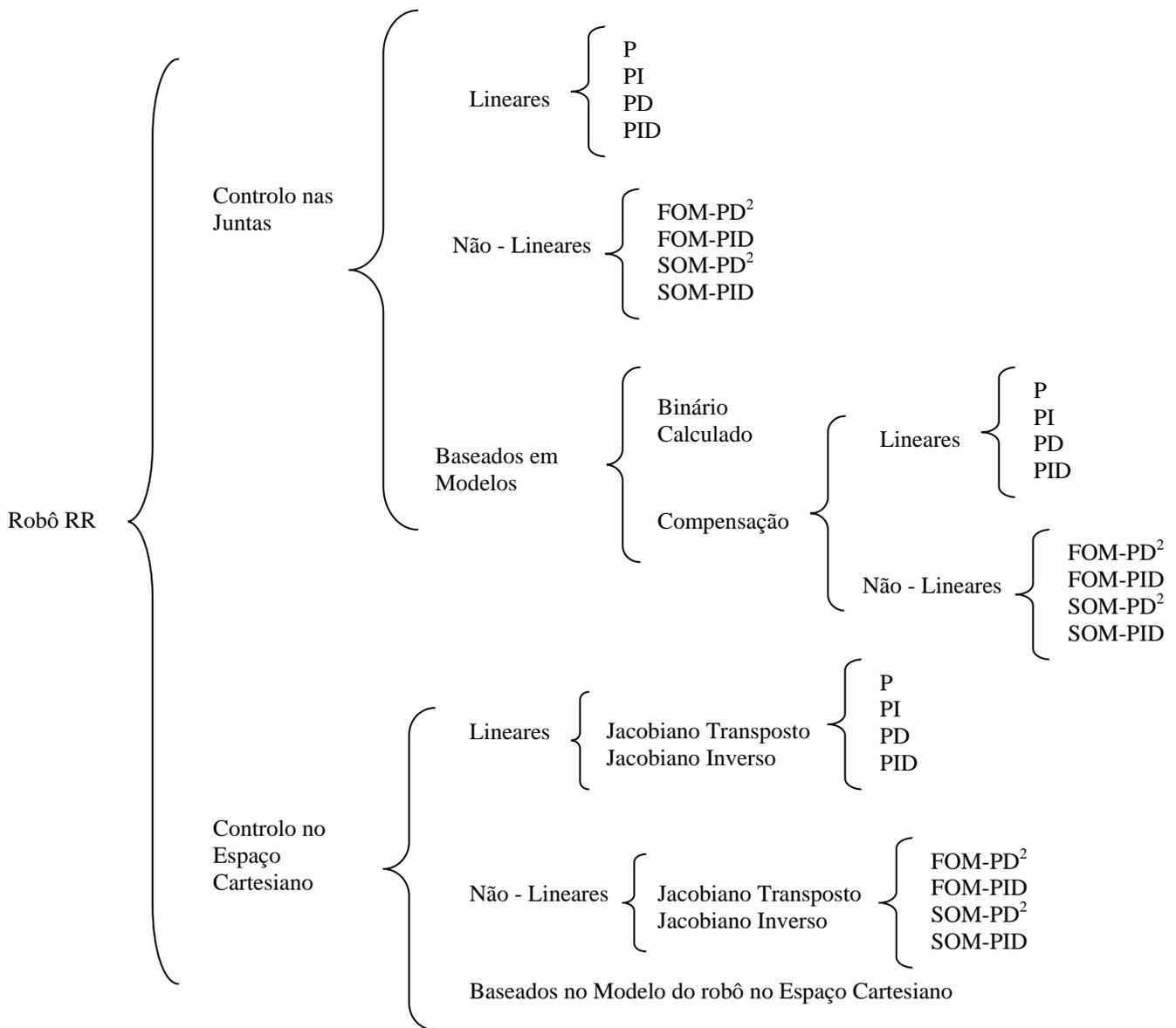
O robô simulado apresenta os parâmetros indicados na tabela 4.1:

**Tabela 4.1** – Parâmetros do robô RR com juntas ideais.

$i$	$m_i(\text{kg})$	$l_i(\text{m})$	$J_{im}(\text{kgm}^2)$	$J_{ig}(\text{kgm}^2)$
1	0.5	1.0	1.0	1.0
2	6.25	0.8	1.0	1.0



**Figura 4.2** – Robô RR com juntas ideais.



**Figura 4.3** – Esquema das simulações para os algoritmos existentes no *RobLib*.

Na figura 4.3 apresenta-se um esquema das possíveis simulações de controlo de posição que se podem executar no simulador *RobLib*. Para não ser demasiado extensa a apresentação de todas as simulações, assim como a visualização das variáveis dos diversos algoritmos, apresenta-se apenas um subconjunto destas. No entanto, se necessário, pode observar-se no *RobLib* todas as variáveis, de posição, de velocidade e de aceleração (em ambos os espaços de trabalho), assim como os binários aplicados nos actuadores do robô, a evolução temporal dos erros de todas as variáveis, a trajectória no espaço e o seu erro de seguimento.

Considera-se um coeficiente de amortecimento crítico porque se pretende um controlo do robot sem oscilações. Deste modo, optou-se pelo valor  $\xi = 1$  na análise dos algoritmos. Na escolha da frequência natural do robot consideraram-se dois casos:  $f_n = 100\text{Hz}$  (i.e.  $\omega_n = 628,3\text{rads}^{-1}$ ) e  $f_n = 10\text{Hz}$  (i.e.  $\omega_n = 62,8\text{rads}^{-1}$ ).

### 4.3 Controlo nas juntas

#### 4.3.1 Algoritmos de controlo linear

Nas figuras 4.4 até 4.10 apresentam-se diferentes simulações de controladores lineares, tais como o controlador proporcional, proporcional integral, proporcional derivativo e o controlador PID. Desta forma, apresentam-se separadamente as diversas acções de controlo, de modo a compreender os seus efeitos.

Os parâmetros da simulação dos controladores lineares, com  $f_n = 10\text{Hz}$  e  $f_n = 100\text{Hz}$ , apresentam-se, respectivamente nas tabelas 4.2 e 4.3.

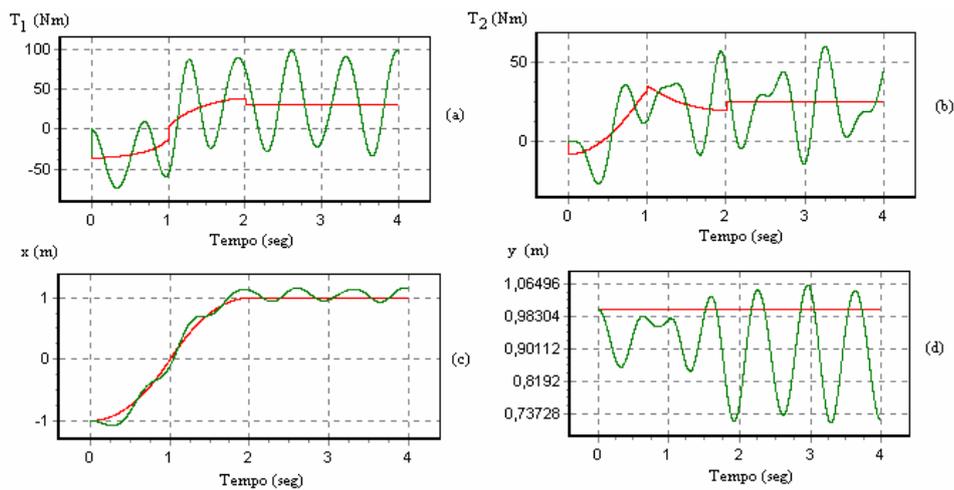
Considerando o robô com os parâmetros da tabela 4.1, atribuem-se os valores da frequência natural do sistema para simular um sistema lento e um sistema rápido.

**Tabela 4.2** – Parâmetros dos controladores lineares para  $f_n = 10\text{Hz}$ .

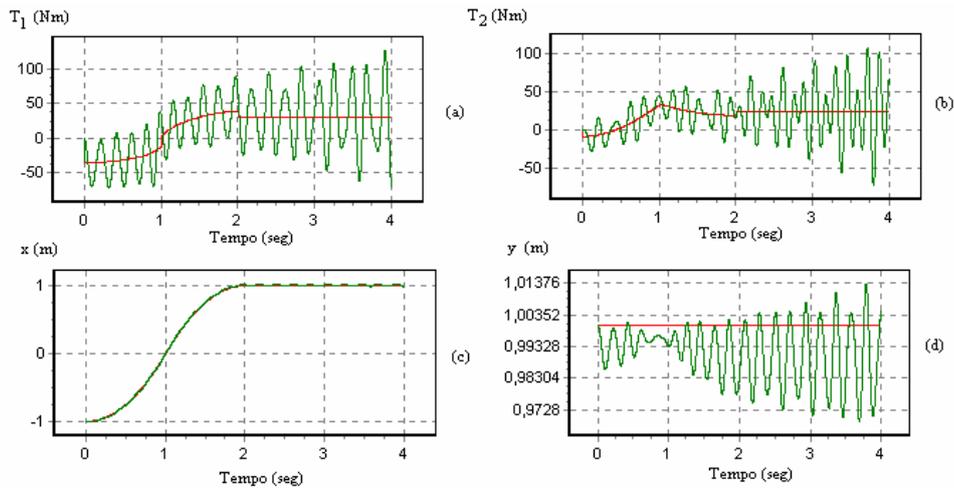
<i>Controlador</i>	<i>Junta i</i>	$Kp_i$	$Ki_i$	$Kd_i$
P	1	500		
	2	500		
PI	1	12566,37	394784,17	
	2	12566,37	394784,17	
PD	1	3947,84		125,66
	2	3947,84		125,66
PID	1	12566,37	394784,17	100
	2	12566,37	394784,17	100

**Tabela 4.3** - Parâmetros dos controladores lineares para  $f_n= 100\text{Hz}$ .

Controlador	Junta $i$	$Kp_i$	$Ki_i$	$Kd_i$
P	1	5000		
	2	5000		
PI	1	125663,7	39478417,6	
	2	125663,7	39478417,6	
PD	1	394784,12		1256,6
	2	394784,12		1256,6
PID	1	125663,7	39478417,6	100
	2	125663,7	39478417,6	100

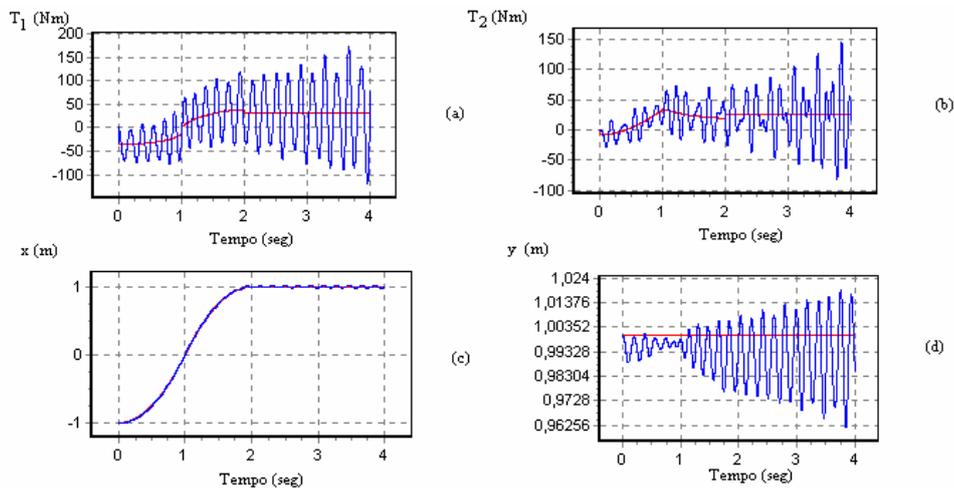


**Figura 4.4** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador proporcional com os parâmetros da tabela 4.2 (*i.e.*  $f_n=10\text{Hz}$ ), considerando a vermelha a trajectória planeada e verde a simulação.



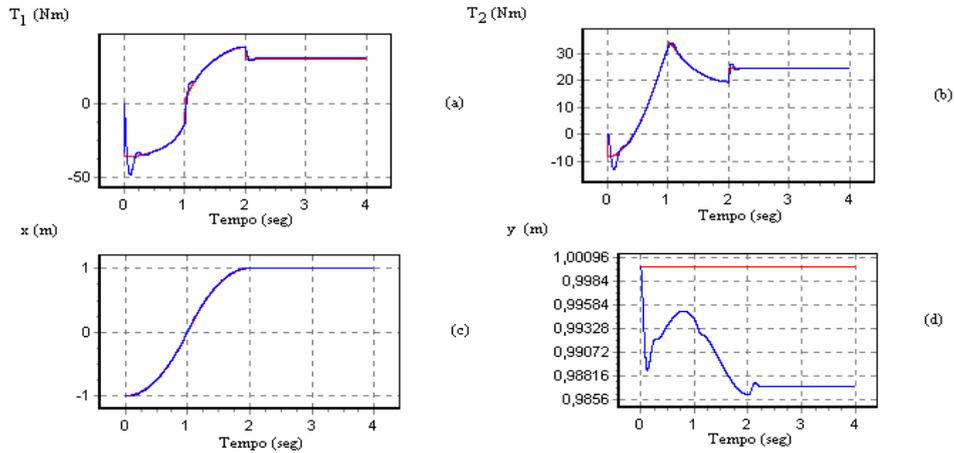
**Figura 4.5** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador proporcional com os parâmetros da tabela 4.3 (*i.e.*  $f_n=100\text{Hz}$ ), considerando a vermelha a trajectória planeada e verde a simulação.

Para o controlador proporcional pode-se observar que o erro de seguimento da trajectória diminui com o aumento da frequência natural do sistema. Em contrapartida, aumenta substancialmente os binários dos actuadores.

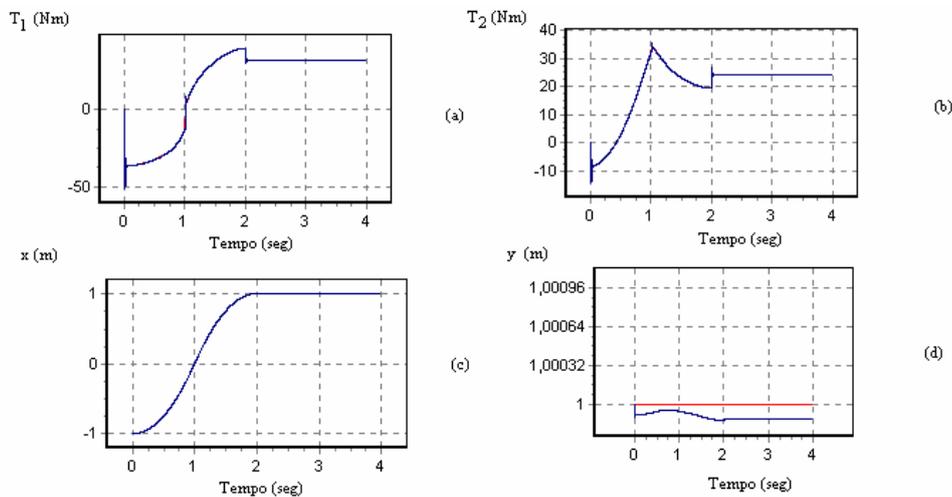


**Figura 4.6** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador proporcional integral com os parâmetros da tabela 4.2 (*i.e.*  $f_n=10\text{Hz}$ ), considerando a vermelha a trajectória planeada e azul a simulação.

Quando se introduz uma acção integral no controlador proporcional pretende-se diminuir o erro em regime permanente. Desta forma, comparando a figura 4.4 com a figura 4.6, para  $f_n=10\text{Hz}$ , nota-se a diminuição do erro mas um aumento elevado do binário nas juntas.

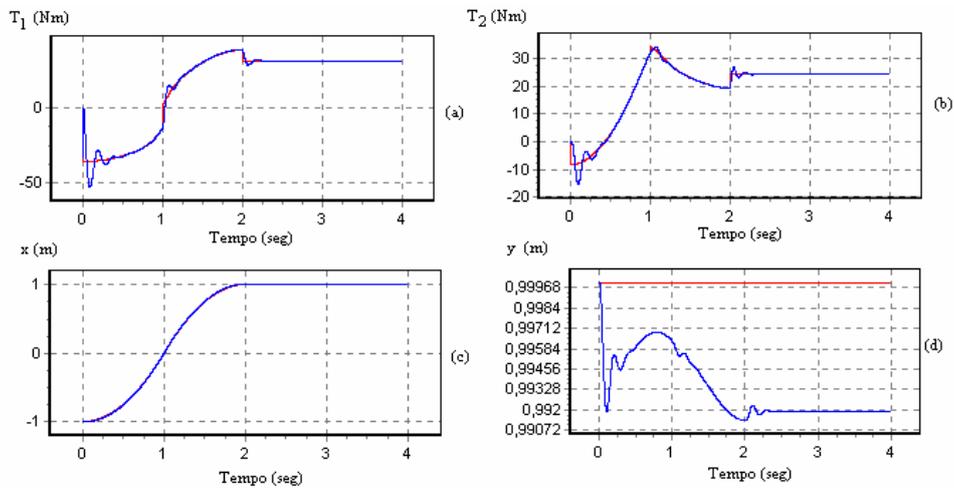


**Figura 4.7** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador proporcional derivativo com os parâmetros da tabela 4.2 (*i.e.*  $f_n=10\text{Hz}$ ), considerando a vermelho a trajectória planeada e azul a simulação.

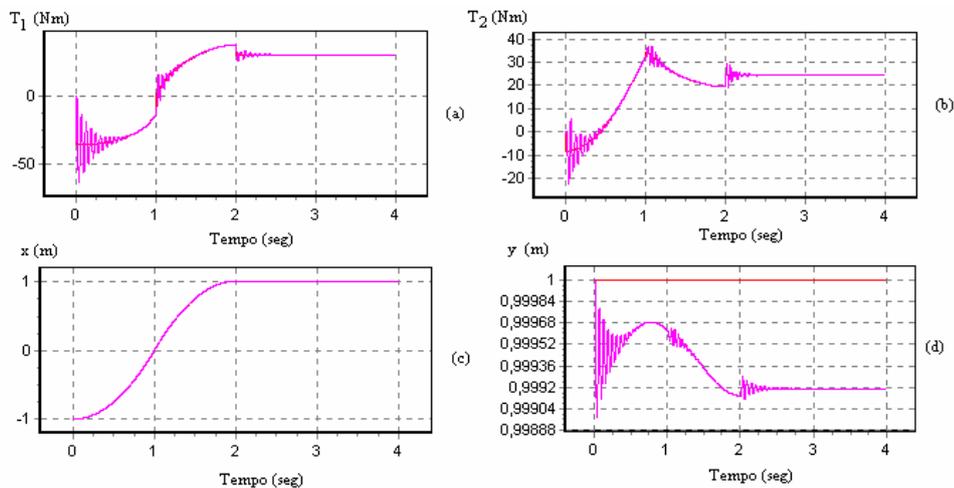


**Figura 4.8** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador proporcional derivativo com os parâmetros da tabela 4.3 (*i.e.*  $f_n=100\text{Hz}$ ), considerando a vermelho a trajectória planeada e azul a simulação.

Por outro lado, verifica-se que o controlo do robô com acção derivativa (figuras 4.7 e 4.8) apresenta um erro em regime permanente bastante menor ao encontrado com o controlador P ou PI (figuras 4.4 a 4.7).



**Figura 4.9** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador proporcional integral derivativo com os parâmetros da tabela 4.2 (*i.e.*  $f_n=10\text{Hz}$ ), considerando a vermelha a trajectória planeada e azul a simulação.



**Figura 4.10** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador proporcional integral derivativo com os parâmetros da tabela 4.3 (*i.e.*  $f_n=100\text{Hz}$ ), considerando a vermelha a trajectória planeada e cor de rosa a simulação.

Apesar da complexidade em sintonizar os controladores, por comparação, verifica-se que a introdução do termo derivativo vai ser responsável pela resposta a pequenas variações de erro, diminuindo a sobre-elevação (*overshoot*) e o tempo de estabelecimento.

### 4.3.2 Algoritmos de controlo não lineares

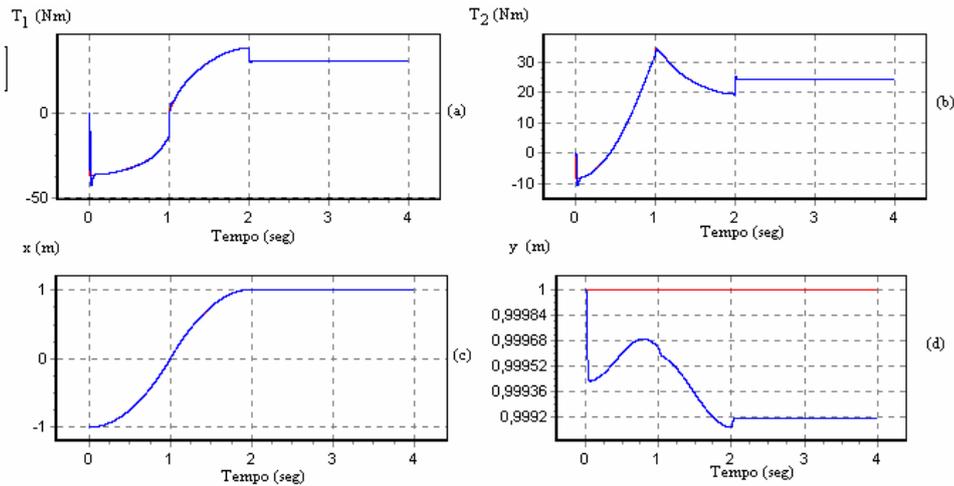
Para os controladores não-lineares do tipo VSS efectuaram-se simulações com  $f_n=10\text{Hz}$  e  $f_n=100\text{Hz}$ , com os ganhos apresentados nas tabelas 4.4 e 4.5, respectivamente. Através das seguintes simulações, pretende-se testar os algoritmos não lineares e estabelecer uma comparação entre eles.

**Tabela 4.4** - Parâmetros dos controladores não lineares para  $f_n= 10\text{Hz}$ .

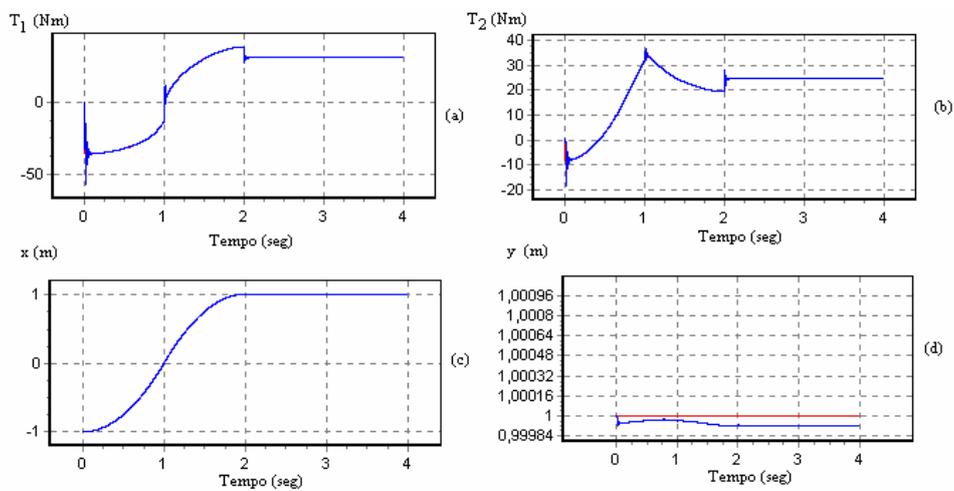
<i>Controlador</i>	<i>Junta i</i>	$CI_i$	$C2_i$	$D_i$	$d_i$
VSS-FOM-PDD <sup>2</sup>	1	62,8	62,8	100	100000
	2	62,8	62,8	100	100000
VSS-FOM-PID	1	62,8	62,8	100	1000000
	2	62,8	62,8	100	1000000
VSS-SOM-PDD <sup>2</sup>	1	62,8	62,8	10	40
	2	62,8	62,8	10	40
VSS-SOM-PID	1	62,8	62,8	10	1000
	2	62,8	62,8	10	1000

**Tabela 4.5** - Parâmetros dos controladores não lineares para  $f_n= 100\text{Hz}$ .

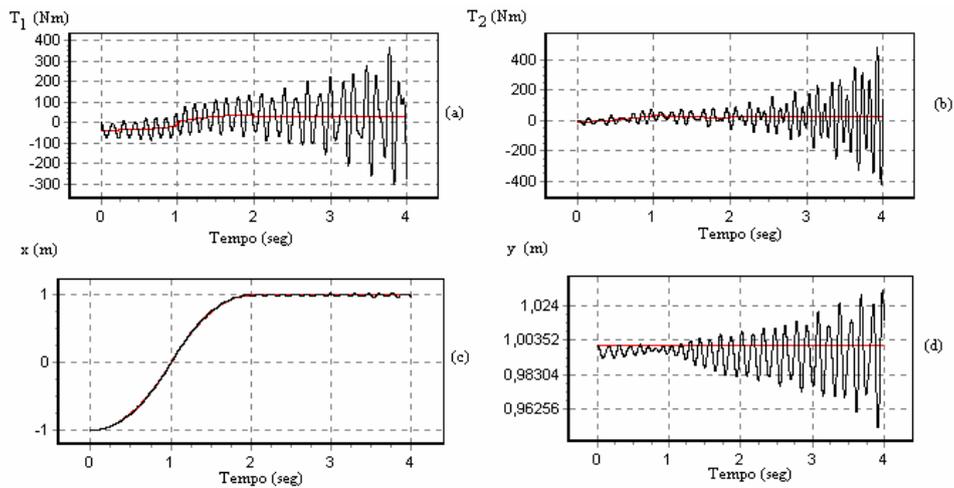
<i>Controlador</i>	<i>Juntas i</i>	$CI_i$	$C2_i$	$D_i$	$d_i$
VSS-FOM-PDD <sup>2</sup>	1	628,3		100	100000
	2	628,3		100	100000
VSS-FOM-PID	1	628,3			1000000
	2	628,3		100	1000000
VSS-SOM-PDD <sup>2</sup>	1	628,3	628,3	10	40
	2	628,3	628,3	10	40
VSS-SOM-PID	1	628,3	628,3	10	1000
	2	628,3	628,3	10	1000



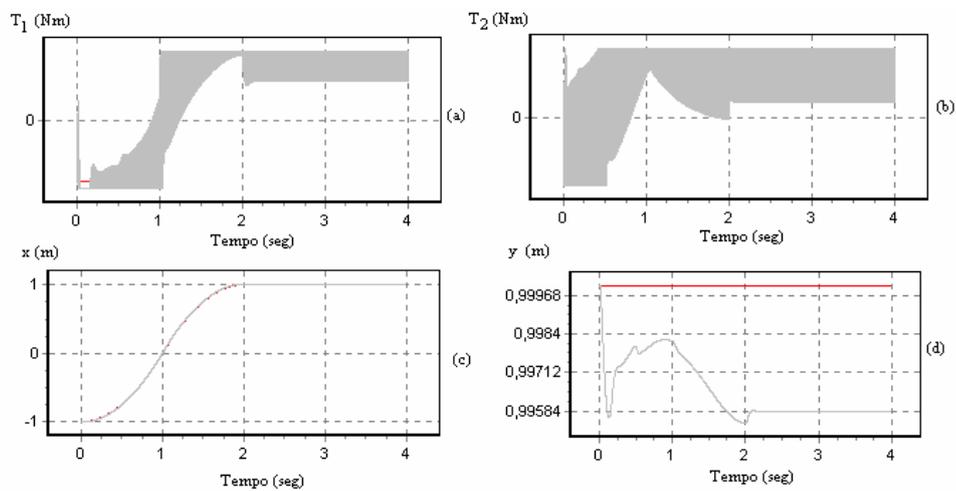
**Figura 4.11** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador VSS-FOM-PDD<sup>2</sup> com os parâmetros da tabela 4.4 (*i.e.*  $f_n=10\text{Hz}$ ), considerando a vermelho o trajectória planeada e a azul a simulação.



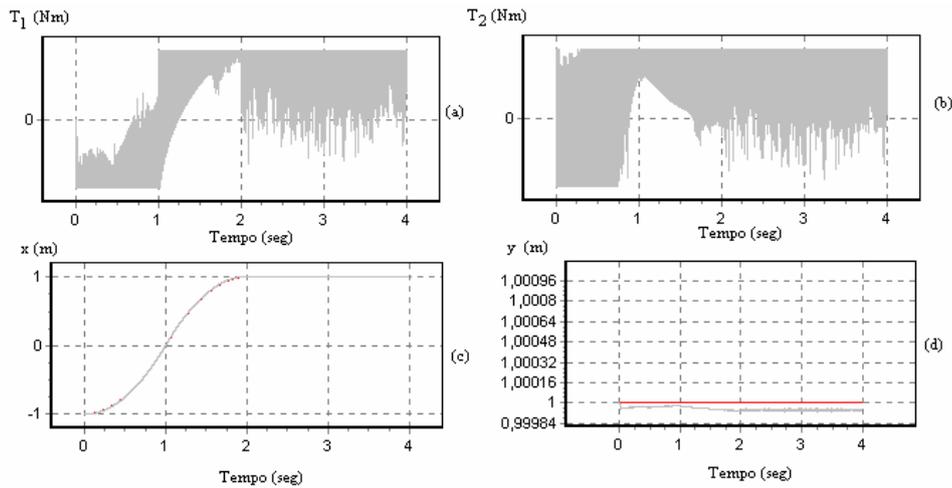
**Figura 4.12** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador VSS-FOM-PDD<sup>2</sup> com os parâmetros da tabela 4.5 (*i.e.*  $f_n=100\text{Hz}$ ), considerando a vermelho a trajectória planeada e azul a simulação.



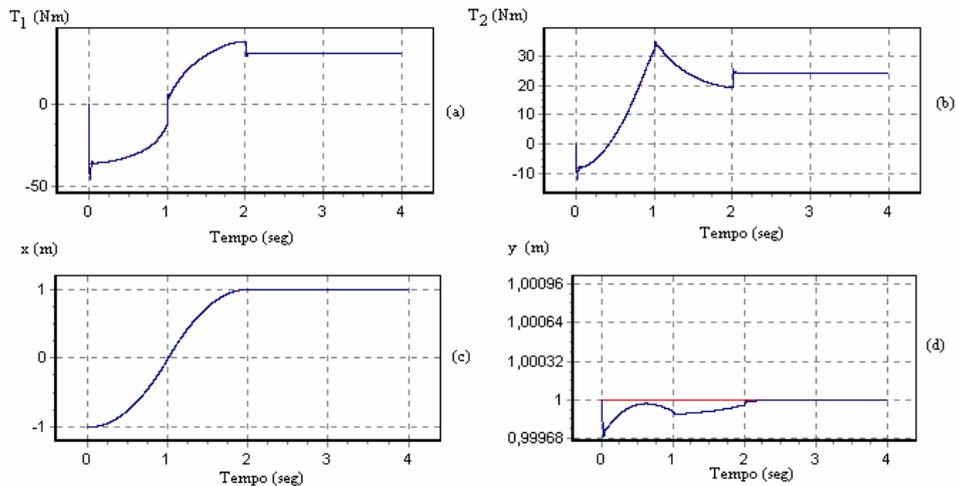
**Figura 4.13** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$  para o controlador VSS-FOM-PID com os parâmetros da tabela 4.4 (*i.e.*  $f_n=10\text{Hz}$ ), considerando a vermelha a trajectória planeada e preto a simulação.



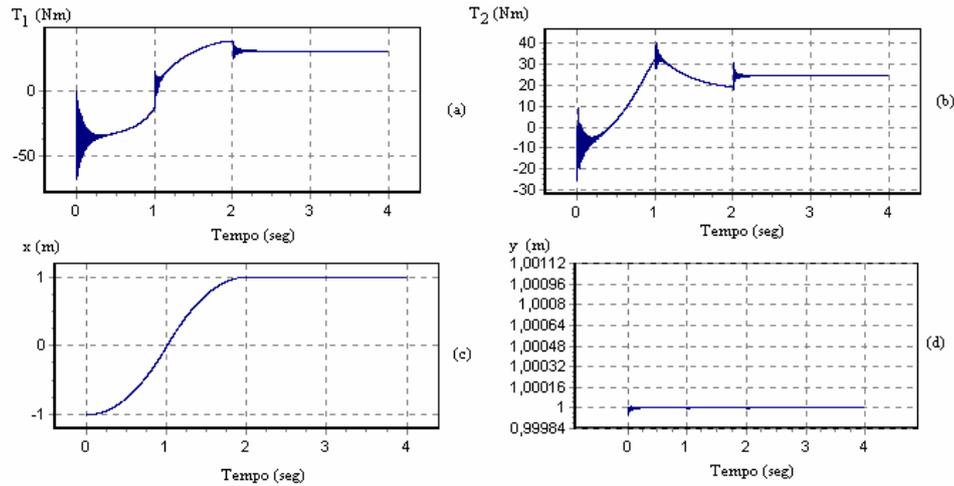
**Figura 4.14** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador VSS-SOM-PDD<sup>2</sup> com os parâmetros da tabela 4.4 (*i.e.*  $f_n=10\text{Hz}$ ), considerando a vermelha a trajectória planeada e cinzento a simulação.



**Figura 4.15** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador VSS-FOM-PDD<sup>2</sup> com os parâmetros da tabela 4.5 (*i.e.*  $f_n=100\text{Hz}$ ), considerando a vermelha a trajectória planeada e cinzento a simulação.



**Figura 4.16** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador VSS-SOM-PID com os parâmetros da tabela 4.4 (*i.e.*  $f_n=10\text{Hz}$ ), considerando a vermelha a trajectória planeada e azul a simulação.



**Figura 4.17** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador VSS-SOM-PID com os parâmetros da tabela 4.5 (*i.e.*  $f_n=100\text{Hz}$ ), considerando a vermelha a trajectória planeada e azul a simulação.

Constata-se que o algoritmo FOM-PID produz bastantes oscilações, para os ganhos apresentados, face aos restantes algoritmos. Observa-se ainda que após efectuada a trajectória se torna difícil de estabilizar o sistema devido à acção integral.

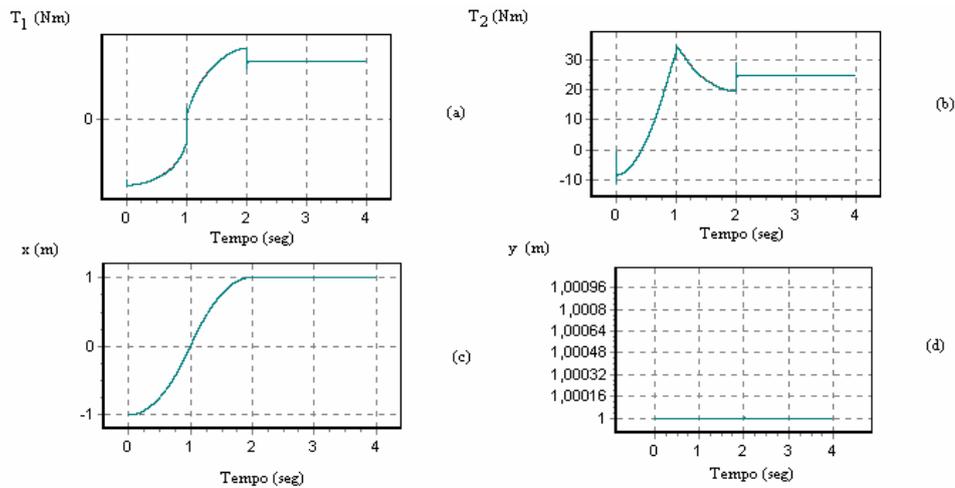
O controlador FOM-PDD<sup>2</sup>, revela-se bastante robusto, mesmo para o caso de um sistema bastante lento ( $f_n=10\text{Hz}$ ).

Observa-se ainda que o algoritmo SOM-PID é o controlador mais robusto e com menor erro de seguimento da trajectória.

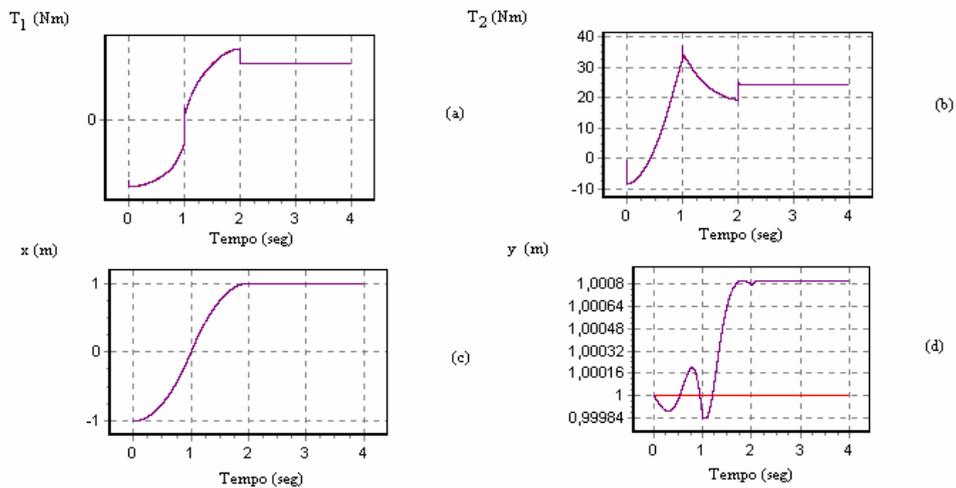
### 4.3.3 Algoritmos de controlo baseados em modelos

As simulações seguintes apresentam o comportamento dos algoritmos baseados em modelos. Desta forma, os controladores pelo método binário calculado (*i.e.* *Computed Torque*) apresenta um ganho de posição de  $\omega_n^2$  e um ganho de velocidade de  $2\xi \omega_n$  para cada um dos actuadores (*i.e.*  $\xi = 1$  e  $\omega_n = 2\pi f_n$ ). Para os sub-controladores do

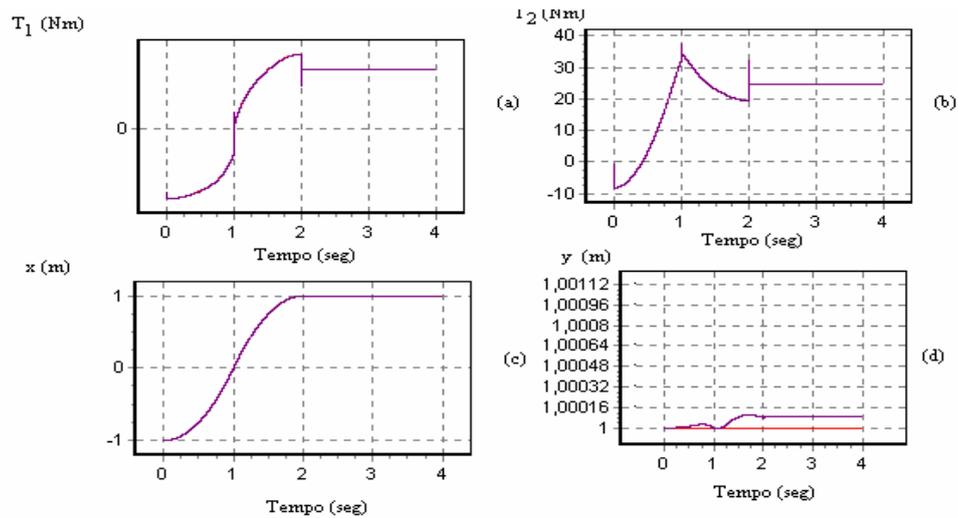
algoritmo com acção antecipativa (*Feedforward*) apresentam-se ganhos idênticos para os mesmos controladores indicados na tabela 4.4 ( $f_n=10\text{Hz}$ ).



**Figura 4.18** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador *Feedforward* com os parâmetros da tabela 4.4 (*i.e.*  $f_n=10\text{Hz}$ ), considerando a vermelho a trajectória planeada e a azul claro a simulação.



**Figura 4.19** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador *Computed Torque* com os parâmetros da tabela 4.2 (*i.e.*  $f_n=10\text{Hz}$ ), considerando a vermelho o trajectória planeada e roxo a simulação.



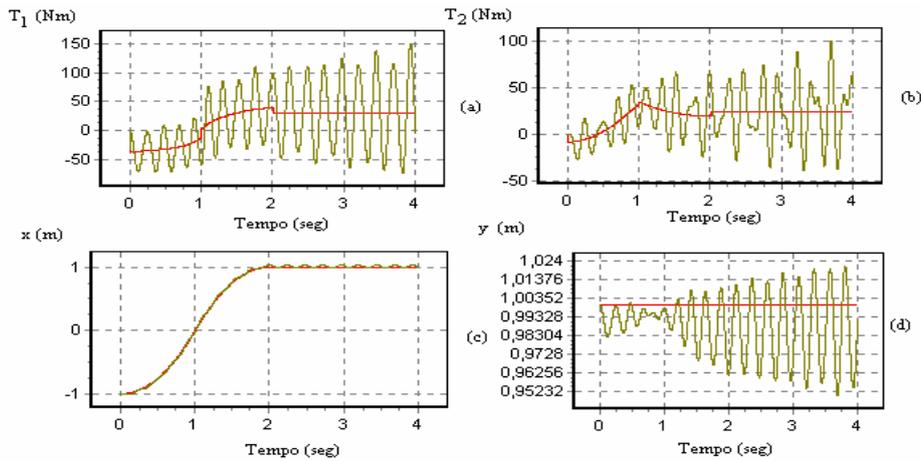
**Figura 4.20** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador *Computed Torque* com os parâmetros da tabela 4.3 (*i.e.*  $f_n=100\text{Hz}$ ), considerando a vermelha a trajetória planeada e cor de rosa a simulação.

Para os algoritmos, baseados em modelos, verifica-se que o desempenho é bastante bom, porque os parâmetros do modelo são iguais ao do robô simulado.

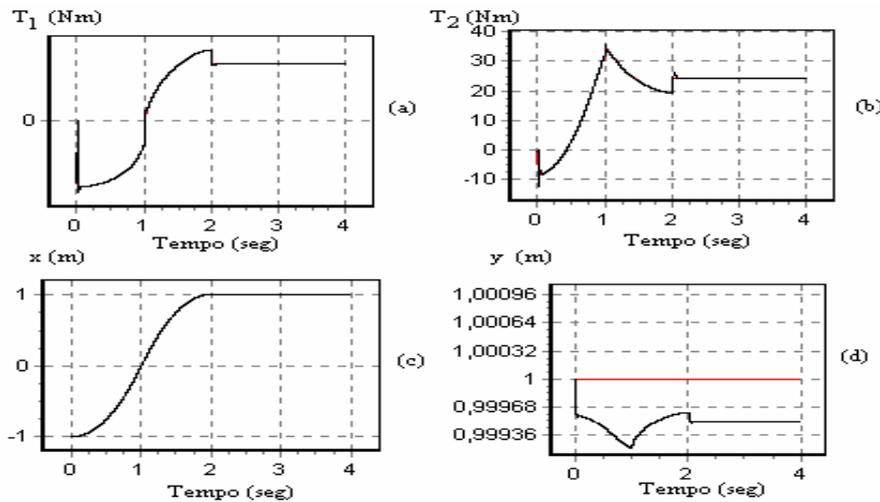
## 4.4 Controladores baseados no espaço cartesiano

### 4.4.1 Jacobiano Transposto e Jacobiano Inverso : Sub-controladores lineares

Os controladores pelos métodos do Jacobiano Transposto e do Jacobiano Inverso, com subcontroladores lineares no espaço cartesiano, apresentam ganhos idênticos aos algoritmos de controlo linear nas juntas (*i.e.* ganhos apresentados na tabela 4.2 com  $f_n=10\text{Hz}$ ).



**Figura 4.21** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador PD pelo método Jacobiano Inverso com os parâmetros da tabela 4.3 (*i.e.*  $f_n=10\text{Hz}$ ), considerando a vermelho a trajectória planeada e a verde claro a simulação.

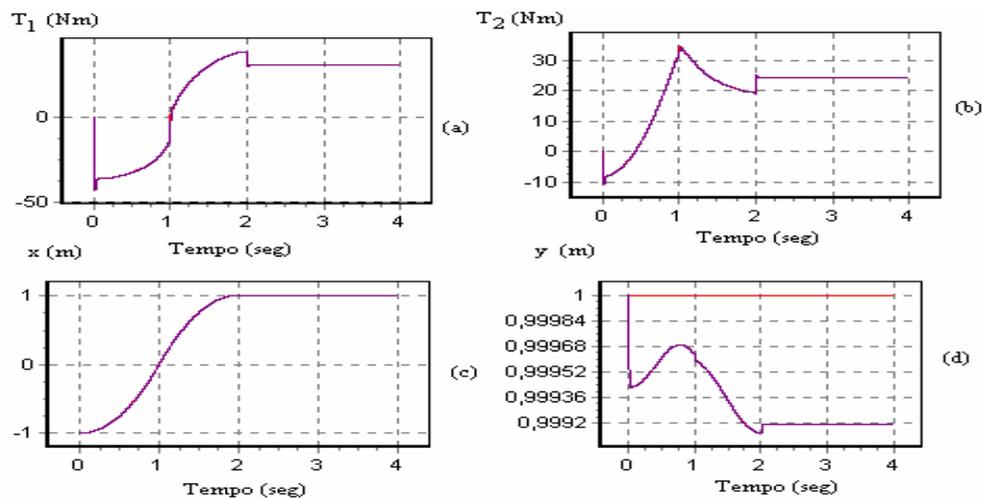


**Figura 4.22** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador PD pelo método Jacobiano Transposto com os parâmetros da tabela 4.3 (*i.e.*  $f_n=10\text{Hz}$ ), considerando a vermelho a trajectória planeada e preto a simulação.

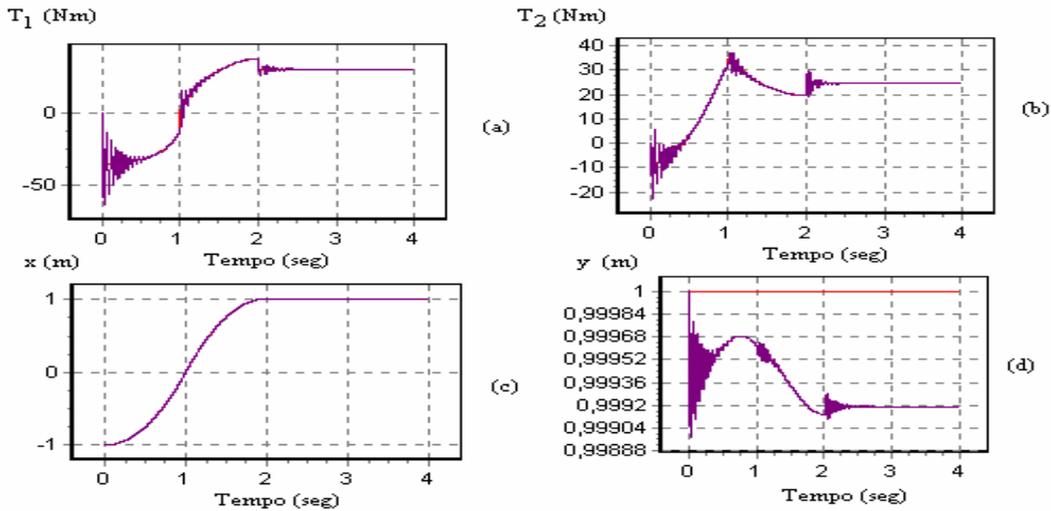
Observando os gráficos relativos aos controladores lineares do tipo PD, seja quando o controlo é efectuado nas juntas, seja quando é no espaço cartesiano, constata-se que o método Jacobiano Inverso é mais instável que os restantes e que o método do Jacobiano Transposto apresenta um menor erro no seguimento da trajectória. Todavia, este método requer mais cálculos, o que significa que para valores de  $f_n$  elevados o sistema tende a ficar instável.

#### 4.4.2 Jacobiano Transposto e Jacobiano Inverso : Sub-controladores não lineares

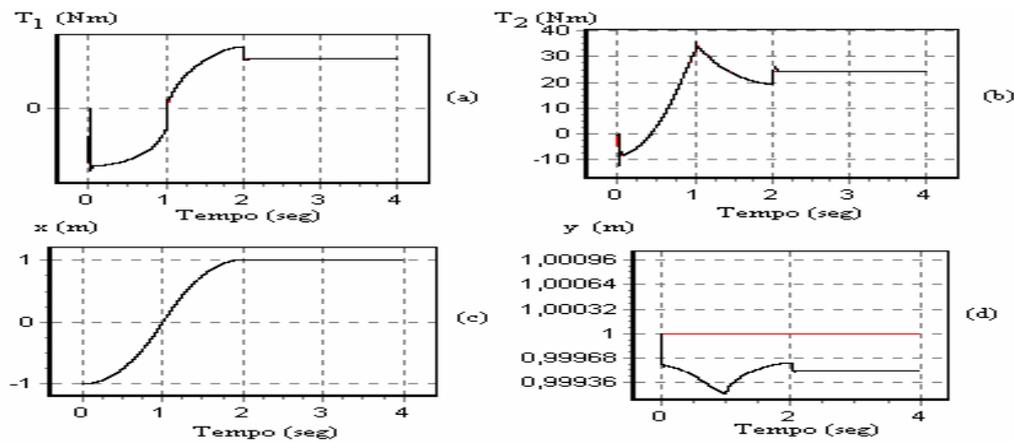
Os controladores pelos métodos do Jacobiano Transposto e do Jacobiano Inverso apresentam ganhos idênticos aos algoritmos não lineares com o controlo nas juntas para  $f_n=10\text{Hz}$ .



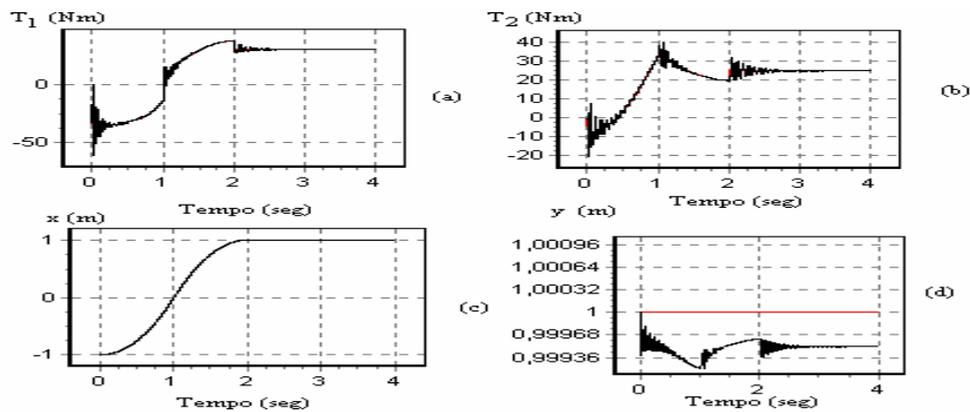
**Figura 4.23** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador VSS-FOM-PDD<sup>2</sup> pelo método Jacobiano Inverso com os parâmetros da tabela 4.4 (*i.e.*  $f_n=10\text{Hz}$ ), considerando a vermelho a trajectória planeada e cor de rosa a simulação.



**Figura 4.24** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador VSS-FOM-PDD<sup>2</sup> pelo método Jacobiano Inverso com os parâmetros da tabela 4.5 (*i.e.*  $f_n=100\text{Hz}$ ), considerando a vermelha a trajectória planeada e roxo a simulação.



**Figura 4.25** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador VSS-FOM-PDD<sup>2</sup> pelo método Jacobiano Transposto com os parâmetros da tabela 4.4 (*i.e.*  $f_n=10\text{Hz}$ ), considerando a vermelha a trajectória planeada e preto a simulação.

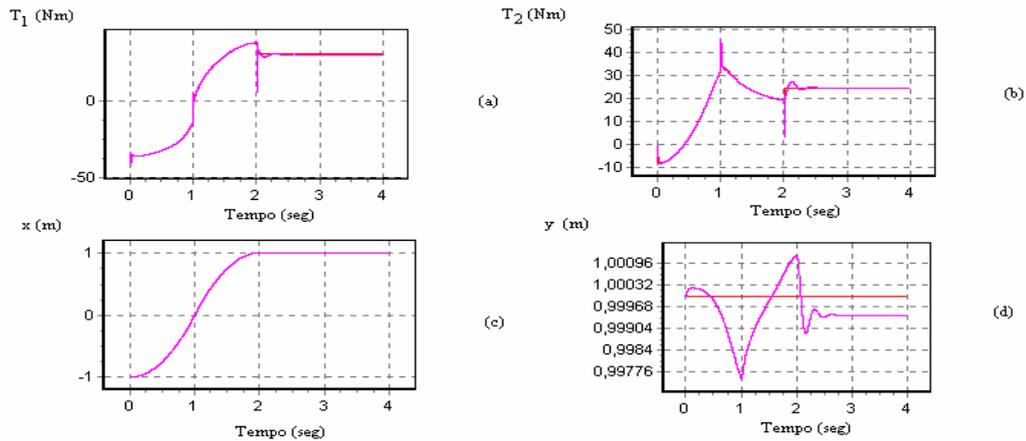


**Figura 4.26** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador VSS-FOM-PDD<sup>2</sup> pelo método Jacobiano Transposto com os parâmetros da tabela 4.5 (*i.e.*  $f_n=100\text{Hz}$ ), considerando a vermelha a trajetória planeada e preto a simulação.

Comparando os controladores do tipo VSS-FOM-PDD<sup>2</sup> verifica-se que para os métodos do Jacobiano Inverso e do Jacobiano Transposto, com o controlo baseado no espaço operacional o desempenho piora para frequências  $f_n$  elevadas.

### 4.4.3 Controlador baseado no modelo do robô no espaço operacional

As simulações que se apresentam a seguir é para o controlador baseado no modelo do robô no espaço cartesiano, os ganhos apresentados são idênticos aos algoritmos baseados em modelos com o controlo nas juntas para  $f_n=10\text{Hz}$ .



**Figura 4.27** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador baseado no modelo do robô no espaço cartesiano com os parâmetros da tabela 4.4 (*i.e.*  $f_n=10\text{Hz}$ ).

Verifica-se que este método apresenta um pior desempenho (comparativamente) que os modelos baseados no modelo no espaço das juntas.

Este facto é devido à inserção da cinemática directa na realimentação das posições e das velocidades das juntas, o que provoca o aumento da carga computacional. Pode verificar-se [2] que existem outros tipos de algoritmos baseados em modelos no espaço.

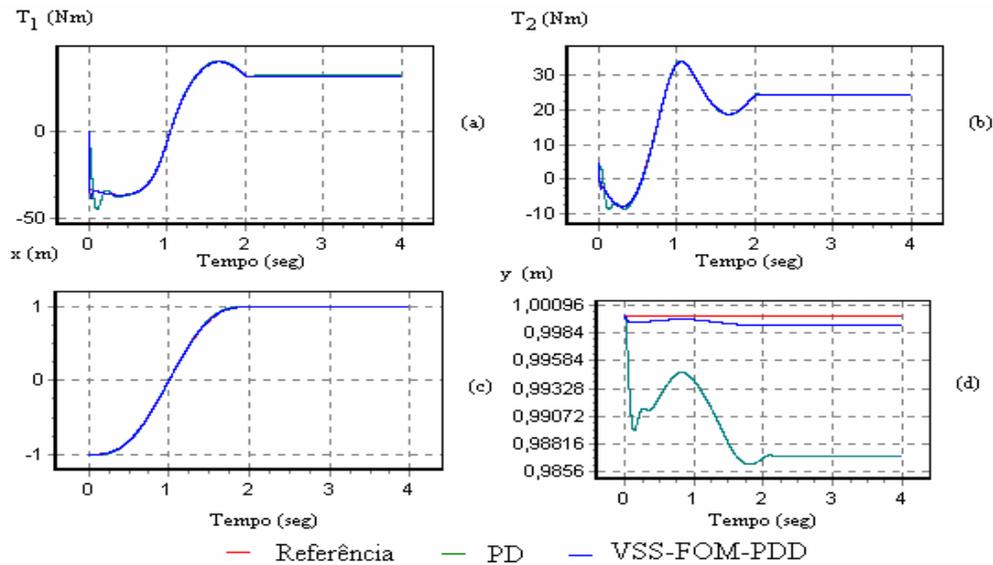
## 4.5 Simulação de trajectórias

As experiências seguintes pretendem visualizar e comparar o comportamento dos algoritmos face a vários tipos de planeamento da trajectória. Desta forma, fez-se uma simulação dos diversos tipos de planeamentos do simulador *RobLib*. Foram planeadas trajectórias no espaço das juntas com aceleração sinusoidal, no espaço cartesiano, quer com aceleração em degrau quer com aceleração sinusoidal.

As simulações, seja para os algoritmos lineares seja para não lineares, adoptam a frequência  $f_n=10\text{Hz}$  (*i.e.* de  $\xi = 1$  e um  $\omega_n = 62.8 \text{ rad. s}^{-1}$ ). O robô tem a estrutura RR com os parâmetros tabela 4.1 Além disso, consideraram-se os mesmos pontos inicial e final da recta  $(-1,1) \rightarrow (1,1)$ .

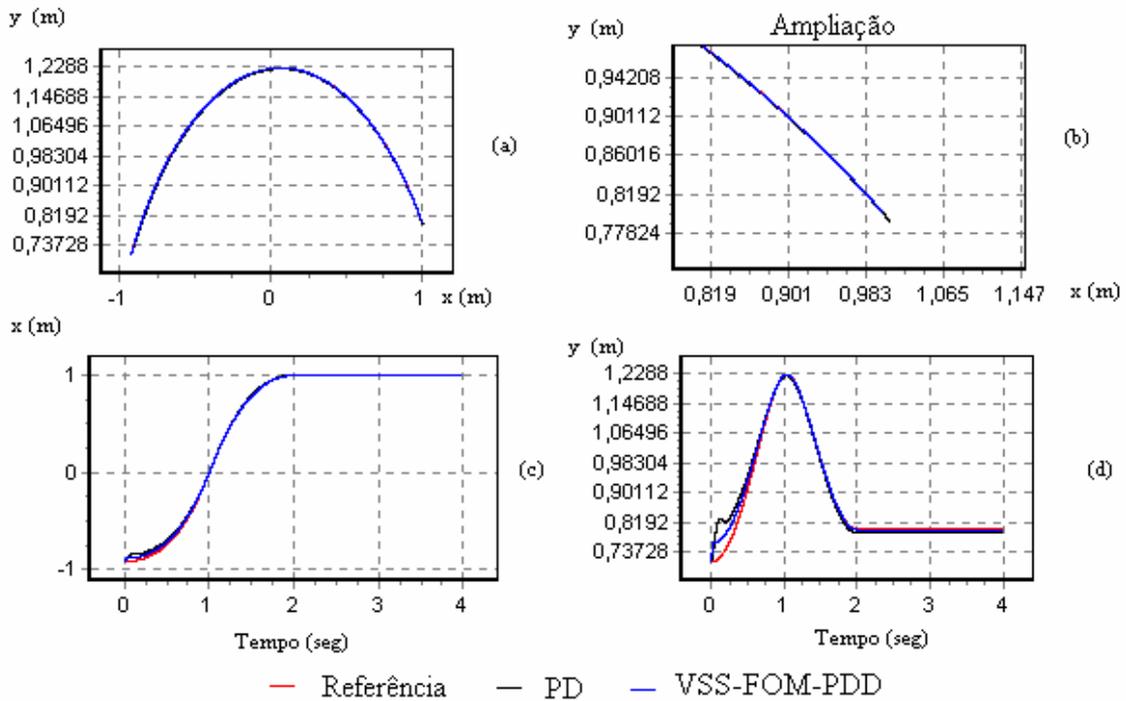
Note-se que a trajectória no espaço cartesiano com aceleração em degrau foi efectuada na secção anterior.

Para o planeamento da trajectória no espaço cartesiano com aceleração sinusoidal são apresentados na figura 4.28, a evolução da trajectória no caso dos algoritmos linear e não linear.



**Figura 4.28** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador PD com os parâmetros da tabela 4.2 (*i.e.*  $f_n=10\text{Hz}$ ) e o controlador VSS-FOM-PDD<sup>2</sup> com os parâmetros da tabela 4.4 (*i.e.*  $f_n=10\text{Hz}$ ).

Considera-se agora o planeamento da trajectória no espaço das juntas com aceleração sinusoidal e para uma recta com os ponto inicial e final  $(q_1, q_2)_A=(90^\circ, 35^\circ)$   $(q_1, q_2)_B=(45^\circ, 25^\circ)$ , respectivamente. Pode-se, verificar na figura 4.29, o comportamento dos algoritmos face à alteração do planeamento da trajectória.

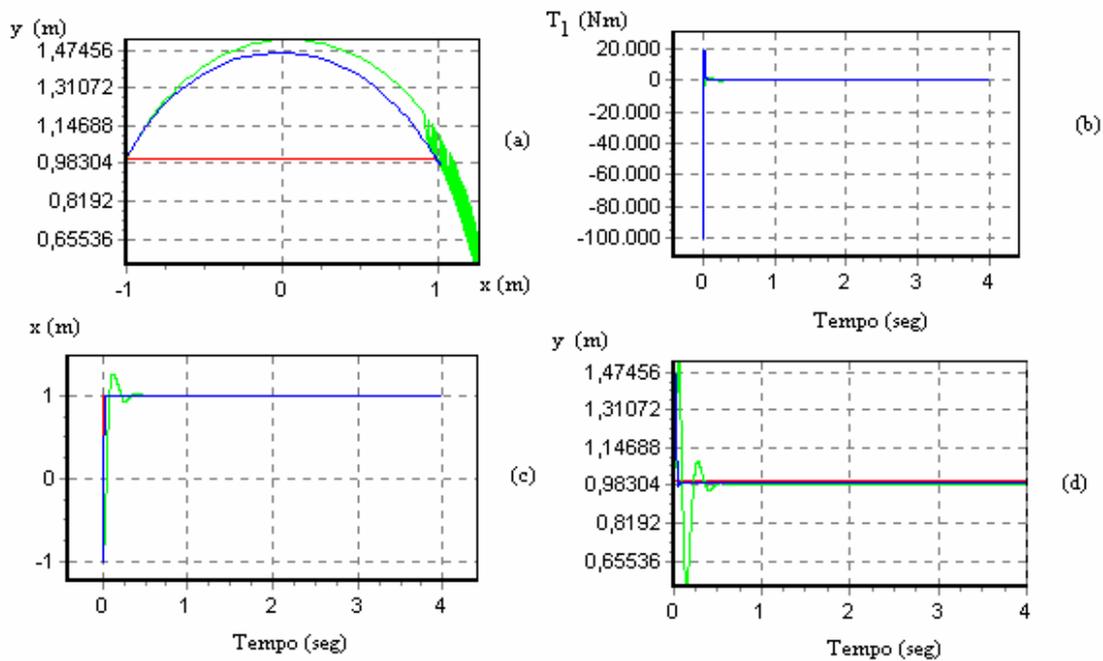


**Figura 4.29** – (a) evolução temporal da trajectória  $y(x)$ , (b) ampliação evolução temporal da trajectória  $y(x)$ , (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador PD com os parâmetros da tabela 4.2 (*i.e.*  $f_n=10\text{Hz}$ ) e o controlador VSS-FOM-PDD<sup>2</sup> com os parâmetros da tabela 4.4 (*i.e.*  $f_n=10\text{Hz}$ ).

Desta forma, o controlador linear PD, para uma frequência natural do sistema relativamente baixa  $f_n=10\text{Hz}$ , apresenta um *overshoot* na ordem de 0,25% do valor final.

É de notar que tanto para trajectórias no espaço das juntas, com aceleração sinusoidal, como no espaço cartesiano, quer com aceleração em degrau quer com aceleração sinusoidal, o comportamento dos algoritmos é semelhante

Para um planeamento de trajectória do tipo ponto a ponto (*i.e.* considerando os mesmos pontos inicial e final da trajectória da figura 4.1), obtêm-se a figura 4.30.



**Figura 4.30** – (a) evolução temporal da trajectória  $y(x)$ , (b) evolução temporal  $T_1(t)$  do binário da junta 1, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador PD com os parâmetros da tabela 4.2 (*i.e.*  $f_n=10\text{Hz}$  e o controlador VSS-FOM-PDD<sup>2</sup> com os parâmetros da tabela 4.4 (*i.e.*  $f_n=10\text{Hz}$ ).

Pode verificar-se que, apesar de não terem sido planeados os diversos pontos intermédios da trajectória, simulação apresentada na figura 4.30, os algoritmos estudados conseguem movimentar o robô para o ponto final desejado com grande rapidez já que se efectua uma trajectória de 2 metros em menos de 0,1 segundos. No entanto, os binários dos actuadores (não representados), atingem valores bastante elevados o que, na prática, não é realizável devido às limitações dos actuadores.

## 4.6 Algoritmos de controlo baseados no Modelo do Robô

Os algoritmos baseados em modelos requerem a modelização do robô. Assim, se esta modelização não for perfeita, o seu controlo pode apresentar algumas dificuldades. Por esta razão, é inserido um controlador para compensar qualquer desvio da trajectória.

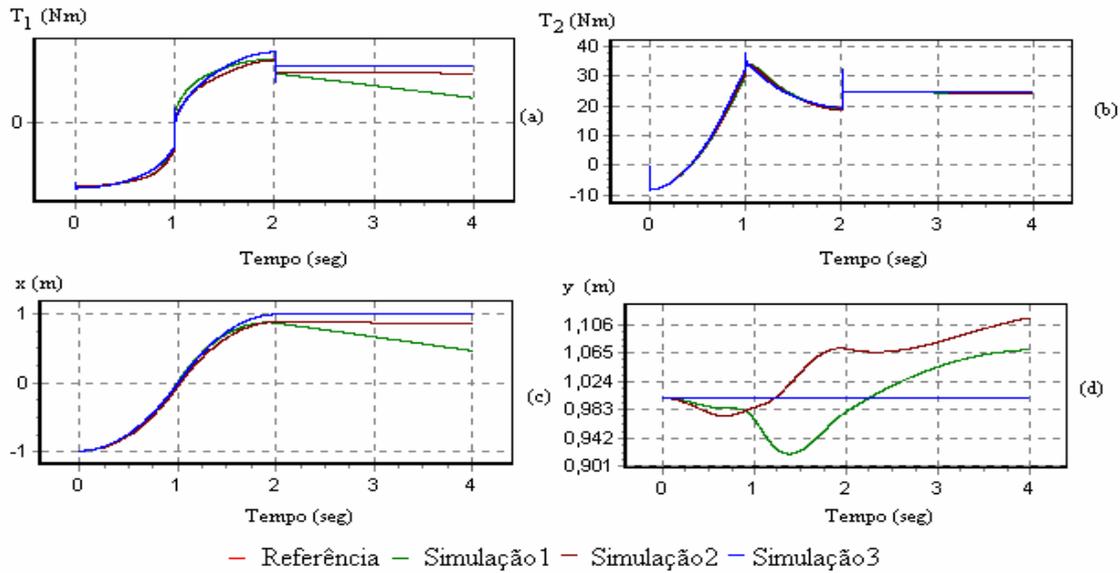
Nesta perspectiva, as simulações apresentadas nesta secção permitem observar o efeito dos algoritmos face a erros de modelização do robô.

O primeiro conjunto de simulações corresponde a uma modelização perfeita para o algoritmo *Computed Torque*, mas apresentando diferentes ganhos nos controladores ( figura 4.31).

O segundo conjunto mostra as simulações quando os parâmetros do modelo do robô não coincidem com valores reais, nomeadamente, nas situações de sobre-estimação e sub-estimação da massa de carga do manipulador (figuras 4.33 e 4.34).

Apresentam-se as simulações do controlador do binário calculado:

- ganhos de posição e de velocidade nulos ( $K_p=0$  ,  $K_v=0$ ).
- ganhos de posição de velocidade baixos ( $K_p=2$  ,  $K_v=1$ ).
- ganhos de posição de velocidade elevados (*i.e.*  $K_p=394784$  e  $K_v=1256$ ).

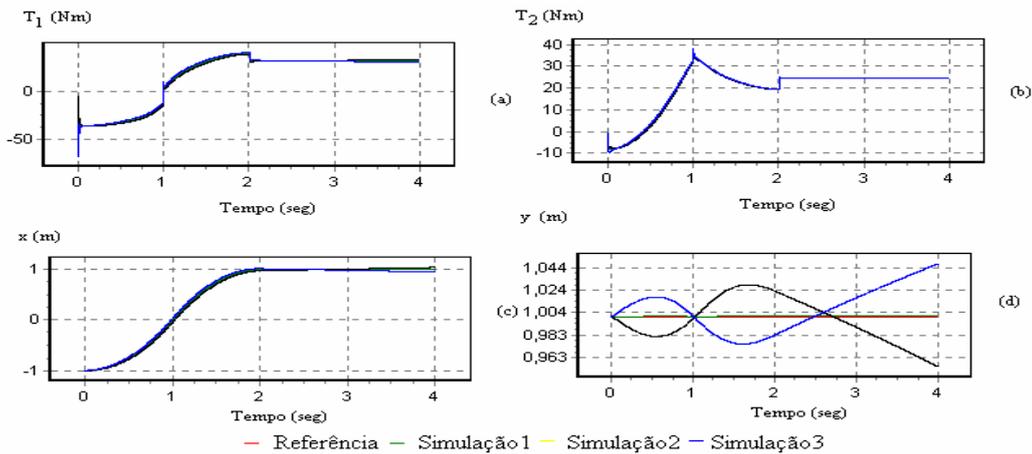


**Figura 4.31** – Simulação para o algoritmo *Computed Torque*. (a) evolução temporal da trajectória  $y(x)$ , (b) evolução temporal  $T_1(t)$  do binário da junta 1, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para a simulação 1 os ganhos de posição e de velocidade são nulos, na simulação 2 os ganhos de posição de velocidade são baixos e para a simulação 3 os ganhos de posição de velocidade são elevados.

No segundo conjunto de simulações consideram-se os casos indicados no esquema da figura 4.32:

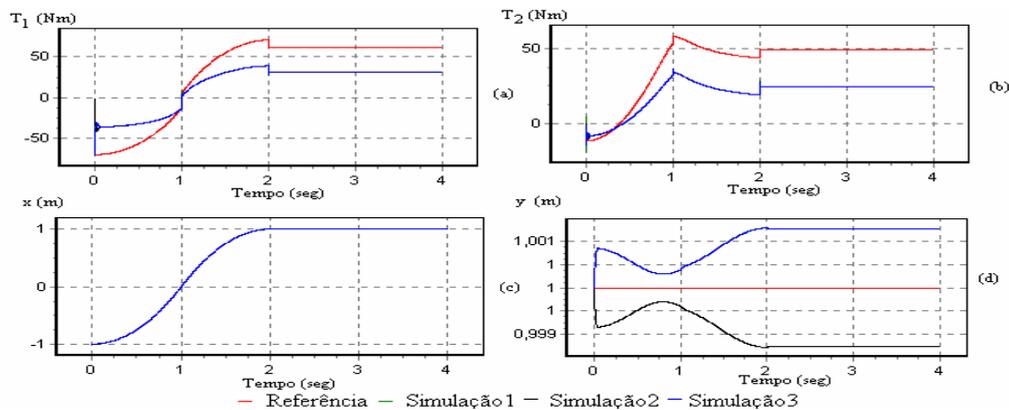


**Figura 4.32** - Esquema das simulações para os algoritmos *Computed Torque* e *Feedforward*.



**Figura 4.33** – Simulação para o algoritmo *Computed Torque*, (a) evolução temporal da trajectória  $y(x)$ , (b) evolução temporal  $T_1(t)$  do binário da junta 1, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para a simulação 1 os ganhos de posição e de velocidade são nulos, na simulação 2 os ganhos de posição de velocidade são baixos e para a simulação 3 os ganhos de posição de velocidade são elevados.

Para o controlador com acção antecipativa (*Feedforward*) com sub-controladores lineares ou não lineares, consideram-se os ganhos anteriormente indicados (secção 4.1) para  $f_n=10\text{Hz}$ .



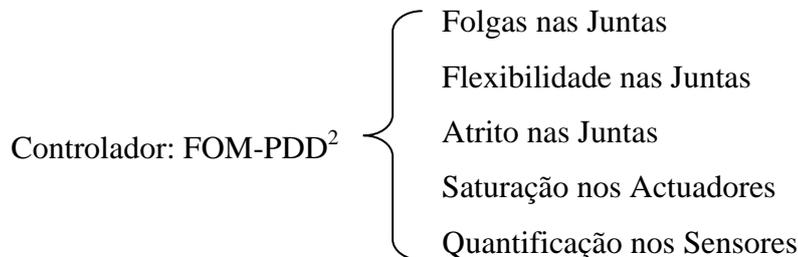
**Figura 4.34** – Simulação para o algoritmo *Feedforward*, (a) evolução temporal da trajectória  $y(x)$ , (b) evolução temporal  $T_1(t)$  do binário da junta 1, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para a simulação 1 os ganhos de posição e de velocidade são nulos, na simulação 2 os ganhos de posição de velocidade são baixos e para a simulação 3 os ganhos de posição de velocidade são elevados.

Através destas simulações pode constatar-se que este tipo de algoritmos eliminam o erro em regime permanente e reduzem significativamente o erro de seguimento da trajectória, comparativamente aos controladores lineares apresentados na secção anterior. Deve salientar-se que a estimação exacta do modelo do robô manipulador, favorece o funcionamento do controlador, mas não elimina totalmente os erros de acompanhamento de trajectória devido à frequência finita de amostragem. Além disso, pode observar-se que a situação de sobre-estimação da carga acarreta erros inferiores, aos que ocorrem na situação oposta de sub-estimação da carga.

## 4.7 Fenómenos dinâmicos nas juntas dos robôs

Nesta secção simulam-se diversos fenómenos dinâmicos nas juntas de acordo com o esquema da figura 4.35. As experiências foram efectuadas para o robô RR com os mesmos parâmetros da secção 4.2.

O controlador, que se considerou nesta análise, é do tipo VSS-FOM-PDD<sup>2</sup> em virtude de ser bastante robusto e de não necessitar de um modelo matemático do sistema a controlar. Além disso, considerou-se os mesmos ganhos do robô ideal da secção 4.2 (*i.e.*  $f_n = 10$  e  $f_n = 100\text{Hz}$ ).



**Figura 4.35** - Esquema das simulações de fenómenos dinâmicos nas juntas.

Na modelização do robô com folgas consideraram-se os parâmetros das tabelas 4.6 e 4.7. Na simulação das juntas flexíveis utilizou-se os parâmetros das tabelas 4.8 e 4.9. Por seu lado, na experiência com atrito não-linear nas juntas os parâmetros considerados são os indicados na tabela 4.10. Nas experiências envolvendo saturação apresentam-se três possíveis parametrizações da saturação dos binários nos actuadores, respectivamente, nas tabelas 4.12, 4.13 e 4.14.

**Tabela 4.6** – Parâmetros das folgas do robô.

$i$	$\varepsilon$	$h$ (rad./s)
1	0,9	0,01
2	0,9	0,01

**Tabela 4.7** – Parâmetros das folgas do robô.

$i$	$\varepsilon$	$h$ (rad./s)
1	0,8	0,01
2	0,8	0,01

**Tabela 4.8** - Parâmetros da flexibilidade nas juntas do robô.

$i$	$B_i(kgm^2)$	$K_i(kgm^2)$
1	1000	100000
2	1000	100000

**Tabela 4.9** – Parâmetros das flexibilidade nas juntas do robô.

$i$	$B_i(kgm^2)$	$K_i(kgm^2)$
1	100	20000
2	100	20000

**Tabela 4.10** - Parâmetros do atrito não linear das juntas do robô.

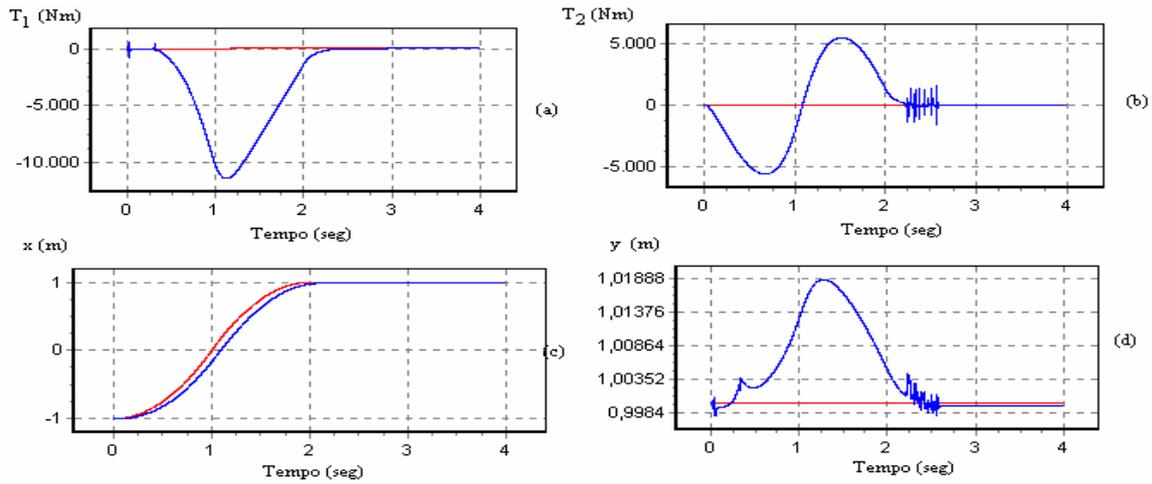
$I$	$A$	$B_i$	$K_i(kgm^2)$
		$(kgm^2)$	
1	0	10	100
2	0	10	100

**Tabela 4.11** - Parâmetros da saturação dos actuadores do robô.

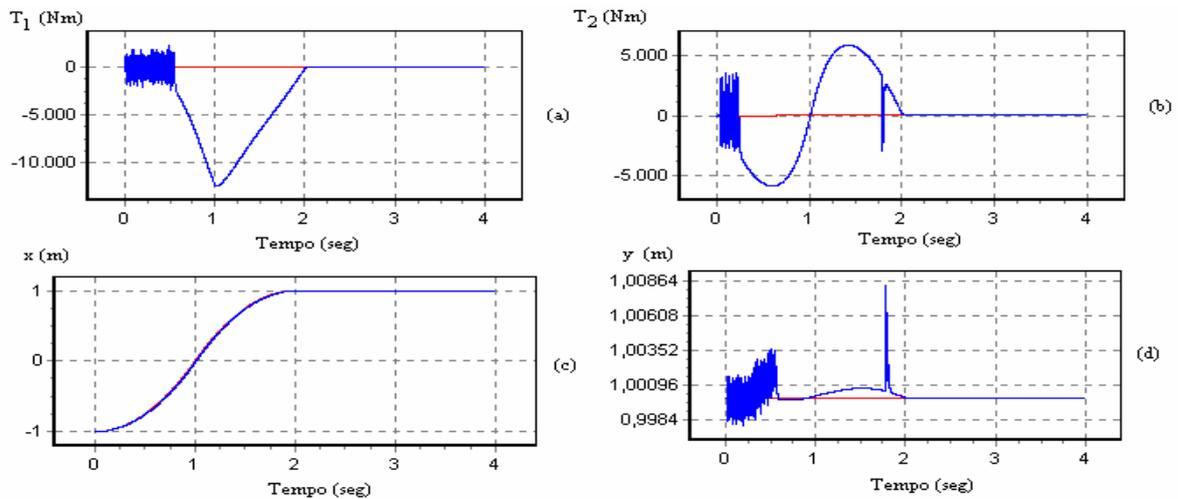
$i$	$T_{iMax}(Nm)$	$T_{iMin}(Nm)$
1	40	-40
2	35	-35

Por último, na modelização da quantificação dos níveis dos sensores adoptaram-se três e quatro casas decimais.

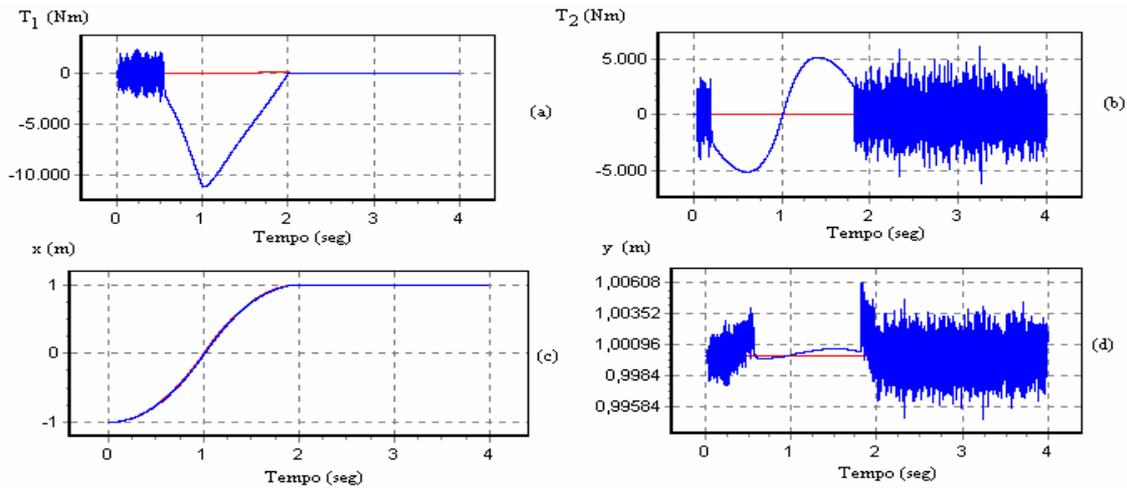
Foram apresentadas na secção 4.1 as simulações para o robô ideal. Para o primeiro conjunto de simulações, com folgas nas juntas, efectuaram-se três experiências alterando os parâmetros, nomeadamente, a constante de elasticidade e os ganhos. (figuras 4.36 a 4.38).



**Figura 4.36** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador VSS-FOM-PDD<sup>2</sup> com os parâmetros da tabela 4.4 (*i.e.*  $f_n=10\text{Hz}$ ) e para o robô com os parâmetros da tabela 4.6, considerando a vermelho a trajectória planeada e azul a simulação.



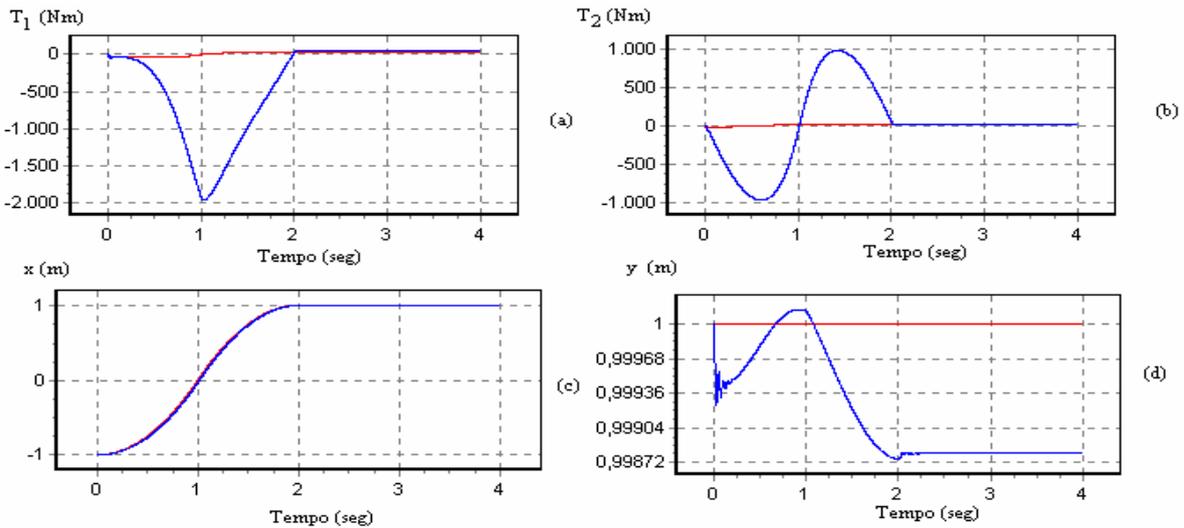
**Figura 4.37** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador VSS-FOM-PDD<sup>2</sup> com os parâmetros da tabela 4.5 (*i.e.*  $f_n=100\text{Hz}$ ) e para o robô com os parâmetros da tabela 4.6, considerando a vermelho a trajectória planeada e azul a simulação.



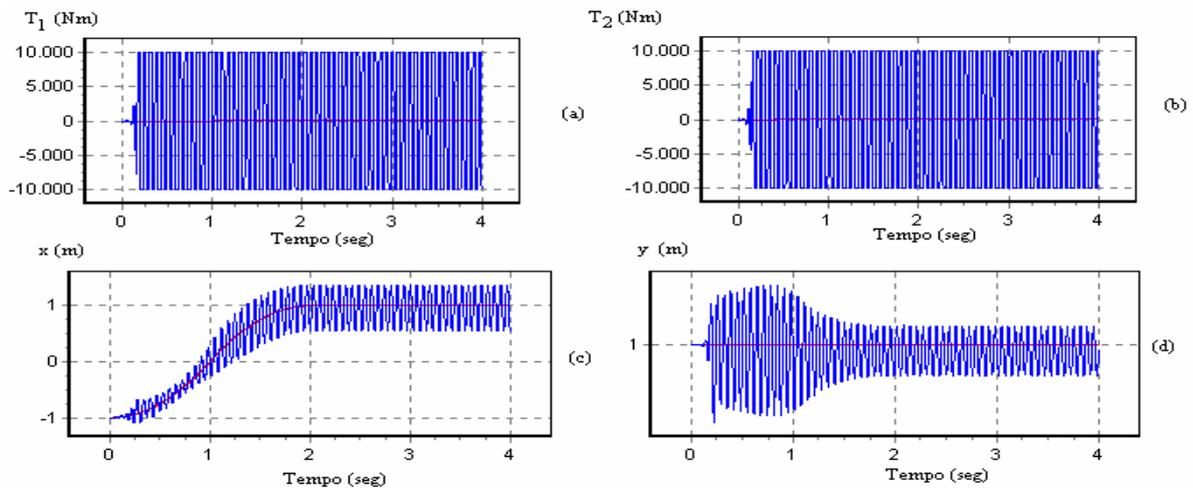
**Figura 4.38** – Para  $f_n=100\text{Hz}$ , (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador VSS-FOM-PDD<sup>2</sup> com os parâmetros da tabela 4.5 (*i.e.*  $f_n=100\text{Hz}$ ) e para o robô com os parâmetros da tabela 4.7, considerando a vermelho a trajectória planeada e azul a simulação.

Nestas experiências nota-se que o aumento da frequência de amostragem melhora o desempenho do controlo do robô. Da mesma forma, observa-se que a diminuição da constante de elasticidade produz uma maior oscilação em regime permanente, como se pode ver no gráfico da figura 4.37.

Para a situação de juntas com flexibilidade, apresentam-se, nas figuras 4.39 e 4.40, duas simulações para valores distintos dos parâmetros da rigidez  $K_m$  do motor e da transmissão, mantendo o mesmo factor de amortecimento  $B_m$ .



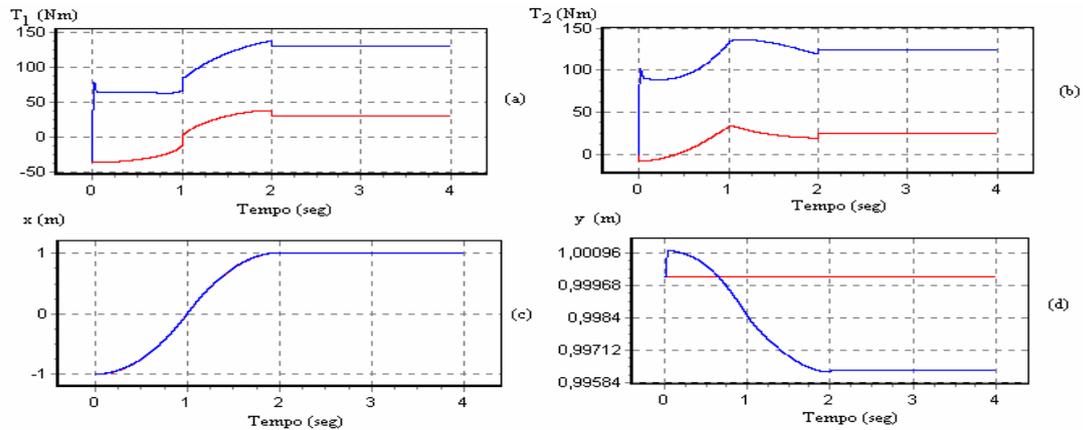
**Figura 4.39** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador VSS-FOM-PDD<sup>2</sup> com os parâmetros da tabela 4.4 (*i.e.*  $f_n=10\text{Hz}$ ) e para o robô com os parâmetros da tabela 4.8, considerando a vermelho a trajetória planeada e azul a simulação.



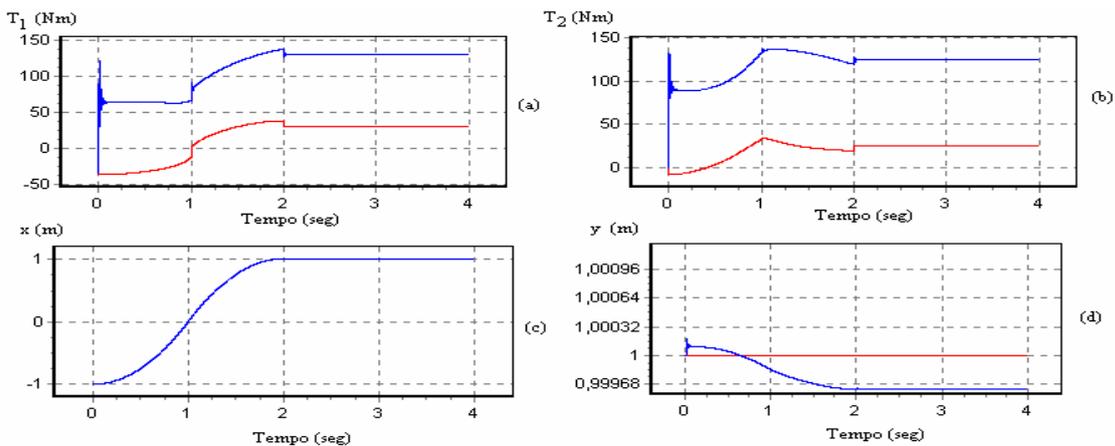
**Figura 4.40** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador VSS-FOM-PDD<sup>2</sup> com os parâmetros da tabela 4.4 (*i.e.*  $f_n=10\text{Hz}$ ) e para o robô com os parâmetros da tabela 4.9, considerando a vermelho a trajetória planeada e azul a simulação.

Verifica-se que a diminuição dos parâmetros da rigidez  $K_m$  e do factor de amortecimento  $B_m$  provoca oscilações do sistema, revelando-se difícil de controlar.

As simulações representadas nas figuras 4.41 e 4.42 correspondem à existência de atrito não-linear nas juntas do robô.



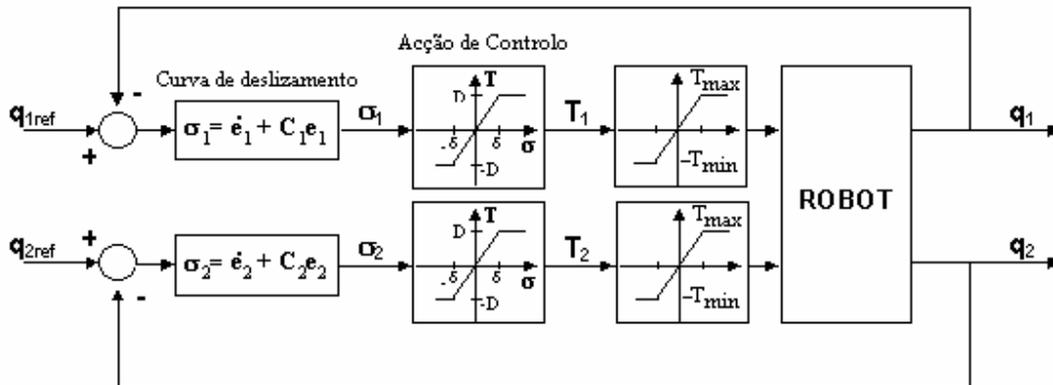
**Figura 4.41** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador VSS-FOM-PDD<sup>2</sup> com os parâmetros da tabela 4.4 (*i.e.*  $f_n=10\text{Hz}$ ) e para o robô com os parâmetros da tabela 4.10, considerando a vermelho a trajectória planeada e azul a simulação.



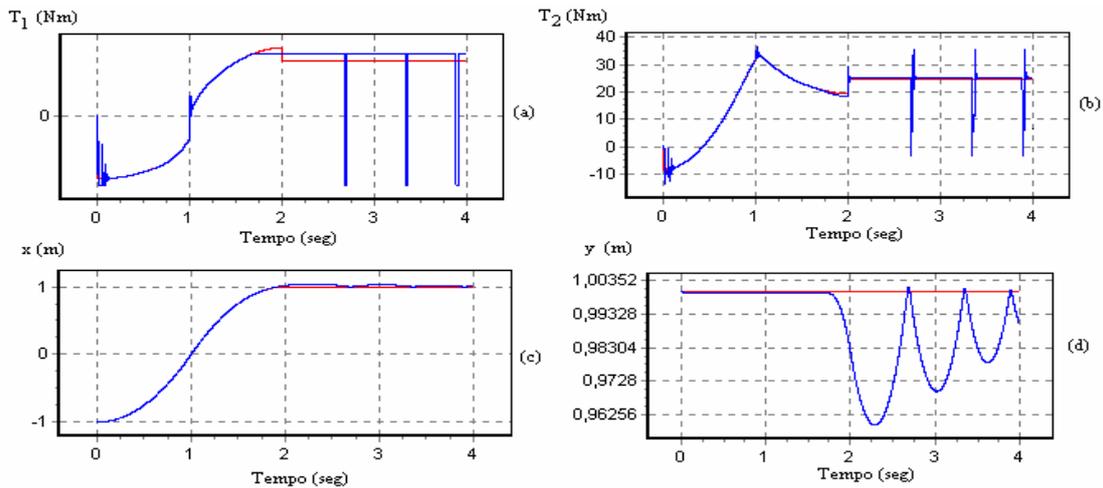
**Figura 4.42** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador VSS-FOM-PDD<sup>2</sup> com os parâmetros da tabela 4.5 (*i.e.*  $f_n=100\text{Hz}$ ) e para o robô com os parâmetros da tabela 4.10, considerando a vermelho a trajectória planeada e azul a simulação.

Verifica-se nas figuras 4.41 e 4.42 no gráfico d) o atraso da trajectória simulada face à trajectória de referência. Observa-se na figura 4.42 que para frequências de amostragem mais elevadas surge uma maior oscilação transitória.

A saturação dos actuadores corresponde ao modelo representado na figura 4.43.



**Figura 4.43** – Modelo da saturação nos actuadores.



**Figura 4.44** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador VSS-FOM-PDD<sup>2</sup> com os parâmetros da tabela 4.4 (*i.e.*  $f_n=10\text{Hz}$ ) e para o robô com os parâmetros da tabela 4.11, considerando a vermelho a trajectória planeada e azul a simulação.

Nesta experiência limitou-se o valor dos binários máximo e mínimo dos actuadores. O efeito deste fenómeno observa-se um pouco antes dos dois segundos (*i.e.* quando o robô requer um binário mais elevado que o do actuador como se observa na figura 4.44.). A introdução da saturação produz alterações significativas no desempenho

do robô, provocando oscilações bruscas no actuador, para compensar os desvios da trajectória de referência.

A modelização da quantificação finita dos sensores de posição está ilustrada na figura 4.45. Assim, foram simulados vários níveis de quantificação das posições angulares, correspondentes a diferentes truncagens dos bits dos conversores A/D (*i.e.* o número níveis é igual a  $2^n$ , onde  $n$  é o número de bits).

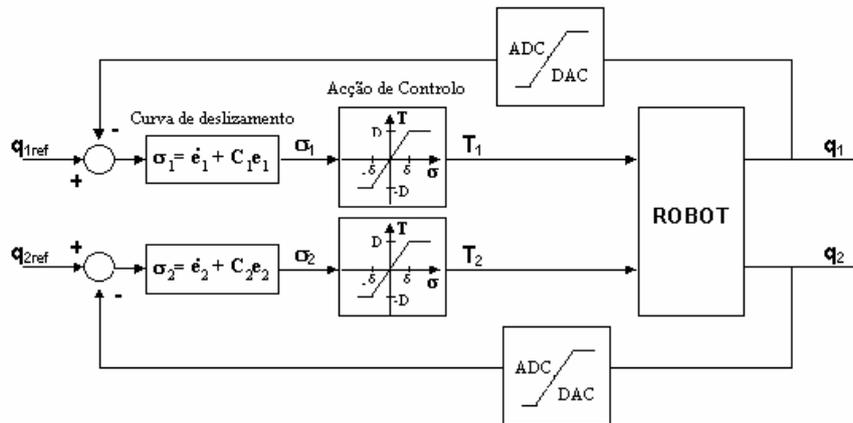


Figura 4.45 – Modelo da quantificação finita dos sensores de posição.

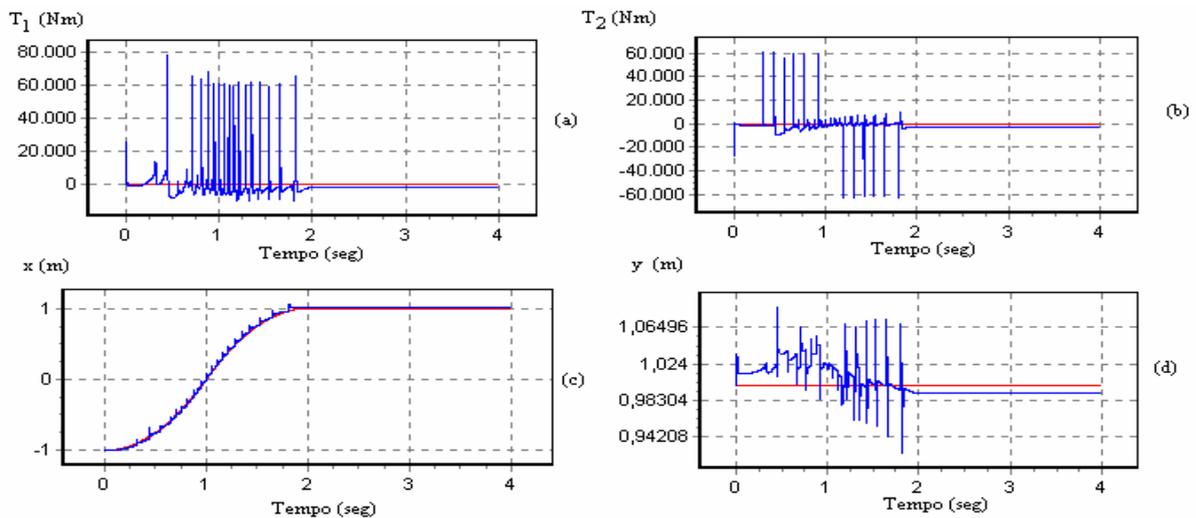
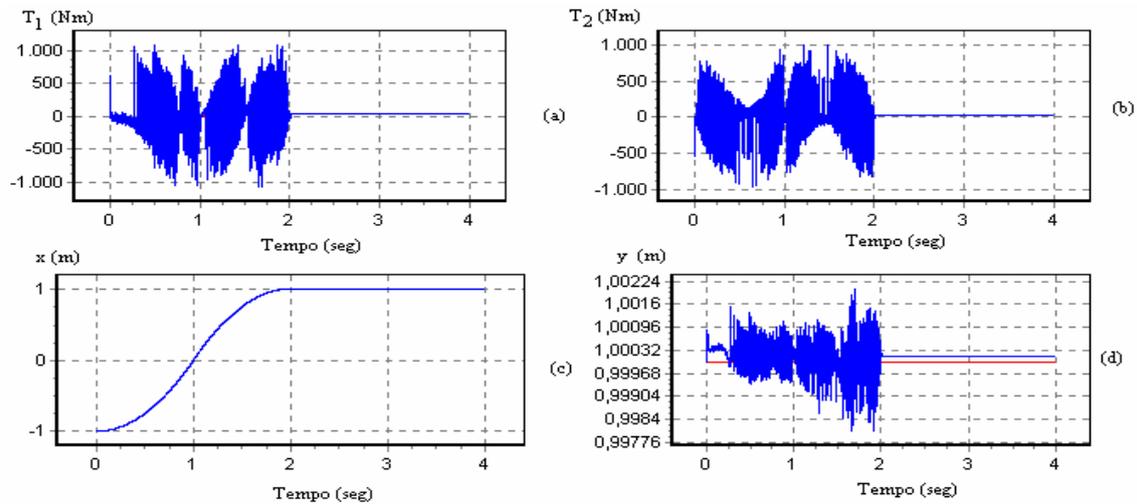


Figura 4.46 – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador VSS-FOM-PDD<sup>2</sup> com os parâmetros da tabela 4.4 (*i.e.*  $f_n=10\text{Hz}$ ) e para um sensor com  $n=3$  níveis de quantificação das posições angulares, considerando a vermelha a trajectória planeada e azul a simulação.



**Figura 4.47** – (a) evolução temporal  $T_1(t)$  do binário da junta 1, (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o controlador VSS-FOM-PDD<sup>2</sup> com os parâmetros da tabela 4.4 (*i.e.*  $f_n=10\text{Hz}$ ) e para um sensor com  $n=4$  níveis de quantificação das posições angulares, considerando a vermelha a trajectória planeada e azul a simulação.

A identificação e a modelização de fenómenos dinâmicos é um problema de difícil solução, não somente na robótica mas também para muitos outros sistemas. Os fenómenos dinâmicos, tais como folgas, atritos não-lineares ou flexibilidade das transmissões mecânicas nas articulações do manipulador, são importantes no desempenho dinâmico, mas são complexos de modelizar. Todavia, esses fenómenos podem ser estudados de uma forma simplificada, pelo que se apresenta uma comparação para os vários casos obtidos no simulador. Os resultados correspondem ao mesmo controlador e a parâmetros iguais, de modo a poderem-se comparar entre si.

Para as juntas flexíveis, o fenómeno dinâmico que se observa é bastante oscilatório e difícil de controlar. Contudo, se os valores de  $K_m$  e  $B_m$  forem menores, e mantendo os ganhos do controlador, os resultados obtidos melhoram. Verifica-se que o erro de seguimento nas juntas flexíveis aumenta quando se dá a inversão de velocidade (na experiência 4.39 para  $t=1$  segundo).

O mesmo se passa para as folgas nas juntas. Se aumentarmos o ganho do controlador o desempenho do robô apresenta-se bastante melhor. O fenómeno das folgas faz-se notar, significativamente, quando se dá uma inversão da velocidade.

O atrito nas juntas origina, valores bastante elevados nos binários dos actuadores. Isto deve-se ao facto de que os valores introduzidos na modelização do atrito também são elevados. Se diminuirmos os valores de  $\mathbf{K}$  e de  $\mathbf{B}$  ou se aumentarmos o ganho do controlador, o tempo de atraso observado nas simulações diminuirá. Verifica-se ainda que o atrito não-linear nas juntas origina um atraso constante da trajectória em relação à trajectória de referência.

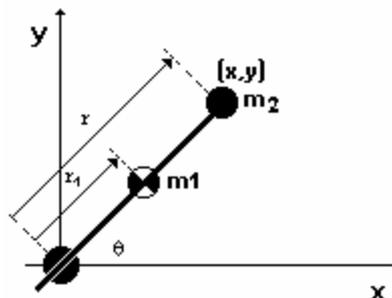
Os valores para a saturação dos actuadores foram atribuídos de modo a limitar tanto nos binários máximos como nos mínimos, um só dos actuadores, ou os dois actuadores. O erro introduzido pelo fenómeno da saturação aumenta quando os actuadores atingem o máximo ou no mínimo o valor do binário de saturação.

## 4.8 Controlo do robô RP

Nesta secção são apresentadas várias simulações com robô RP, para a mesma trajectória da secção 4.1, com vista a analisar os desempenhos dos algoritmos para robôs com estruturas mecânicas distintas.

As simulações seguintes correspondem ao primeiro conjunto de simulações realizada para o robô RR. As restantes experiências não foram efectuadas visto que muitos dos efeitos encontrados no controlo do robô RR são semelhantes para o robô RP.

O robô é simulado através do método de integração numérica de Runge-Kutta de ordem quatro a uma frequência de  $f_{sim}=100\text{kHz}$  e os controladores adoptam uma frequência de amostragem  $f_c=10\text{kHz}$ .



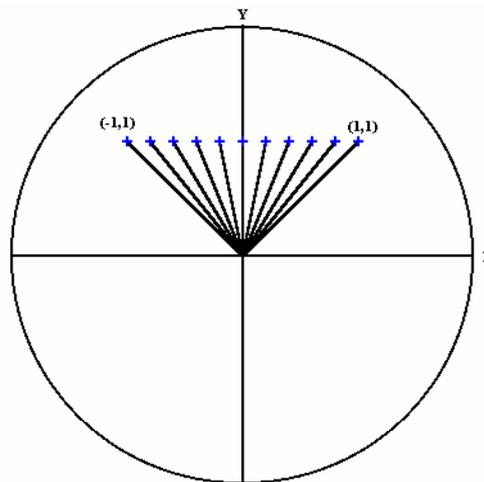
**Figura 4.48** – Robô RP ideal.

Os parâmetros utilizados para o robô ideal nesta simulação encontram-se na tabela 4.12.

**Tabela 4.12** – Parâmetros do robô RP.

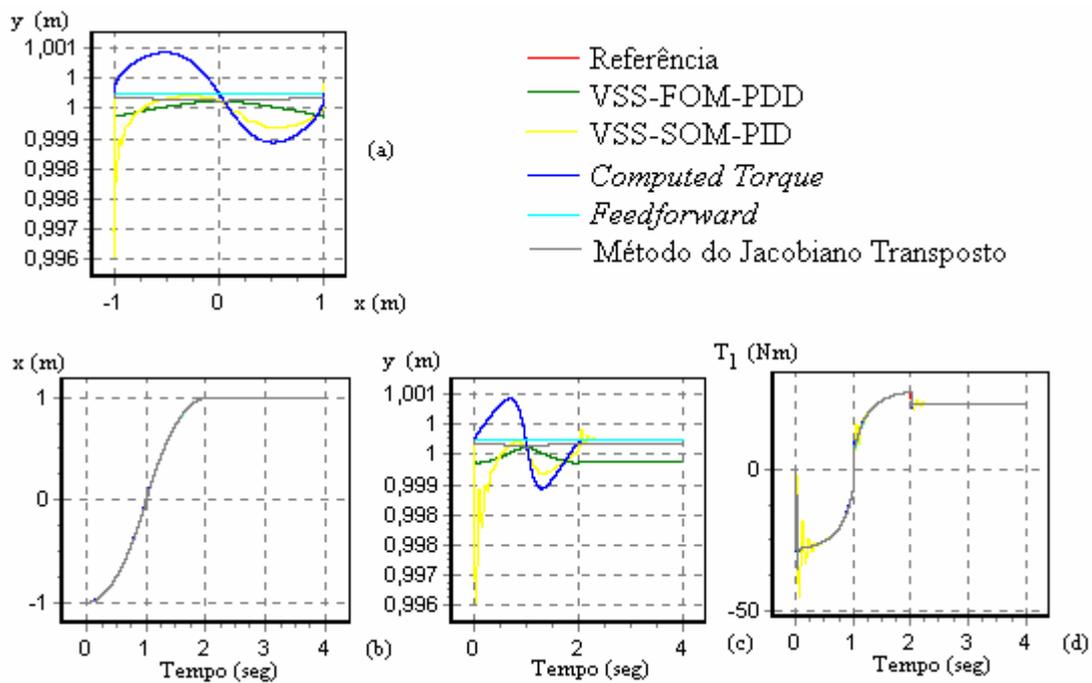
$i$	$m_i(kg)$	$r_i(m)$	$J_{im}(kgm^2)$	$J_{ig}(kgm^2)$
1	0,5	1,0	1,0	1,0
2	6,25	0,8	1,0	1,0

A trajectória teste para o robô RP está representada na figura 4.49:



**Figura 4.49** –Trajectória do Robô RP.

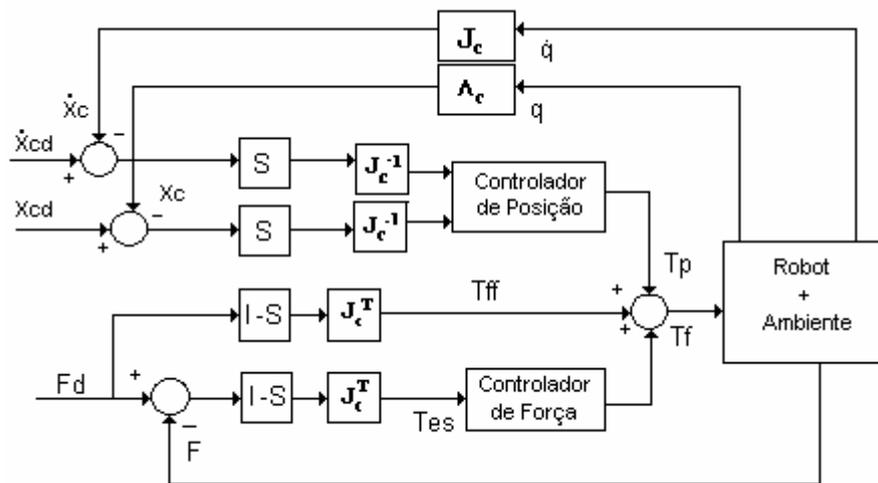
A figura 4.50 apresenta um subconjunto das possíveis simulações e os resultados e conclusões são semelhantes às obtidas anteriormente para o robô RR.



**Figura 4.50** – (a) evolução da trajectória  $y(x)$ , (b) evolução temporal  $x(t)$  da posição do eixo  $x$ , (c) evolução temporal  $y(t)$  da posição do eixo  $y$ , (d) evolução temporal  $T_1(t)$  do binário da junta 1, para todos os controladores considerou-se a  $f_n=10\text{Hz}$ .

## 4.9 Simulação do Algoritmo Controlo Híbrido

Como mencionado, anteriormente, o algoritmo CH, serve para controlar a posição e a força em situações que envolvem o contacto do robô com uma superfície. Todavia, para aplicações como, por exemplo, a pintura, a opção será de controlo de posição em ambos os eixos. Deste modo, neste caso apenas funcionam os blocos correspondentes ao controlador de posição, como se constata no diagrama de blocos da figura 4.51. No caso de uma aplicação como, por exemplo, a rebarbagem ou o seguimento de contornos, a opção de controlo será o controlo de posição segundo uma direcção e força de contacto na outra.

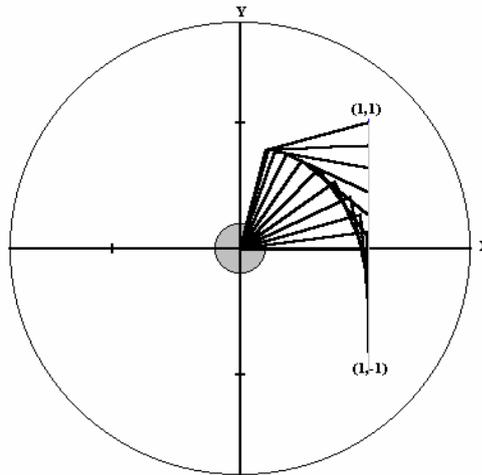


**Figura 4.51** – Controlador híbrido.

Nesta ordem de ideias, as simulações apresentadas nas subsecções seguintes testam as diversas opções de controlo do algoritmo CH para o robô RR ideal. Um conjunto final de simulações analisa os efeitos dos fenómenos dinâmicos nas juntas do robô a quando do controlo de posição numa determinada direcção e de força na outra. Assim, compara-se o desempenho do algoritmo face à existência de fenómenos, tais como o atrito não-linear, as folgas nas engrenagens e a flexibilidade nas transmissões.

Planeou-se uma trajectória com uma evolução temporal da aceleração em degrau como os seguintes parâmetros:

- posição inicial (1,1);
- posição final (1,-1);
- tempo de simulação de 4 segundos;
- tempo de trajectória de 2 segundos;
- aplicação de força após decorridos 0,1 segundos.
- os parâmetros da superfície de restrição  $M_i=0,001 \text{ kg}$ ,  $B_i=1 \text{ Nsm}^{-1}$ ,  $K_i=100 \text{ Nm}^{-1}$  e  $M_i=0,001\text{kg}$ ,  $B_i=10 \text{ Nsm}^{-1}$  e  $K_i=100 \text{ Nm}^{-1}$



**Figura 4.52** – Movimento do robô na superfície de restrição.

### 4.9.1 Simulação do controlo de posição

Se se adoptar um controlo de posição em ambas as direcções então o sub-controlador de força CH (independentemente do controlador seleccionado) “não tem significado”, visto que a matriz  $I-S$  (ver no diagrama de blocos da figura 4.51) é nula e o anel de controlo de força não entra em funcionamento.

Os sub-controladores de posição adoptados de seguida são do tipo VSS-FOM-PDD<sup>2</sup> e do tipo VSS-SOM-PID com os ganhos apresentados, respectivamente, nas tabelas 4.13 e 4.14.

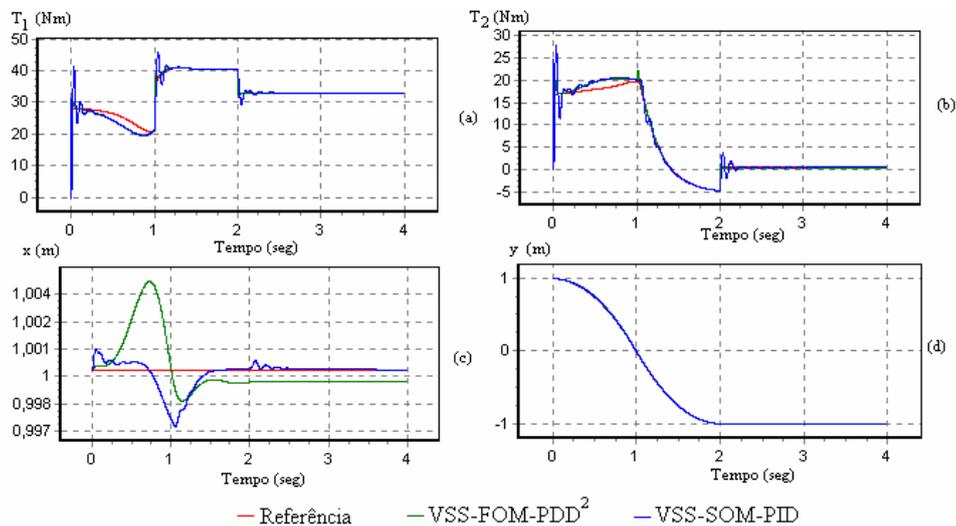
**Tabela 4.13** – Parâmetros do controlador VSS-FOM-PDD<sup>2</sup>.

$i$	$C_i$	$D_i$	$d_i$
1	100	100	1000000
2	100	100	1000000

**Tabela 4.14** - Parâmetros do controlador VSS-SOM-PID.

$I$	$C_i$	$D_i$	$d_i$
1	100	100	10000
2	100	100	10000

Para o caso de um simples controlo de posição obtêm-se os gráficos do acompanhamento da trajectória, a evolução temporal das posições e os binários nos actuadores representados na figura 4.53.



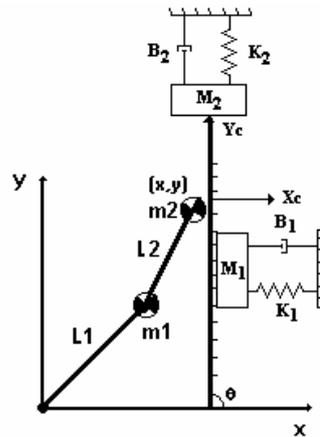
**Figura 4.53** – (a) evolução temporal  $T_1(t)$  do binário da junta 1 (b) evolução temporal  $T_2(t)$  do binário da junta 2, (c) evolução temporal  $x(t)$  da posição do eixo x, (d) evolução temporal  $y(t)$  da posição do eixo y, para  $f_n=10\text{Hz}$ .

Pode verificar-se pela figura anterior que o algoritmo do tipo VSS-SOM-PID apresenta um comportamento bastante melhor que o algoritmo VSS-FOM-PDD<sup>2</sup>. Nota-se também que o efeito da acção integral do controlador VSS-SOM-PID elimina o erro em regime permanente.

### 4.9.2 Simulação do controlo de força

No controlo de força de ambos os eixos efectuaram-se três simulações para diferentes ganhos do anel de força. Assim, vai-se aplicar no ponto (1,1) uma força  $F_{xd}=3\text{N}$  segundo o eixo x e uma força de  $F_{yd}=1\text{N}$  segundo o eixo y.

Considerou-se uma superfície de restrição com uma inclinação de  $\theta = \frac{\pi}{2}$  como se pode ver na figura 4.54.



**Figura 4.54** – Robô RR e superfície de restrição para  $\theta = \frac{\pi}{2}$ .

A restrição de contacto foi modelizada por um sistema mola-massa-amortecedor. Na primeira e na segunda simulações os parâmetros da superfície de restrição são  $M_i=0,001$  kg,  $B_i=1$  Nsm<sup>-1</sup> e  $K_i=100$  Nm<sup>-1</sup> e na terceira simulação os parâmetros são  $M_i=0,001$  kg,  $B_i=10$  Nsm<sup>-1</sup> e  $K_i=100$  Nm<sup>-1</sup>. Para um controlo de força seguindo ambos os eixos isto significa colocar a matriz  $S$  igual a zero pelo que não existe qualquer influência do anel de controlo de posição.

As simulações efectuadas para o sub-controlador de força adoptaram um algoritmo VSS-FOM-PDD<sup>2</sup> com os parâmetros apresentados nas tabelas 4.15 e 4.16.

**Tabela 4.15** – Parâmetros do controlador de força do tipo VSS-FOM-PDD<sup>2</sup>.

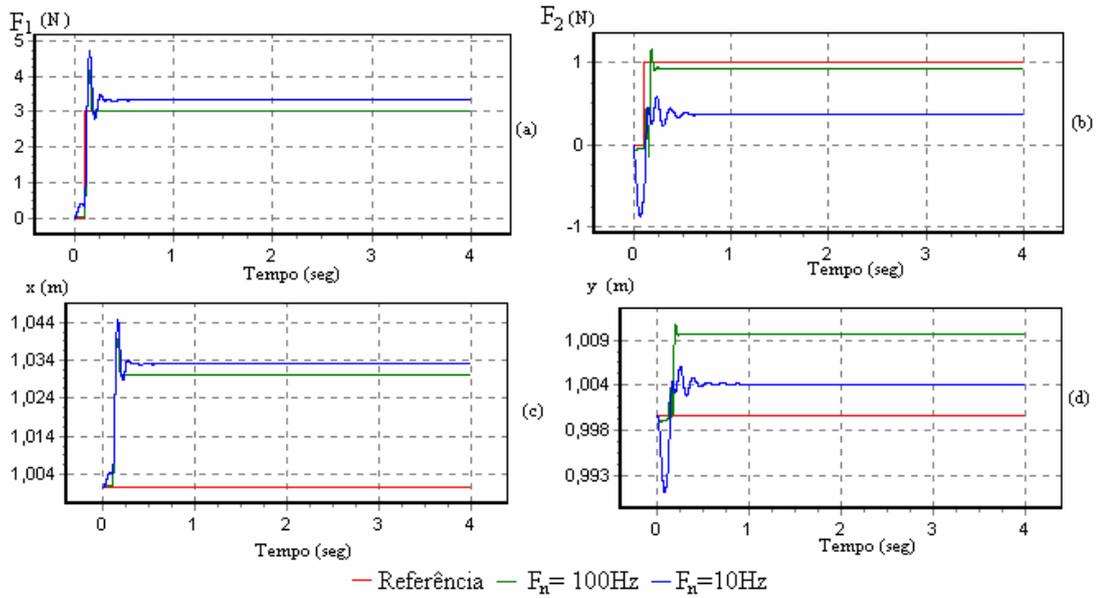
$i$	$C_i$	$D_i$	$d_i$
1	100	100	200
2	100	100	200

**Tabela 4.16** – Parâmetros da superfície de restrição.

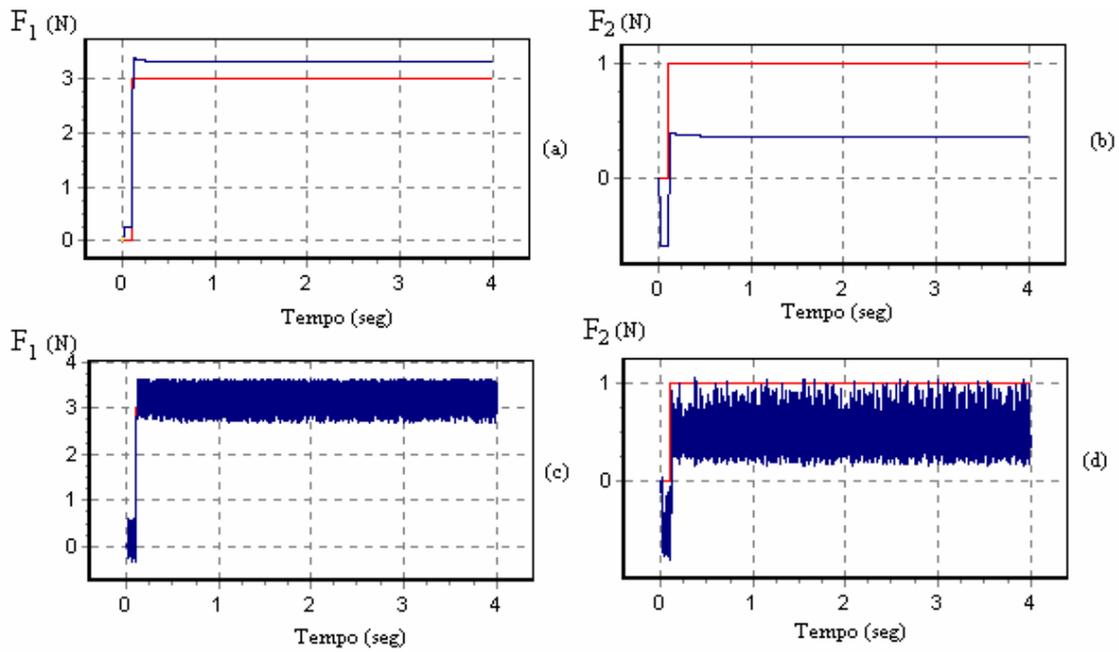
$i$	$B_i$	$K_i$	$M_i$
x	1	100	0,001
y	1	100	0,001

**Tabela 4.17** – Parâmetros da superfície de restrição.

$i$	$B_i$	$K_i$	$M_i$
x	10	100	0,001
y	10	100	0,001



**Figura 4.55** – (a) evolução temporal  $F_1(t)$ , (b) evolução temporal  $F_2(t)$ , (c) evolução temporal  $x(t)$  da posição do eixo x, (d) evolução temporal  $y(t)$  da posição do eixo y, para os parâmetros do controlador na tabela 4.15 e da superfície de restrição na tabela 4.16.



**Figura 4.56** – (a) evolução temporal  $F_1(t)$ , (b) evolução temporal  $F_2(t)$ , (c) evolução temporal  $F_1(t)$ , (d) evolução temporal  $F_2(t)$ , para os parâmetros do controlador na tabela 4.15 e da superfície de restrição na tabela 4.17. As figuras a) e b) são para  $f_n=10\text{Hz}$  e as figuras c) e d) para  $f_n=100\text{Hz}$ .

Observa-se nestas experiências que o robô exerce uma força, num ponto do seu espaço de trabalho e para um robô de 1,8 metros de comprimento este apresenta um erro bastante pequeno.

Constata-se na primeira experiência (figura 4.56) que o aumento da frequência de amostragem melhora o desempenho do algoritmo. Na segunda experiência observa-se que o aumento da constante de amortecimento do modelo da superfície de restrição influencia desfavoravelmente o controlo do robô, considerando os ganhos utilizados na experiência anterior.

### 4.9.3 Simulação dos fenómenos dinâmicos nas juntas no controlo de posição segundo o eixo y e força segundo o eixo x

Nesta secção apresenta-se o controlo de posição/força do robô considerando os diferentes fenómenos dinâmicos nas juntas. O planeamento da trajectória apresenta-se na figura 4.52, com uma superfície de restrição  $\theta = \frac{\pi}{2}$  como se ilustra na figura 4.54. Os parâmetros da superfície de restrição são  $M_i=0,01$  kg,  $B_i=1$  Nsm<sup>-1</sup>e  $K_i=100$  Nm<sup>-1</sup>, e a frequência de amostragem do controlador  $f_c=10$ kHz. Os subcontroladores utilizados no anel de posição são do tipo VSS-FOM-PDD<sup>2</sup>, com os ganhos apresentados na tabela 4.15, e do tipo sub-controlador VSS-FOM-PID, com os ganhos apresentados na tabela 4.16. Para o sub-controlador de força adoptou-se o VSS-FOM-PDD<sup>2</sup>, com os ganhos apresentados na tabela 4.20.

**Tabela 4.18** – Controlador de posição do tipo VSS-FOM-PDD<sup>2</sup>.

<i>i</i>	<i>C<sub>i</sub></i>	<i>D<sub>i</sub></i>	<i>d<sub>i</sub></i>
1	100	100	100000
2	100	100	100000

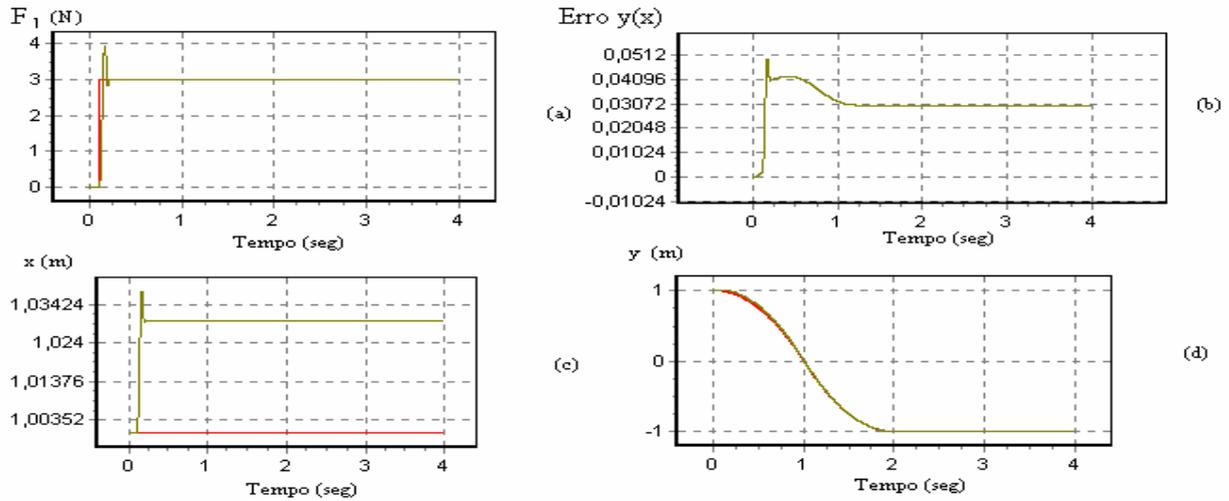
**Tabela 4.19** - Controlador de posição do tipo VSS-FOM-PID.

<i>I</i>	<i>C<sub>i</sub></i>	<i>D<sub>i</sub></i>	<i>d<sub>i</sub></i>
1	100	100	10000
2	100	100	10000

**Tabela 4.20** - Controlador de força do tipo VSS-FOM-PDD<sup>2</sup>.

<i>I</i>	<i>C<sub>i</sub></i>	<i>D<sub>i</sub></i>	<i>d<sub>i</sub></i>
1	100	10	100
2	100	10	100

Numa primeira simulação considerou-se o robô ideal que conduziu aos resultados apresentados na figura 4.57.



**Figura 4.57** – (a) evolução temporal  $F_1(t)$ , (b) erro  $e_p$  de seguimento de trajectória  $y(x)$ , (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o robô ideal, com o controlador de posição da tabela 4.19 e para o controlador de força da tabela 4.20.

De seguida, considerou-se um robô RR com juntas flexíveis com os parâmetros da tabela 4.21. Adoptaram-se os mesmos sub-controladores e os parâmetros, para o controlo de posição e de força são apresentados, respectivamente, nas tabelas 4.22 e 4.23. Os resultados obtidos encontram-se na figura 4.58.

**Tabela 4.21** - Parâmetros do robô com flexibilidade nas juntas.

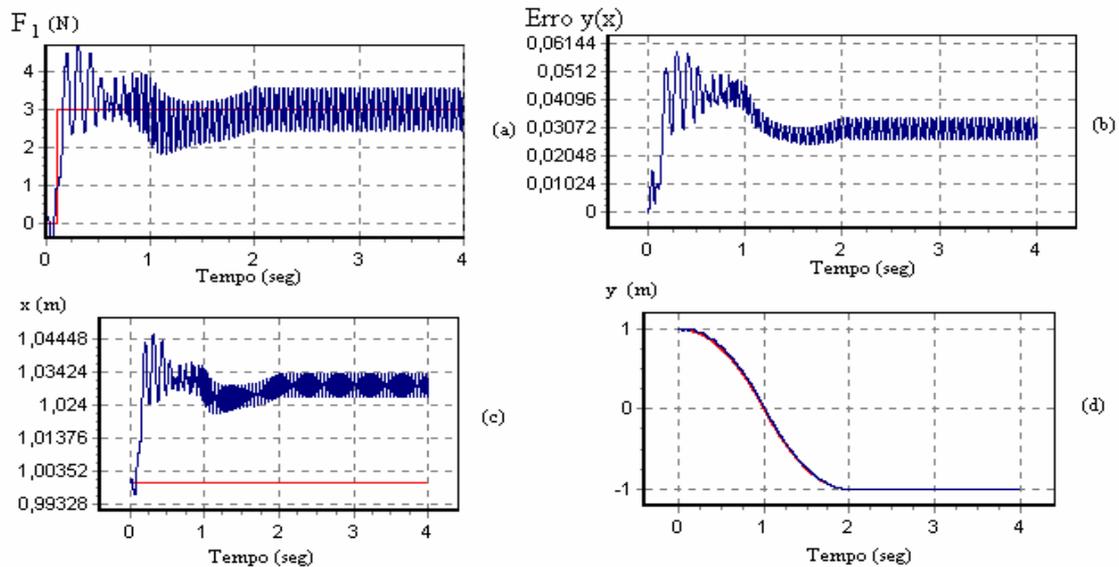
$i$	$B_i(kgm^2)$	$K_i(kgm^2)$
1	100	20000
2	100	20000

**Tabela 4.22.**- Parâmetros do sub-controlador de posição.

$i$	$C_i$	$D_i$	$d_i$
1	100	100	100000
2	100	100	100000

**Tabela 4.23-** Parâmetros do sub-controlador de força.

$i$	$C_i$	$D_i$	$D_i$
1	100	100	10000
2	100	100	10000



**Figura 4.58** – (a) evolução temporal  $F_1(t)$ , (b) erro  $e_p$  de seguimento de trajectória  $y(x)$ , (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o robô com juntas flexíveis, com o controlador de posição da tabela 4.22 e para o controlador de força da tabela 4.23, considerando a vermelho a trajectória planeada e azul a simulação.

Numa outra experiência considerou-se o robô RR com atrito não-linear nas juntas com os parâmetros da tabela 4.24. Além disso, os parâmetros dos subcontroladores de posição e de força são apresentados, respectivamente nas tabelas 4.25 e 4.26, os resultados estão apresentados na figura 4.59.

**Tabela 4.24** - Parâmetros do robô com atrito nas juntas.

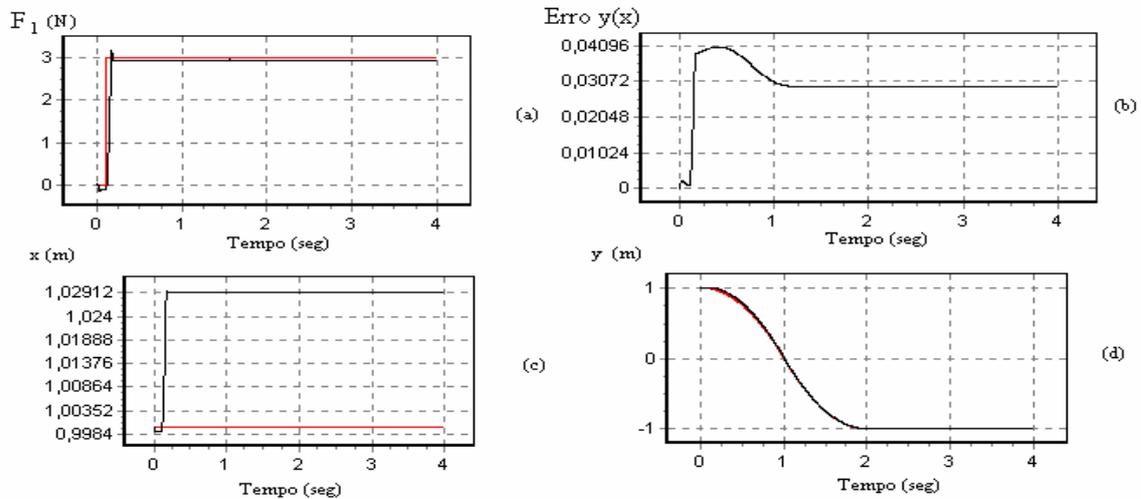
$I$	$A$	$B_i$ ( $kgm^2$ )	$K_i$ ( $kgm^2$ )
1	0	10	100
2	0	10	100

**Tabela 4.25**- Parâmetros do sub-controlador de posição.

$I$	$C_i$	$D_i$	$d_i$
1	100	100	100000
2	100	100	100000

**Tabela 4.26** - Parâmetros do sub-controlador de força.

$i$	$C_i$	$D_i$	$d_i$
1	100	100	10000
2	100	100	10000



**Figura 4.59** – (a) evolução temporal  $F_1(t)$ , (b) erro  $e_p$  de seguimento de trajectória  $y(x)$ , (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para robô com atrito não linear, com o controlador de posição da tabela 4.25 e para o controlador de força da tabela 4.26, considerando a vermelho a trajectória planeada e preto a simulação.

Por último, a simulação para o robô com folgas nas juntas foi efectuada com os parâmetros, indicados na tabela 4.27, enquanto que os ganhos dos subcontroladores de posição e de força estão representados nas tabelas 4.28 e 4.29, os resultados estão apresentados na figura 4.60.

**Tabela 4.27** - Parâmetros do robô com folgas nas juntas.

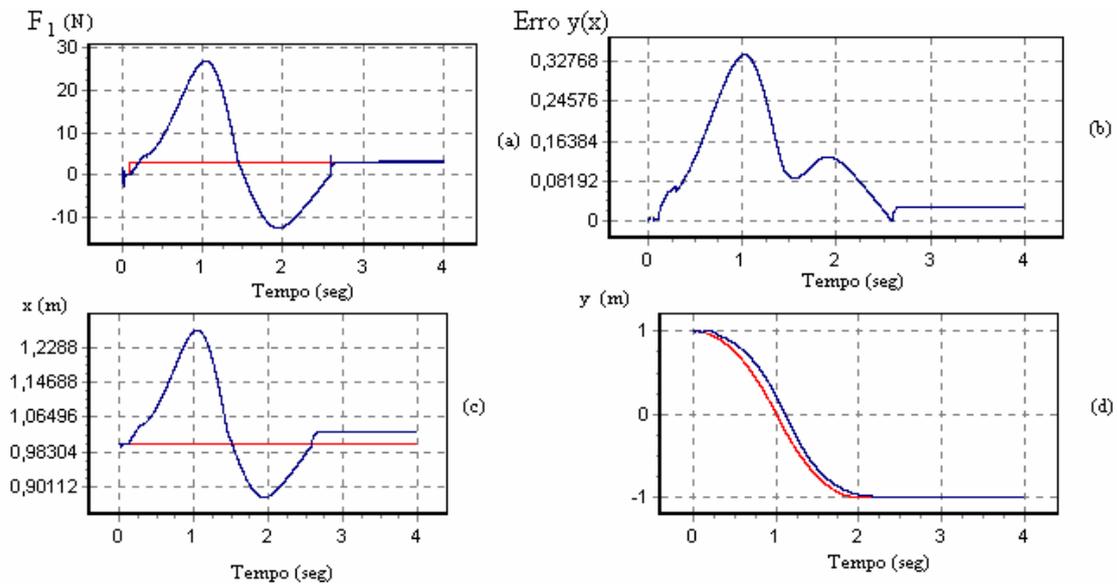
$i$	$e_{ci}$	$H_i$
	$0 \leq e_c \leq 1$	(rad)
1	0,9	0,01
2	0,9	0,01

**Tabela 4.28** - Parâmetros do sub-controlador de posição.

$I$	$C_i$	$D_i$	$d_i$
1	100	100	1000000
2	100	100	1000000

**Tabela 4.29** - Parâmetros do sub-controlador de força.

$I$	$C_i$	$D_i$	$d_i$
1	100	10	2000
2	100	10	2000



**Figura 4.60** – (a) evolução temporal  $F_1(t)$ , (b) erro  $e_p$  de seguimento de trajectória  $y(x)$ , (c) evolução temporal  $x(t)$  da posição  $x$ , (d) evolução temporal  $y(t)$  da posição  $y$ , para o robô com folgas nas juntas, com o controlador de posição da tabela 4.28 e para o controlador de força da tabela 4.29, considerando a vermelho a trajectória planeada e azul a simulação.

Em conclusão:

- Na experiência da flexibilidade nas juntas observam-se as oscilações devido ao valor reduzido da constante de rigidez da junta;
- Na experiência do atrito não linear nas juntas observa-se um atraso da trajectória simulada face à referência planeada.
- Na experiência com o efeito da folga da junta torna-se bastante difícil o controlo do sistema.

#### **4.10 Conclusões**

Neste capítulo estudaram-se (através do simulador *RobLib*) diversos algoritmos de controlo de posição e de posição/força. Assim, efectuaram-se numerosas experiências, tanto para o robô com estrutura RR, como para o robô RP, para diferentes estratégias de controlo e tipos de planeamentos de trajectórias.

Desenvolveram-se também diversas simulações, de modo a analisar o efeito de fenómenos dinâmicos nas juntas. Através da base de dados integrada no simulador estabeleceu-se uma comparação entre essas simulações e o caso de um robô ideal.

Por último, realizaram-se diversas simulações do algoritmo de controlo de posição/força de modo a analisar a dinâmica do sistema e os efeitos da superfície de restrição.

## CAPÍTULO 5

### **Manipulabilidade de Sistemas Robóticos com Um e Dois Braços**

#### **5.1 Introdução**

A escolha de um mecanismo robótico depende da tarefa ou do tipo de trabalho a efectuar. A selecção é determinada pela posição do robô, pelas suas dimensões e pela sua estrutura e, geralmente, é feita através da experiência e da intuição. Portanto, torna-se importante conceber uma medida quantitativa da capacidade da manipulação dos braços do robô, tanto em posição como em orientação, o que pode ser útil na concepção do robô, no controlo e no planeamento das trajectórias para efectuar as tarefas. Nesta perspectiva, foi proposto por Yoshikawa, em 1983, o conceito de medida de manipulabilidade cinemática [24]. Posteriormente, este e outros investigadores tentaram generalizar o conceito para manipulabilidade dinâmica com sucesso limitado[25][26].

Neste capítulo, pretende-se testar medidas da manipulabilidade, pelo que são apresentadas interpretações de algumas medidas, utilizando técnicas gráficas. Desta forma, vão-se efectuar simulações para o robô RR, e, de seguida, vai-se alargar este conceito para dois robôs RR a trabalhar em cooperação.

## 5.2 A Manipulabilidade do Robô RR

A medida da manipulabilidade do robô RR indica a melhor postura do robô no seu espaço de trabalho, no ponto de vista da manipulação de objectos.

Segundo Yoshikawa [24] para o robô RR a medida da manipulabilidade  $\mu$  é dada por:

$$\mu = [\det J(q)] = l_1 l_2 |\sin q_2| \quad (126)$$

que para o robô RR vem  $\mu = l_1 l_2 |\sin q_2|$ . A partir desta expressão verifica-se que a melhor postura do robô ocorre para  $q_2 = \pm 90^\circ$ . Além disso, para um, comprimento máximo constante, isto é para  $l_1 + l_2 = \text{constante}$ , a medida da manipulabilidade  $\mu$  atinge o seu máximo quando  $l_1 = l_2$  (*i.e.* para quaisquer valores de  $q_1$  e  $q_2$ ).

Para um robô, o desenvolvimento analítico da expressão relativa a  $\mu$  é fácil.

Todavia, para dois robôs não parece ser assim tão evidente. Deste modo, adopta-se de seguida, uma aproximação numérica, com semelhanças ao Método de Monte Carlo, aplicando também alguns conceitos relativos à Distribuição Normal Bidimensional.

Numa primeira fase, efectuaram-se simulações para um só robô a fim de comparar o novo método numérico com a expressão “clássica” de Yoshikawa e, de seguida, efectuam-se simulações para dois robôs a trabalhar em cooperação.

## 5.3 Cinemática de Dois Robôs em Cooperação

Muitas aplicações da robótica, tais como levantar objectos necessitam de mais do que um robô, nomeadamente de dois robôs a trabalhar em cooperação. Contudo, a utilização de vários robôs acarreta uma maior complexidade no seu estudo, pelo que este tema tem sido pouco abordado.

As equações cinemáticas relativas aos dois robôs RR representado na figura 5.1 são descritas nos pontos 5.3.1 a 5.3.4.

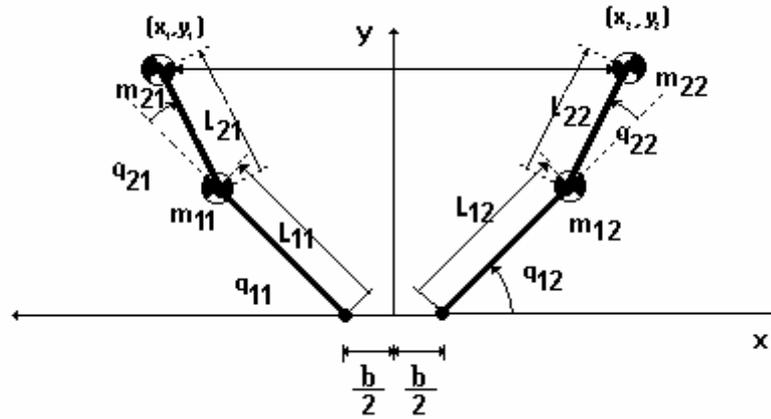


Figura 5.1 – Dois robôs a trabalhar em cooperação.

### 5.3.1 A cinemática directa do robô A

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} l_{11} \cdot \cos q_{11} + l_{21} \cdot \cos (q_{11} + q_{21}) - \frac{b}{2} \\ l_{11} \cdot \sin q_{11} + l_{21} \cdot \sin (q_{11} + q_{21}) \end{bmatrix} \quad (127)$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \mathbf{J}_A \begin{bmatrix} \dot{q}_{11} \\ \dot{q}_{21} \end{bmatrix} \quad (128)$$

$$\mathbf{J} = \begin{bmatrix} -l_{11} \cdot \sin q_{11} - l_{21} \cdot \sin (q_{11} + q_{21}) & -l_{21} \cdot \sin (q_{11} + q_{21}) \\ l_{11} \cdot \cos q_{11} + l_{21} \cdot \cos (q_{11} + q_{21}) & l_{21} \cdot \cos (q_{11} + q_{21}) \end{bmatrix} \quad (129)$$

### 5.3.2 A cinemática inversa do robô A

$$\begin{bmatrix} q_{11} \\ q_{21} \end{bmatrix} = \begin{bmatrix} \tan^{-1} \left( \frac{y}{x} \right) - \tan^{-1} \left( \frac{l_{21} \cdot \sin q_{21}}{l_{11} + l_{21} \cdot \cos q_{21}} \right) \\ \cos^{-1} \left( \frac{x^2 + y^2 - l_{11}^2 - l_{21}^2}{2 \cdot l_{11} \cdot l_{21}} \right) \end{bmatrix} \quad (130)$$

$$\begin{bmatrix} \dot{q}_{11} \\ \dot{q}_{21} \end{bmatrix} = \mathbf{J}_A^{-1} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \quad (131)$$

$$\begin{bmatrix} \dot{q}_{11} \\ \dot{q}_{21} \end{bmatrix} = \frac{1}{l_{11}l_{21}\text{sen}q_{21}} \underbrace{\begin{bmatrix} l_2 \cos(q_{11} + q_{21}) l_{21} \text{sen}(q_{11} + q_{21}) \\ -l_{11} \cos q_{11} - l_{21} \cos(q_{11} + q_{21}) & -l_{11} \text{sen} q_{11} - l_{21} \text{sen}(q_{11} + q_{21}) \end{bmatrix}}_{\mathbf{J}^{-1}} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \quad (132)$$

### 5.3.3 A cinemática directa do robô B

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} l_{12} \cdot \cos q_{12} + l_{22} \cdot \cos (q_{12} + q_{22}) + \frac{b}{2} \\ l_{12} \cdot \text{sen} q_{12} + l_{22} \cdot \text{sen} (q_{12} + q_{22}) \end{bmatrix} \quad (133)$$

$$(134)$$

$$\mathbf{J} = \begin{bmatrix} -l_{12} \cdot \text{sen} q_{12} - l_{22} \cdot \text{sen} (q_{12} + q_{22}) & -l_{22} \cdot \text{sen} (q_{12} + q_{22}) \\ l_{22} \cdot \cos q_{12} + l_{22} \cdot \cos (q_{12} + q_{22}) & l_{22} \cdot \cos (q_{12} + q_{22}) \end{bmatrix} \quad (135)$$

### 5.3.4 A cinemática inversa do robô B

$$\begin{bmatrix} q_{12} \\ q_{22} \end{bmatrix} = \begin{bmatrix} \tan^{-1}\left(\frac{y}{x}\right) - \tan^{-1}\left(\frac{l_{22} \cdot \text{sen} q_{22}}{l_{12} + l_{22} \cdot \cos q_{22}}\right) \\ \cos^{-1}\left(\frac{x^2 + y^2 - l_{12}^2 - l_{22}^2}{2 \cdot l_{12} \cdot l_{22}}\right) \end{bmatrix} \quad (136)$$

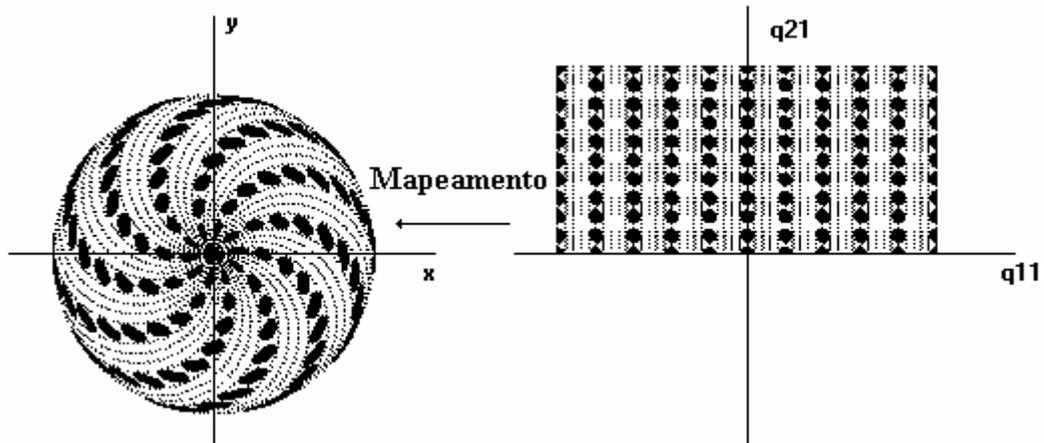
$$\begin{bmatrix} \dot{q}_{12} \\ \dot{q}_{22} \end{bmatrix} = \mathbf{J}_B^{-1} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \quad (137)$$

$$\begin{bmatrix} \dot{q}_{11} \\ \dot{q}_{21} \end{bmatrix} = \frac{1}{l_{11}l_{21}\text{sen}q_{21}} \underbrace{\begin{bmatrix} l_2 \cos(q_{11} + q_{21}) l_{21} \text{sen}(q_{11} + q_{21}) \\ -l_{11} \cos q_{11} - l_{21} \cos(q_{11} + q_{21}) & -l_{11} \text{sen} q_{11} - l_{21} \text{sen}(q_{11} + q_{21}) \end{bmatrix}}_{\mathbf{J}^{-1}} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \quad (138)$$

## 5.4 A Distribuição Normal Bidimensional

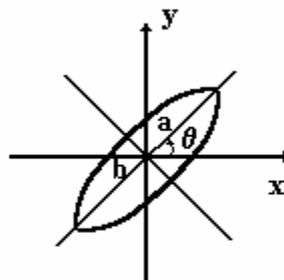
Para medir a manipulabilidade dos manipuladores robóticos, numa perspectiva numérica, testaram-se os movimentos do robô dentro de vários pontos na sua área de trabalho. Desta forma, consideram-se pequenos deslocamentos aleatórios do robô dentro de uma determinada esfera no espaço das junta, como se apresenta na figura 5.2.

Essa circunferência, no espaço das juntas, ao ser mapeada para o espaço operacional é transformada numa elipse. A manipulabilidade é determinada pela área da elipse que indica o factor de ampliação entre a juntas e a mão do manipulador robótico.



**Figura 5.2** – Movimentos aleatórios do robô com o cotovelo para cima no espaço das juntas  $\{O, q_1, q_2\}$  dentro de uma determinada esfera e respectivo mapeamento para o espaço operacional  $\{O, x, y\}$ .

Como já foi referido, a escolha do manipulador depende da tarefa a efectuar. Por esta razão, pretende-se criar um mapa no espaço operacional que indique, quantitativamente, a manipulabilidade do robô em cada ponto do espaço operacional.



**Figura 5.3** – Elipse calculada através do mapeamento para um robô.

A medição de uma elipse obtida numericamente no mapeamento, deve tomar em linha de conta as expressões derivada de seguida.

Sabe-se que a equação de uma elipse cujos eixos  $a$  e  $b$  estão rodados (no sentido positivo) de um ângulo  $\theta$  é dada por:

$$x^2(a^2S_0^2 + b^2C_0^2) + y^2(a^2C_0^2 + b^2S_0^2) - 2xy(a^2 - b^2)S_0C_0 = a^2b^2 \quad (139)$$

Para obter os parâmetros, a partir de uma amostra numérica, vai-se utilizar uma aproximação a partir da estatística. Assim, considera-se uma variável aleatória contínua bidimensional  $(x, y)$ , tomando todos os valores do plano euclidiano. A variável apresenta uma distribuição normal bidimensional se a sua função densidade probabilidade conjunta, for dada pela seguinte expressão:

$$f(x, y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} \times \text{EXP} \left\{ -\frac{1}{2(1-\rho^2)} \left[ \left( \frac{x-\mu_x}{\sigma_x} \right)^2 - 2\rho \frac{(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y} + \left( \frac{y-\mu_y}{\sigma_y} \right)^2 \right] \right\} \quad (140)$$

$$-\infty < x < +\infty \quad -\infty < y < +\infty \quad (141)$$

Devemos impor as seguintes restrições aos parâmetros:

$$-\infty < \mu_x < +\infty, \quad -\infty < \mu_y < +\infty, \quad \sigma_x > 0, \quad \sigma_y > 0, \quad -1 < \rho < 1 \quad (142)$$

Considerando que a superfície  $z = f(x, y)$ , onde  $f$  é a função densidade de probabilidade normal bidimensional é constante e corta a superfície numa elipse, obtêm-se.

$$\left( \frac{x - \mu_x}{\sigma_x} \right)^2 - 2\rho \frac{(x - \mu_x)(y - \mu_y)}{\sigma_x \sigma_y} + \left( \frac{y - \mu_y}{\sigma_y} \right)^2 = \text{Constante} \quad (143)$$

Fazendo a mudança de variável,  $\mu = x - \mu_x$ ,  $v = y - \mu_y$ , isto é, considerando uma elipse centrada na origem em vez de centrada no ponto  $(\mu_x, \mu_y)$ , vem:

$$\left( \frac{\mu}{\sigma_x} \right)^2 - 2\rho \left( \frac{\mu}{\sigma_x} \right) \left( \frac{v}{\sigma_y} \right) + \left( \frac{v}{\sigma_y} \right)^2 = C \quad (144)$$

$$\sigma_y^2 \mu^2 - 2\rho \sigma_x \sigma_y \mu v + \sigma_x^2 v^2 = C \quad (145)$$

Comparando as equações (142) e (145) resulta:

$$\begin{cases} \sigma_y = a^2 S_\theta^2 + b^2 C_\theta^2 \\ \sigma_x = a^2 C_\theta^2 + b^2 S_\theta^2 \\ \rho \sigma_x \sigma_y = (a^2 - b^2) S_\theta C_\theta \end{cases} \quad (146)$$

Ou seja:

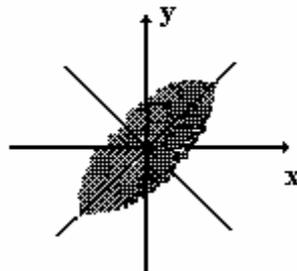
$$\theta = \frac{1}{2} \text{Arctg} \left( \frac{2\rho \sigma_x \sigma_y}{\sigma_x^2 - \sigma_y^2} \right) \quad (147)$$

$$a^2 = \frac{\sigma_x^2 C_\theta^2 - \sigma_y^2 S_\theta^2}{C_{2\theta}} \quad (148)$$

$$b^2 = \frac{\sigma_y^2 C_\theta^2 - \sigma_x^2 S_\theta^2}{C_{2\theta}} \quad (149)$$

Note-se que se  $\rho = 0$  e  $\sigma_x = \sigma_y$  então a equação (145) resulta  $\mu^2 + v^2 = C'^2$ , onde  $C'^2 = \frac{C}{\sigma}$ , ou seja, a equação de uma circunferência centrada no ponto  $(\mu_x, \mu_y)$  do plano Oxy.

Observa-se na figura 5.2 um conjunto de elipses no espaço operacional. Para um determinado ponto pode-se observar, na figura 5.4, uma elipse obtida numericamente, muito próxima da elipse “ideal” apresentada na figura 5.3. Isto verifica-se, porque o conjunto de amostras no espaço das juntas é relativamente elevado e porque os pontos calculados são bastante próximos uns dos outros.



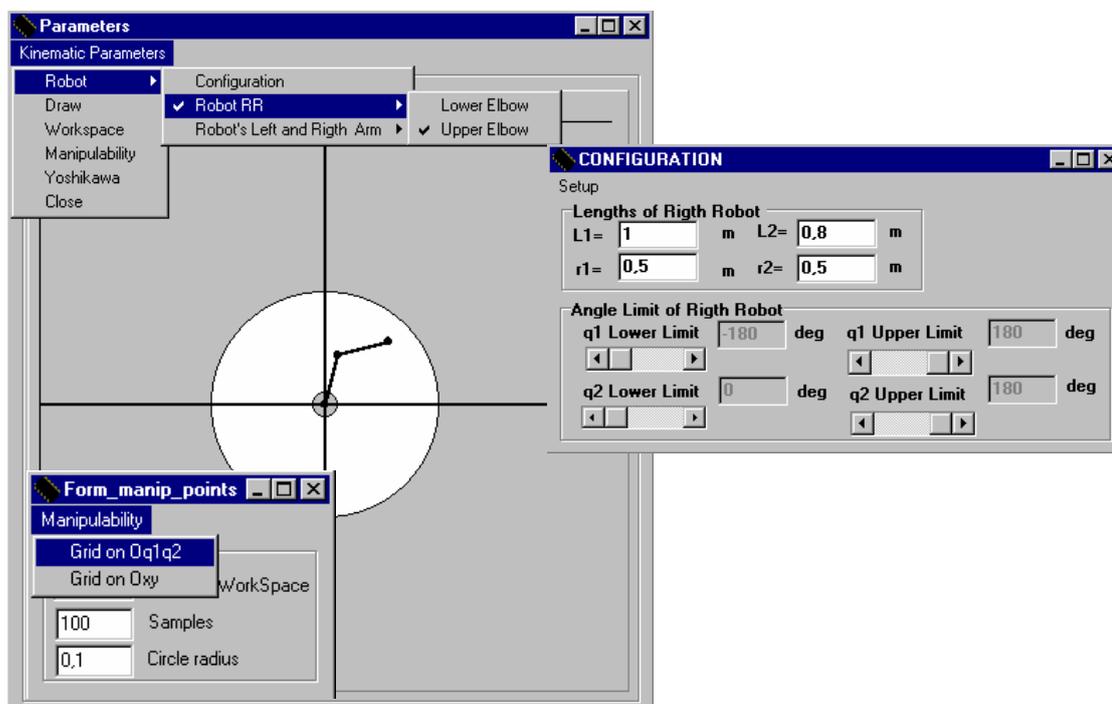
**Figura 5.4** – Elipse obtida numericamente através do mapeamento de um conjunto de pontos para um braço robô.

## 5.5 RobLib - Manipulabilidade

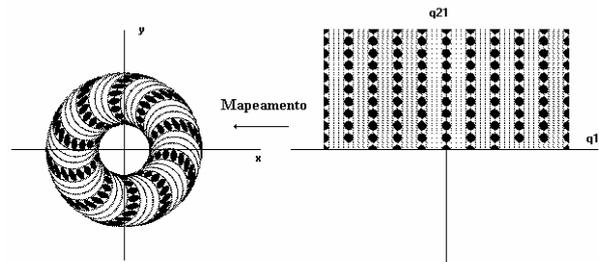
Desenvolveu-se no *RobLib*, uma ferramenta que permite o cálculo da manipulabilidade de um ou dois robôs com estrutura do tipo RR.

Na figura 5.5, apresentam-se as funcionalidade desta ferramenta. Permite-se a alteração da configuração e dos parâmetros de um, ou dos dois robôs, assim como a animação do robô, e o cálculo da manipulabilidade na sua área de trabalho.

Os resultados obtidos, são apresentados através de uma figura que representa a manipulabilidade no espaço de trabalho ( figura 5.6). Esses resultados são gravados para um ficheiro em formato texto, de modo a poderem ser importados para outras aplicações, tal como o Matlab. Deste modo, os dados podem ser apresentados gráficamente, através de curvas de nível, gráficos bidimensionais ou tridimensionais.

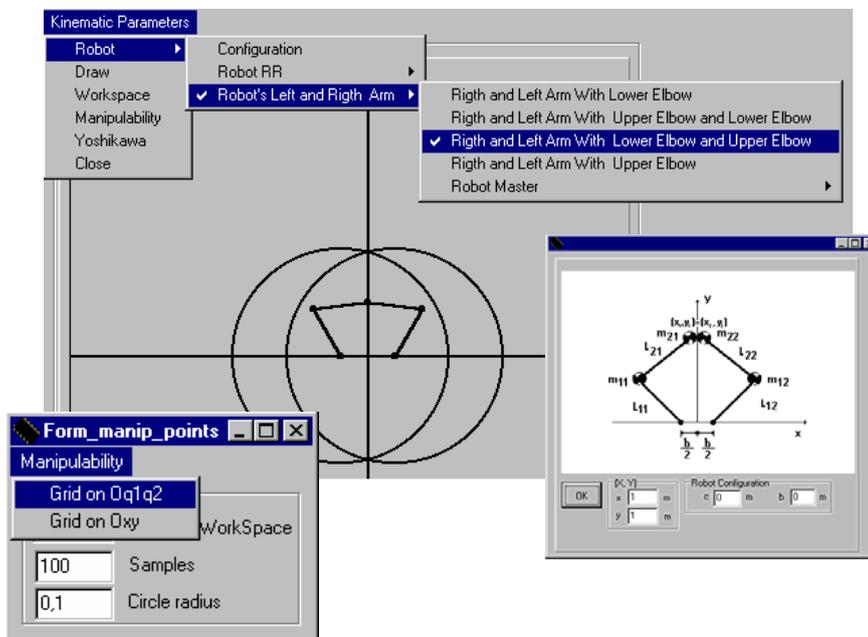


**Figura 5.5** – Janelas correspondentes ao robô RR, com a configuração pretendida, a selecção dos parâmetros do robô, número de pontos, número de amostras e a dimensão pretendida da esfera no espaço das juntas para o cálculo da manipulabilidade, e opção relativa à grelha de pontos num dos espaços.



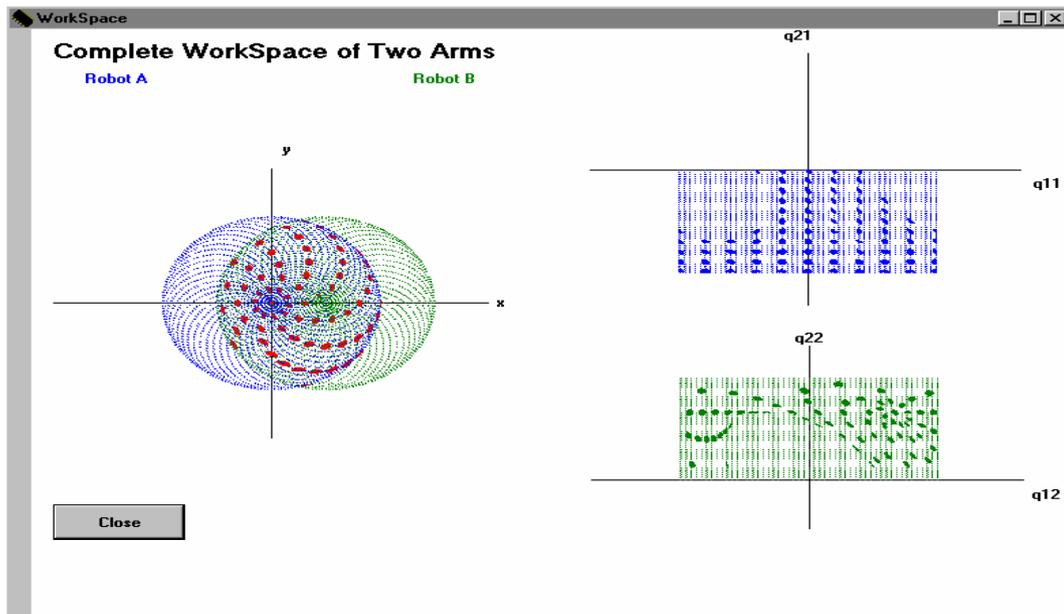
**Figura 5.6** – Resultados gráficos correspondentes do cálculo da manipulabilidade para o robô RR, com diferentes comprimentos dos elos.

Na figura seguinte, apresentam-se as configurações dos parâmetros de dois robôs para o cálculo da manipulabilidade “conjunta”, na sua área de trabalho.

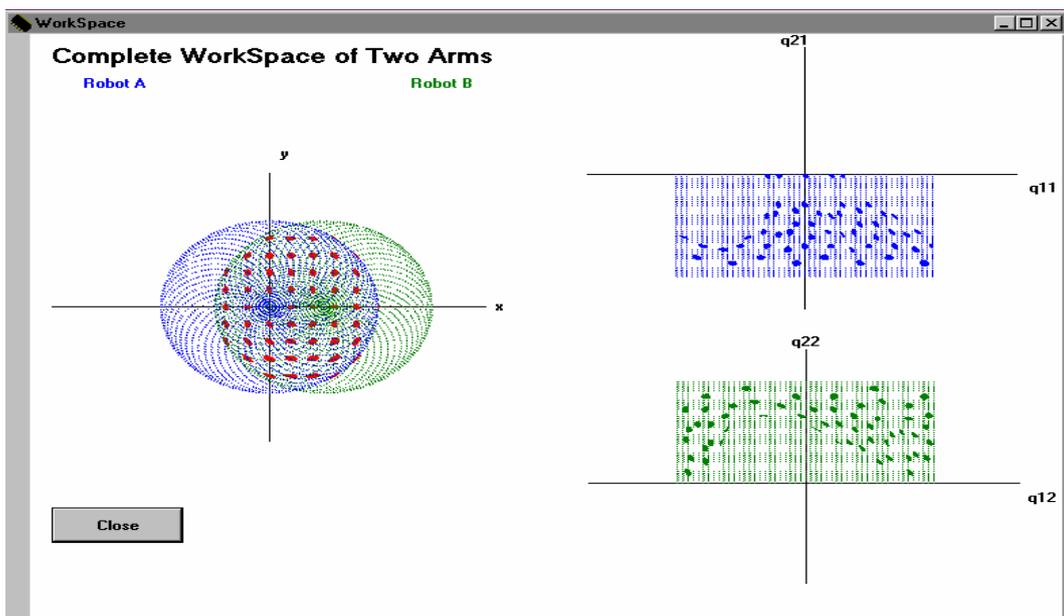


**Figura 5.7** – Janelas correspondentes a dois robôs com estrutura RR, com a configuração pretendida, a selecção dos parâmetros do robô, número de pontos, número de amostras e a dimensão pretendida da esfera no espaço das juntas para o cálculo da respectiva manipulabilidade, e opção relativa à grelha de pontos num dos espaços.

As figuras 5.8 e 5.9 mostram a grelha em dois espaços distintos.



**Figura 5.8** – Resultados gráficos correspondentes do cálculo da manipulabilidade para dois robôs RR, a uma distância de um metro, com a grelha no espaço das juntas.



**Figura 5.9** – Resultados gráficos correspondentes do cálculo da manipulabilidade para dois robôs RR, distanciados de um metro, com a grelha no espaço operacional.

## 5.6 Resultados Numéricos

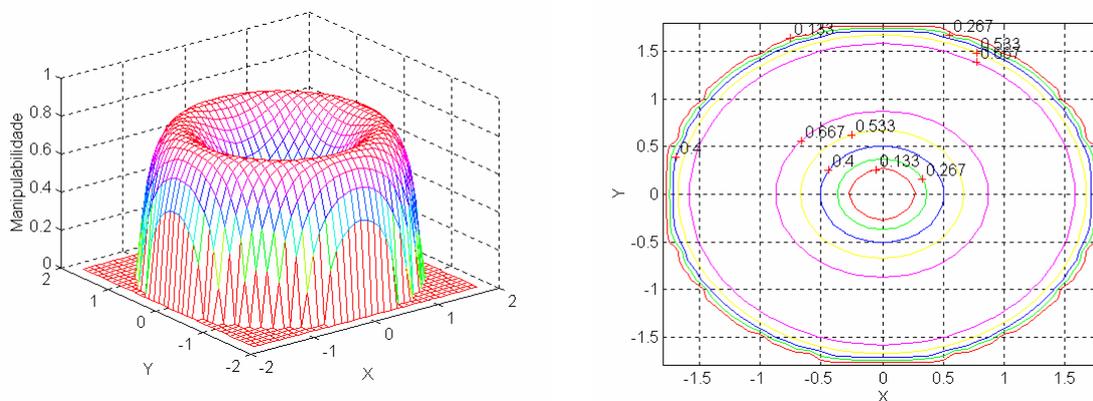
As experiências que se vão apresentar a seguir, consideram os casos de um e de dois robôs com estrutura RR e analisam a manipulabilidade do sistema no espaço operacional.

No primeiro conjunto de experiências, considera-se, somente, um robô. Neste caso, a experiência pretende determinar a medida de manipulabilidade do robô no espaço operacional, a configuração do robô correspondente à máxima manipulabilidade e comparar o novo método com os resultados obtidos por Yoshikawa.

No segundo conjunto de experiências, consideram-se dois robôs a trabalhar em cooperação. Neste caso, pretende-se determinar a medida de manipulabilidade “conjunta” e a configuração que conduz a um valor máximo.

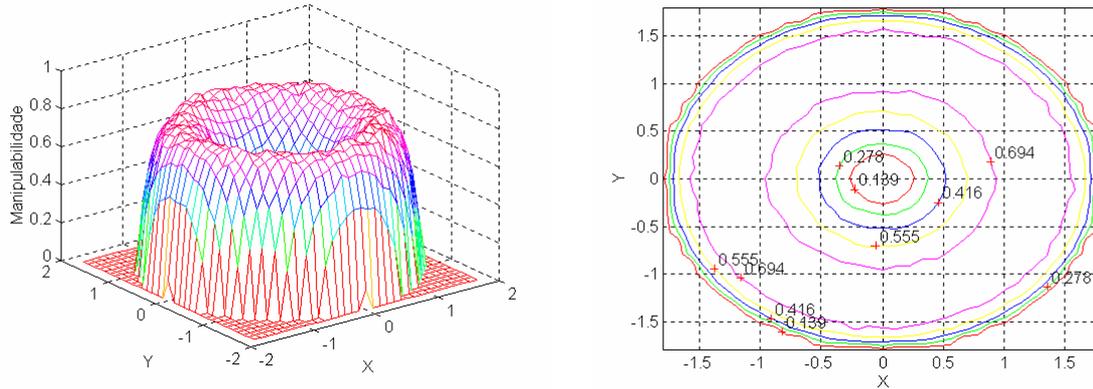
### 5.6.1 Um Robô RR

Para um robô RR, efectuou-se um conjunto de simulações teste, de modo a comparar o método de Yoshikawa, com o método numérico utilizado.

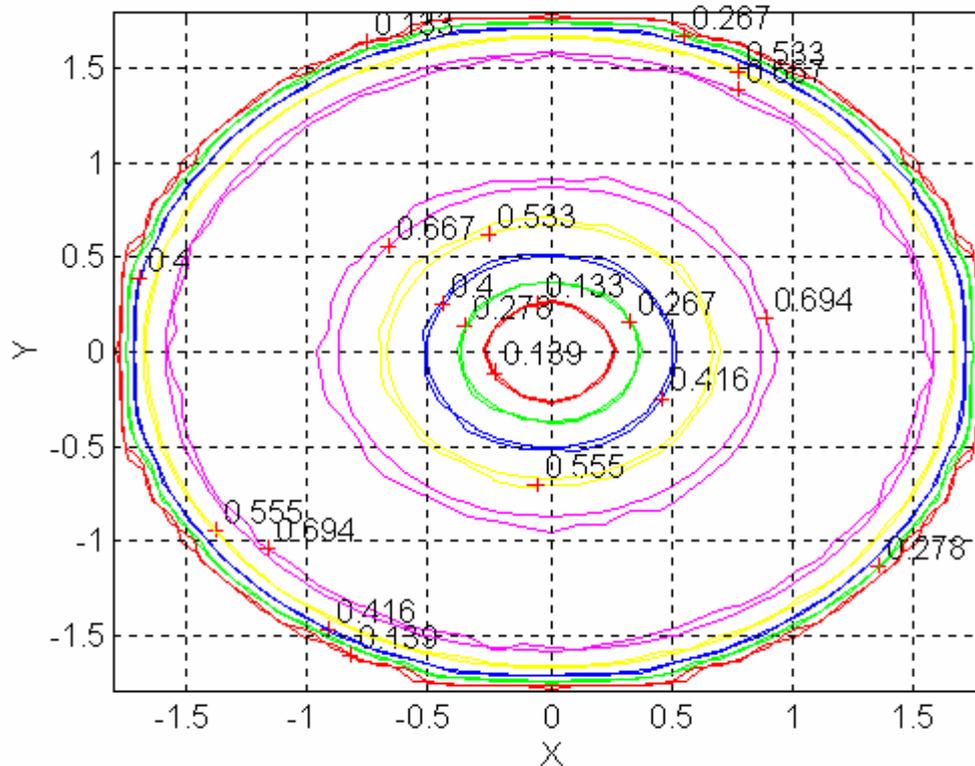


**Figura 5.10** – Manipulabilidade do robô RR com  $l_1=1$  e  $l_2=0,8$ , obtida pelo método de Yoshikawa.

As figuras 5.10 e 5.11 mostram a manipulabilidade, no espaço operacional, obtida pelos dois métodos alternativos e a figura 5.12 estabelece uma comparação entre os resultados.



**Figura 5.11** – Manipulabilidade do robô RR com  $l_1=1$  e  $l_2=0,8$ , obtida pelo método numérico, para mil pontos distintos, no espaço operacional do robô.

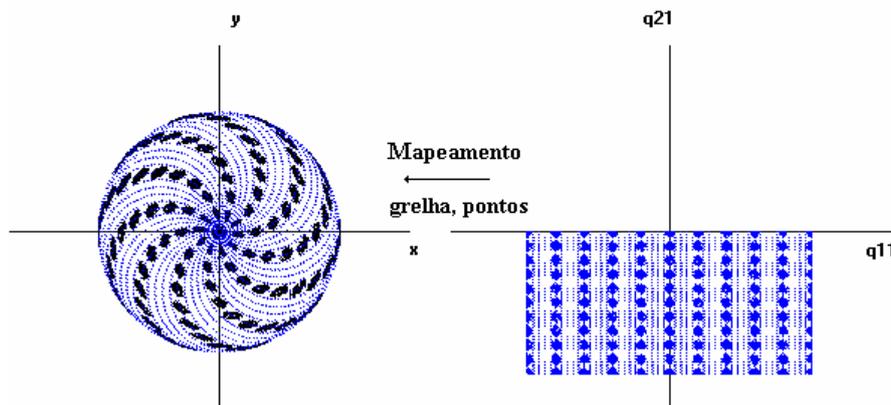


**Figura 5.12** –Curvas de nível sobrepondo os resultados de Yoshikawa/Método numérico.

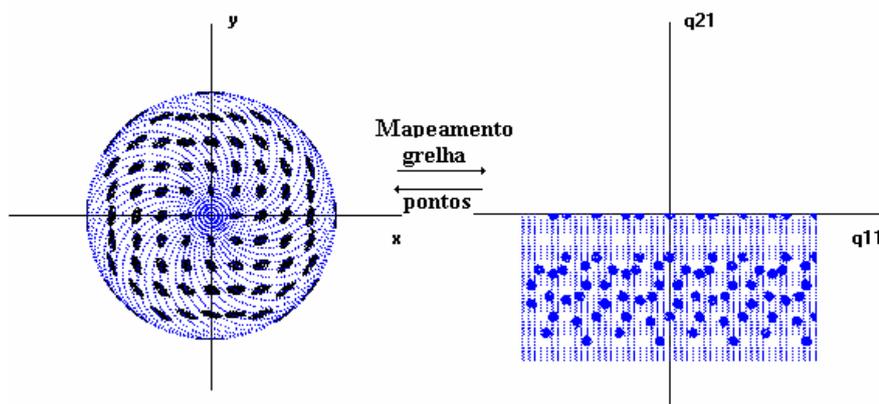
Como se observa, na figura 5.12, o método numérico apresenta um erro, relativamente, baixo face ao método analítico de Yoshikawa. Além disso, o novo algoritmo é aceitável, computacionalmente, porque se considerou para cada esfera do espaço das juntas, mil pontos simulados, o que se traduz numa razoável rapidez. Para diminuir o erro numérico, é necessário aumentar o número de amostras para cada esfera,

mas o tempo para o cálculo da manipulabilidade no espaço de trabalho aumenta proporcionalmente.

As figuras 5.13 e 5.14 representam duas formas para obter a distribuição em  $\{O,x,y\}$  da medida de manipulabilidade através do programa *Roblib*. Neste estudo, optou-se por efectuar a grelha de pontos no espaço operacional  $\{O,x,y\}$ , porque os resultados pretendidos são para espaço operacional  $\{O,x,y\}$  e o método de colocar os dados para o Matlab é mais expedito.

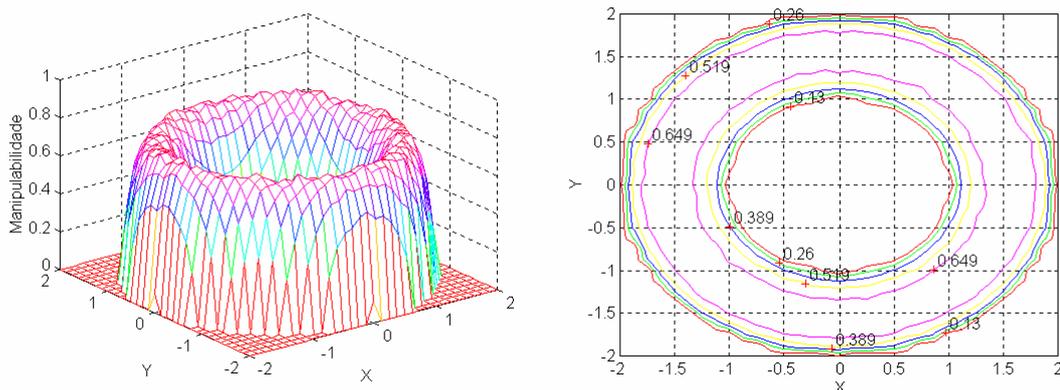


**Figura 5.13** – Movimentos aleatórios do robô no espaço das juntas  $\{O,q_1,q_2\}$  dentro de uma pequena esfera e respectivo mapeamento para o espaço operacional  $\{O,x,y\}$ . Robô com o cotovelo em baixo e grelha de pontos no espaço das juntas  $\{O,q_1,q_2\}$ .

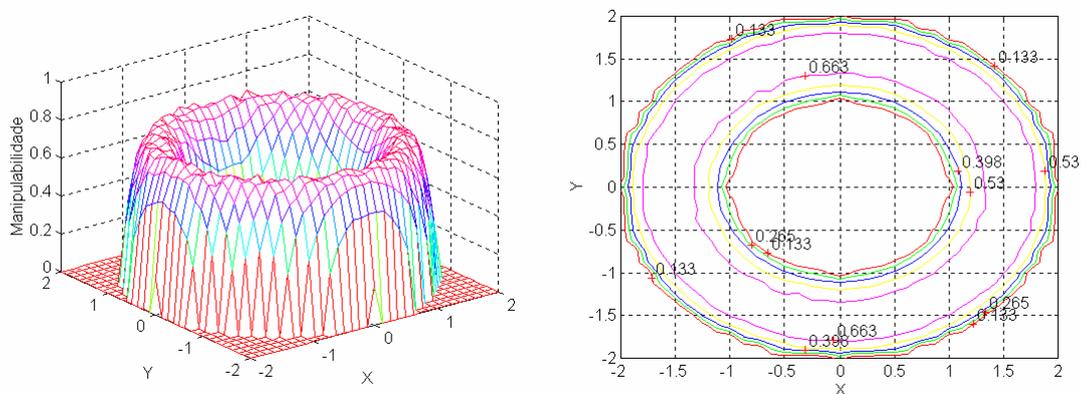


**Figura 5.14** – Movimentos aleatórios do robô no espaço das juntas  $\{O,q_1,q_2\}$  dentro de uma pequena esfera e respectivo mapeamento para o espaço operacional  $\{O,x,y\}$ . Robô com o cotovelo em baixo e grelha de pontos no espaço operacional  $\{O,x,y\}$ .

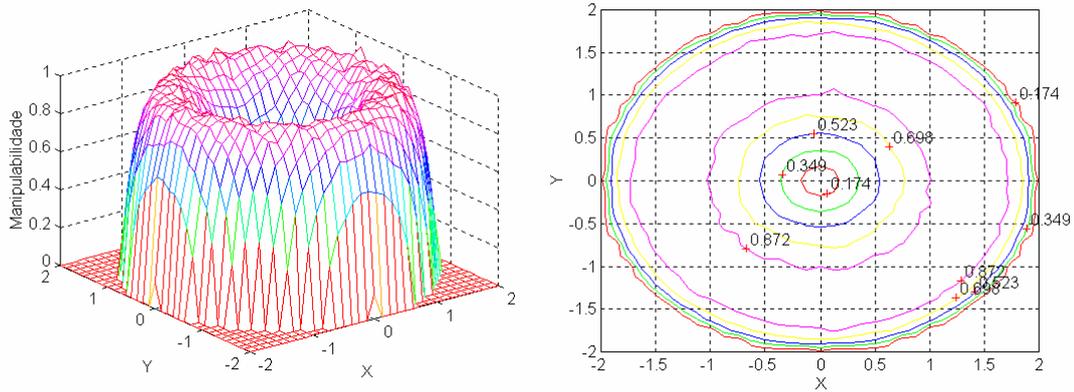
No primeiro conjunto de experiências (figuras 5.15 a 5.17) que se efectuou para o robô RR, fizeram-se três simulações considerando o comprimento total constante (*i.e.*  $l_1+l_2=Constante$ ) de modo, a obter a medida de manipulabilidade do robô no seu espaço operacional e determinar qual a configuração do que confere a máxima manipulabilidade. Adoptou-se uma circunferência no espaço das juntas com raio de 0.1 rad.



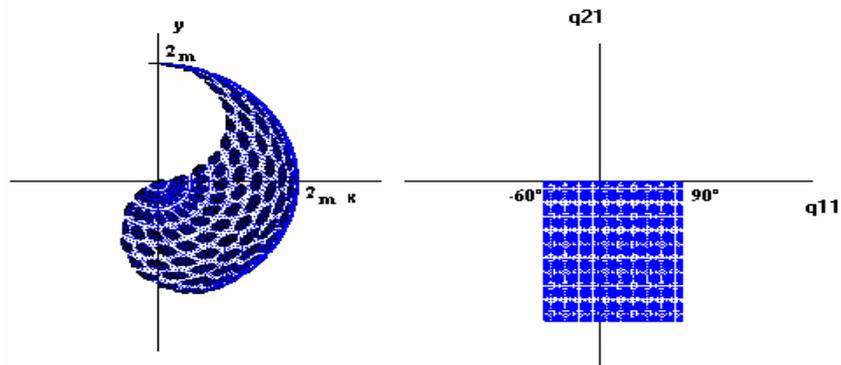
**Figura 5.15** – Gráfico tridimensional e curvas de nível para o robô RR com os parâmetros  $l_1=1.5$  e  $l_2=0,5$  para uma amostra de mil pontos no espaço operacional do robô.



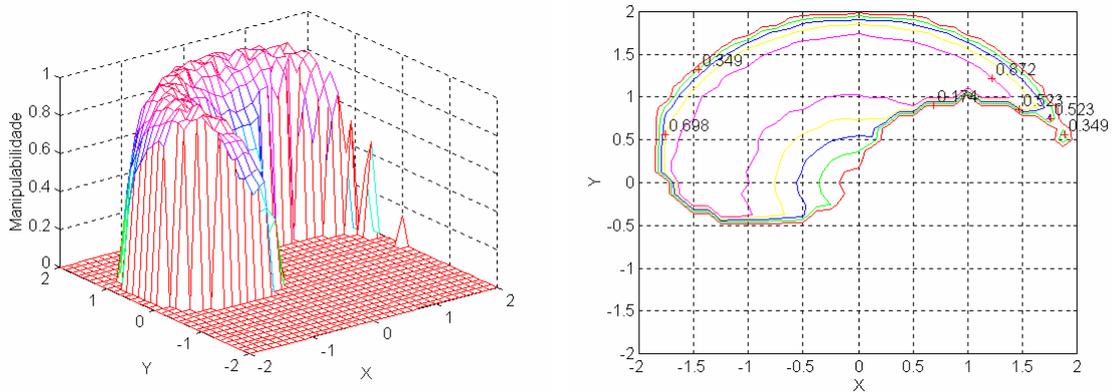
**Figura 5.16** – Gráfico tridimensional e curvas de nível para o robô RR com os parâmetros  $l_1=0.5$  e  $l_2=1,5$  para uma amostra de mil pontos no espaço operacional do robô.



**Figura 5.17** – Gráfico tridimensional e curvas de nível para o robô RR com os parâmetros  $l_1=1$  e  $l_2=1$ , para uma amostra de mil pontos, no espaço operacional do robô.



**Figura 5.18** – Mapa de elipses de manipulabilidade para um robô RR com  $l_1=1$  e  $l_2=1$  com limitações nas juntas  $-\frac{\pi}{3} < q_1 < \frac{\pi}{2}$  e  $-\pi < q_2 < 0$ , para mil pontos distintos, no espaço operacional do robô.



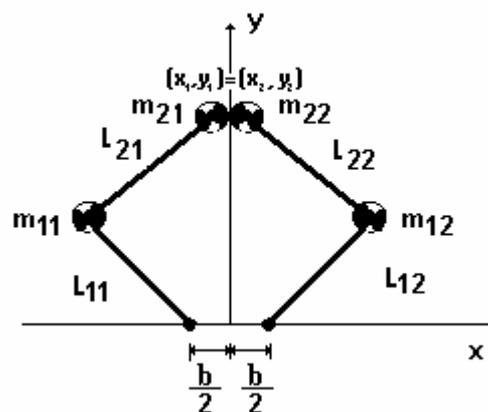
**Figura 5.19** – Gráfico tridimensional e curvas de nível para o robô RR com os parâmetros  $l_1=1$  e  $l_2=$  com limitações nas juntas  $-\frac{\pi}{3} < q_1 < \frac{\pi}{2}$  e  $-\pi < q_2 < 0$ , para uma amostra de mil pontos, no espaço operacional do robô .

Constata-se, a partir das figuras, que para comprimentos idênticos dos elos resulta a máxima manipulabilidade, o que confere com o resultado obtido por, Yoshikava, segundo a formula da manipulabilidade (126). Além disso, nota-se também, que o volume médio é maior  $l_1 = l_2$ , visto que, a “cavidade” no interior é menor do que  $l_1 \neq l_2$ .

Deve, ainda, referir-se nas (figuras 5.18 e 5.19) que o RobLib, também permite efectuar o estudo da manipulabilidade considerando limitações nos eixos.

### 5.6.2 Simulações de dois robôs a trabalhar em cooperação

Nas simulações seguintes, consideraram-se dois robôs a trabalhar em cooperação, manipulando objectos muito pequenos como se apresenta na figura 5.20. Os parâmetros do robô A são iguais aos do robô B e a distância entre as mãos dos robôs é nula.



**Figura 5.20** – Dois robôs RR a trabalhar em cooperação.

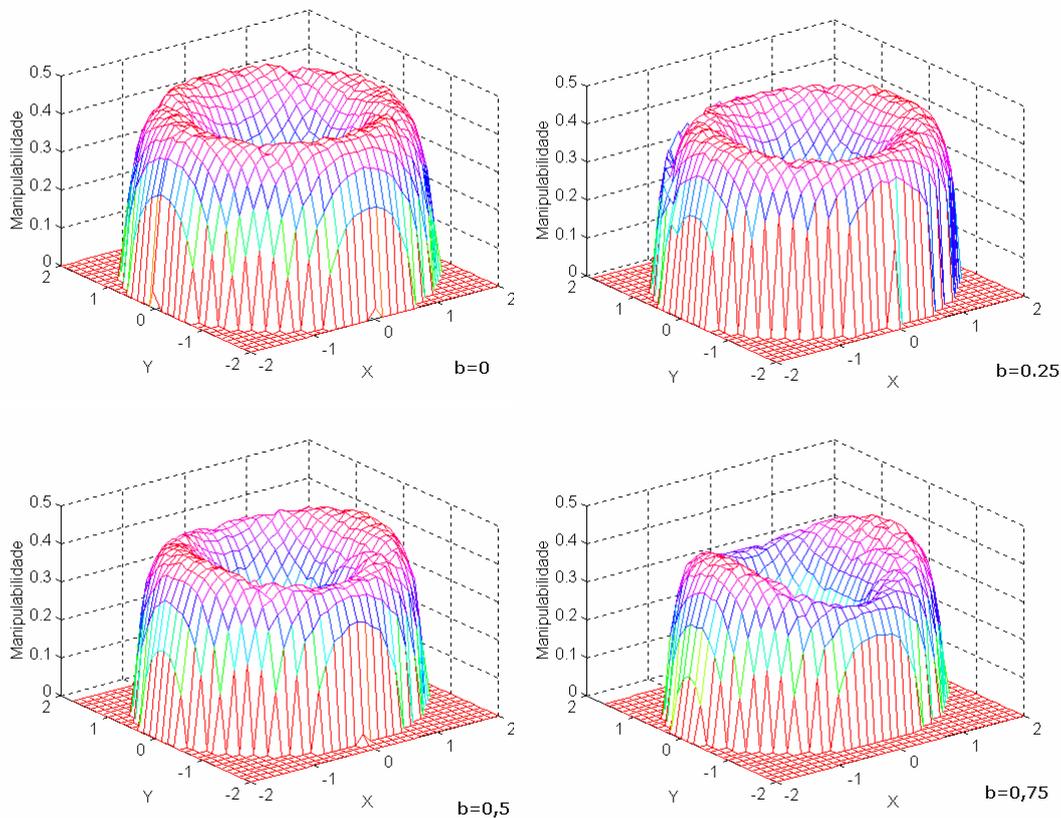
Desta forma, efectuaram-se diversas experiências para obter uma medida de manipulabilidade conjunta dos braços, no seu espaço de trabalho. No primeiro caso, considerou-se uma distância entre os braços  $b=0$  e, de seguida, aumentou-se até ao valor máximo, que é:  $b=2 \times (l_1 + l_2)$ .

Dada a redundância cinemática dos sistemas de dois braços, para cada ponto simulado, considerou-se que, alternadamente, o braço esquerdo e o braço direito do robô definiam a posição conjunta das mãos. Considerou-se uma grelha, de mil pontos no espaço operacional e, para cada ponto, efectuaram-se mil amostras no interior de uma esfera com um raio de 0.1rad no espaço das juntas.

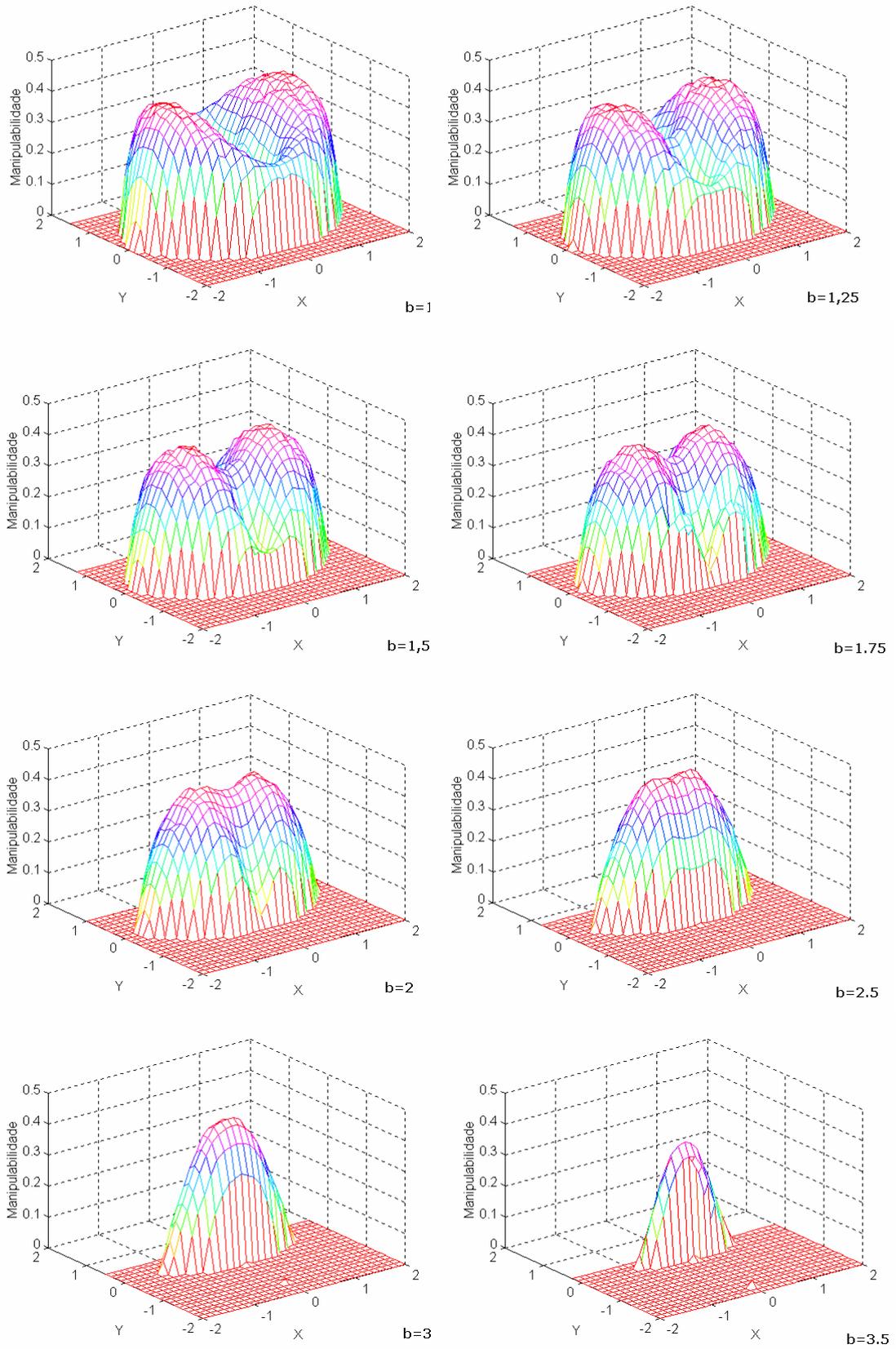
O conjunto de pontos simulados é mapeado para o espaço operacional, obtendo-se um conjunto de elipses, que permite obter uma medida de manipulabilidade no espaço de trabalho dos dois robôs.

Nesta perspectiva, as simulações seguintes permitem determinar qual a configuração dos robôs e a que distância devem estar os dois braços para se obter a máxima manipulabilidade conjunta.

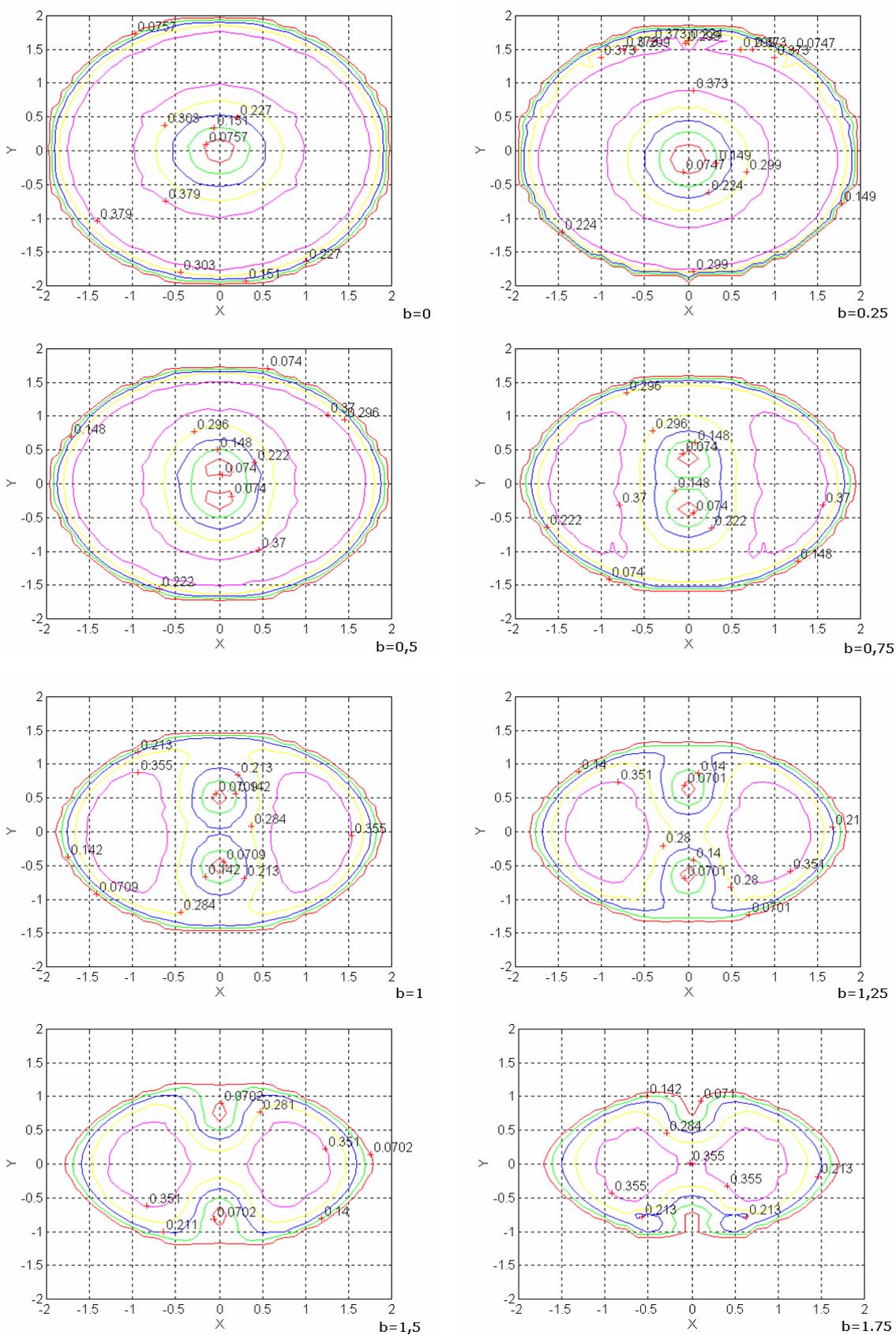
Os gráficos das figuras 5.21 a 5.24 apresentam a manipulabilidade no espaço de trabalho de dois robôs a trabalhar em cooperação com  $l_1 = l_2$ . Para cada simulação efectuada, consideram-se os mesmos parâmetros do robô A e B, alterando apenas a distância  $b$  entre os robôs com  $b \in [0, 4[$ .



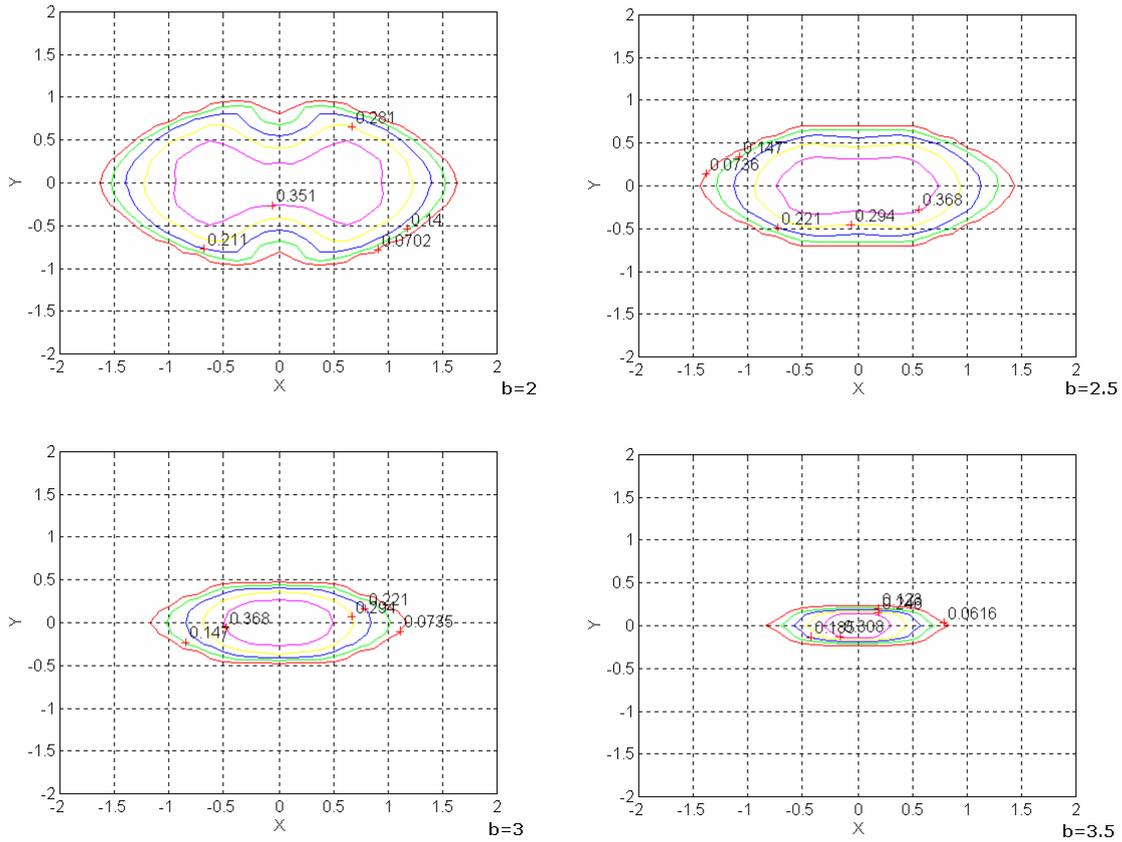
**Figura 5.21** – Medida de manipulabilidade conjunta de dois robôs RR no espaço operacional para vários valores de  $b \in [0, 4[$ .



**Figura 5.22** – Medida de manipulabilidade conjunta de dois robôs RR no espaço operacional para vários valores de  $b \in [0, 4[$ .

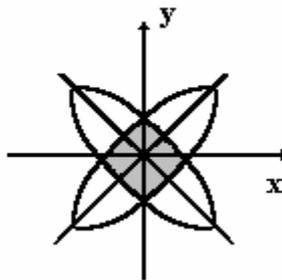


**Figura 5.23** – Curvas de nível da medida de manipulabilidade de dois robôs RR do espaço operacional de  $b \in [0, 4[$ .



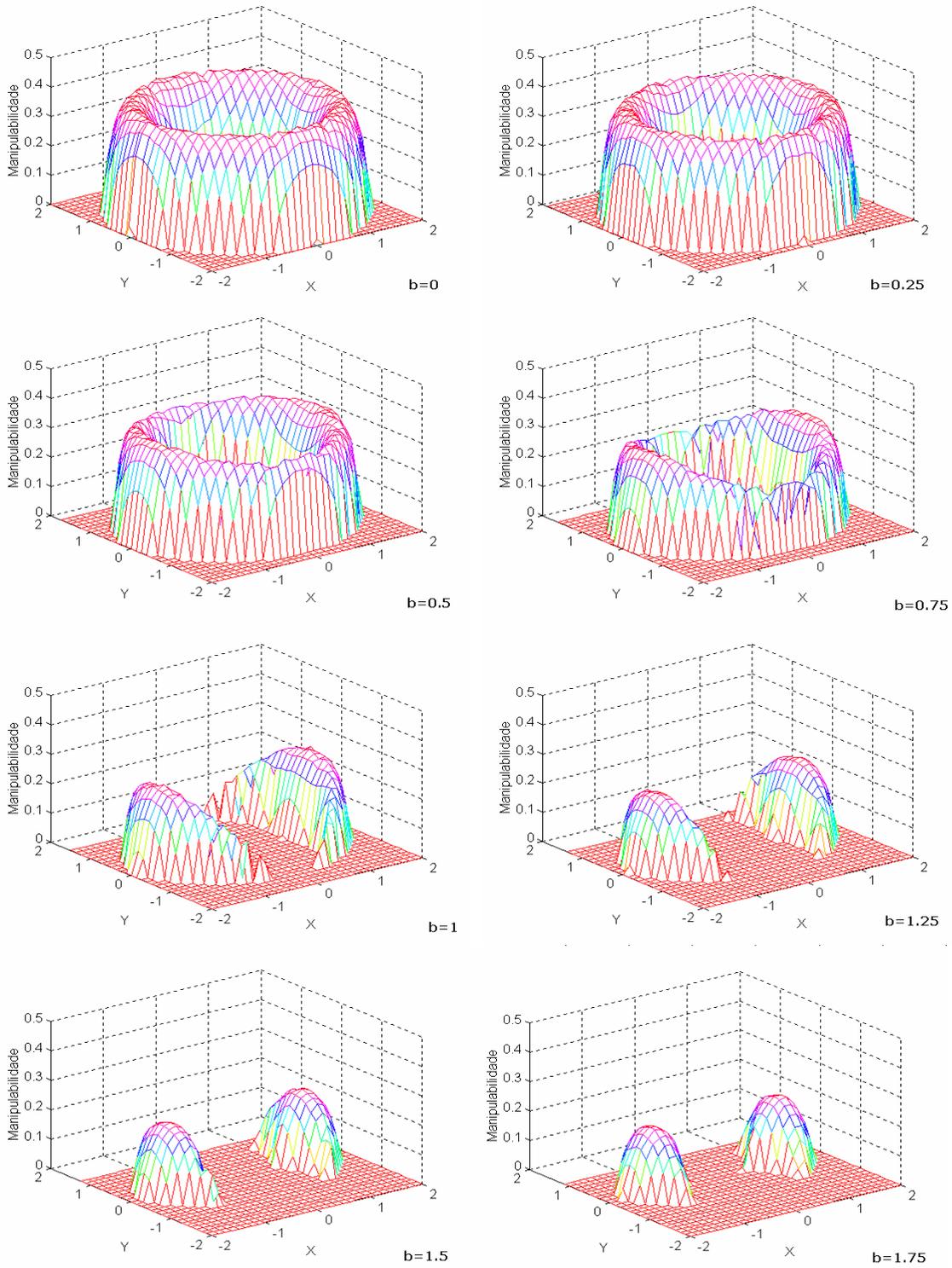
**Figura 5.24** – Curvas de nível da medida de manipulabilidade de dois robôs RR do espaço operacional de  $b \in [0, 4[$ .

Como se pode observar, o valor da manipulabilidade diminuiu em relação a um só braço robô. Este resultado era de esperar visto que a manipulabilidade de dois robôs é calculada da mesma forma que para um só robô mas tendo em conta que, neste caso, se tem uma área da elipse que é uma espécie de intersecção das duas áreas de cada robô, como se pode observar na figura 5.25.



**Figura 5.25** – Elipse calculada através do mapeamento para dois braços.

Os gráficos seguintes apresentam a manipulabilidade considerando que os parâmetros do robô A e B são  $l_1=1,5$  e  $l_2=0,5$ , respectivamente.



**Figura 5.26** – Medida de manipulabilidade do espaço operacional a 3D de  $b \in [0, 4]$ .

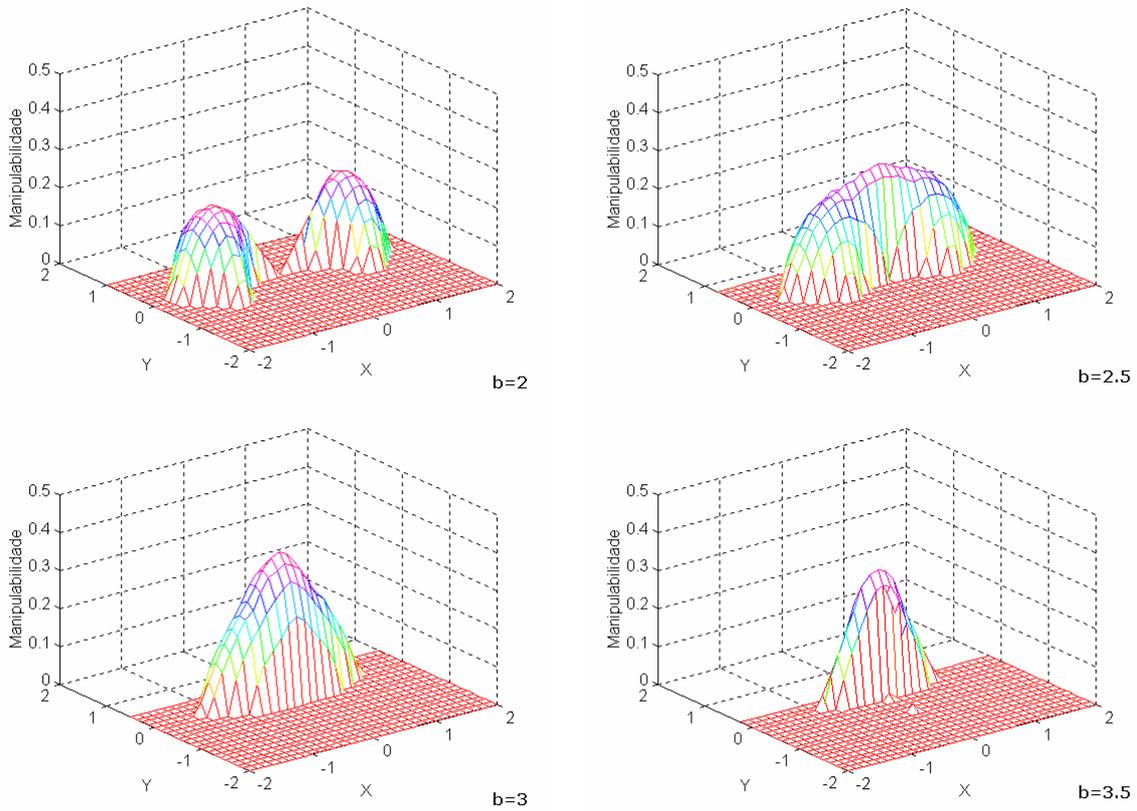


Figura 5.27 – Medida de manipulabilidade do espaço operacional a 3D de  $b \in [0, 4[$ .

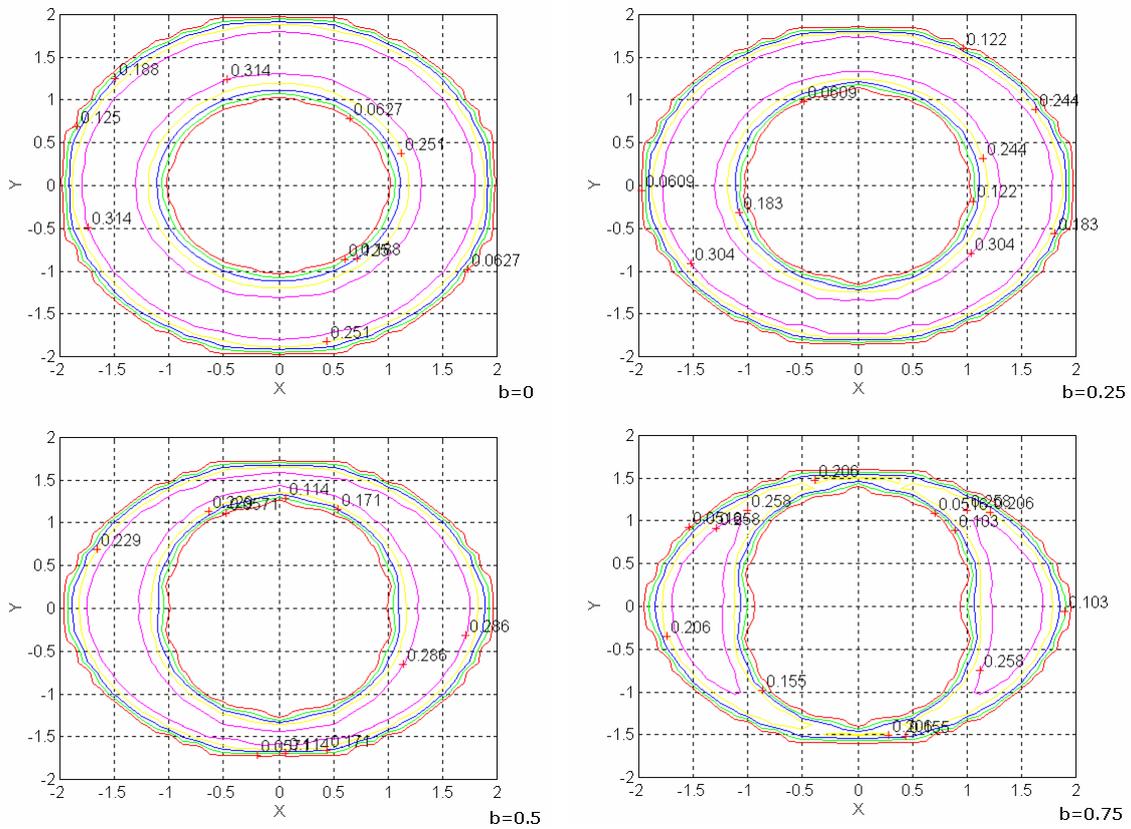
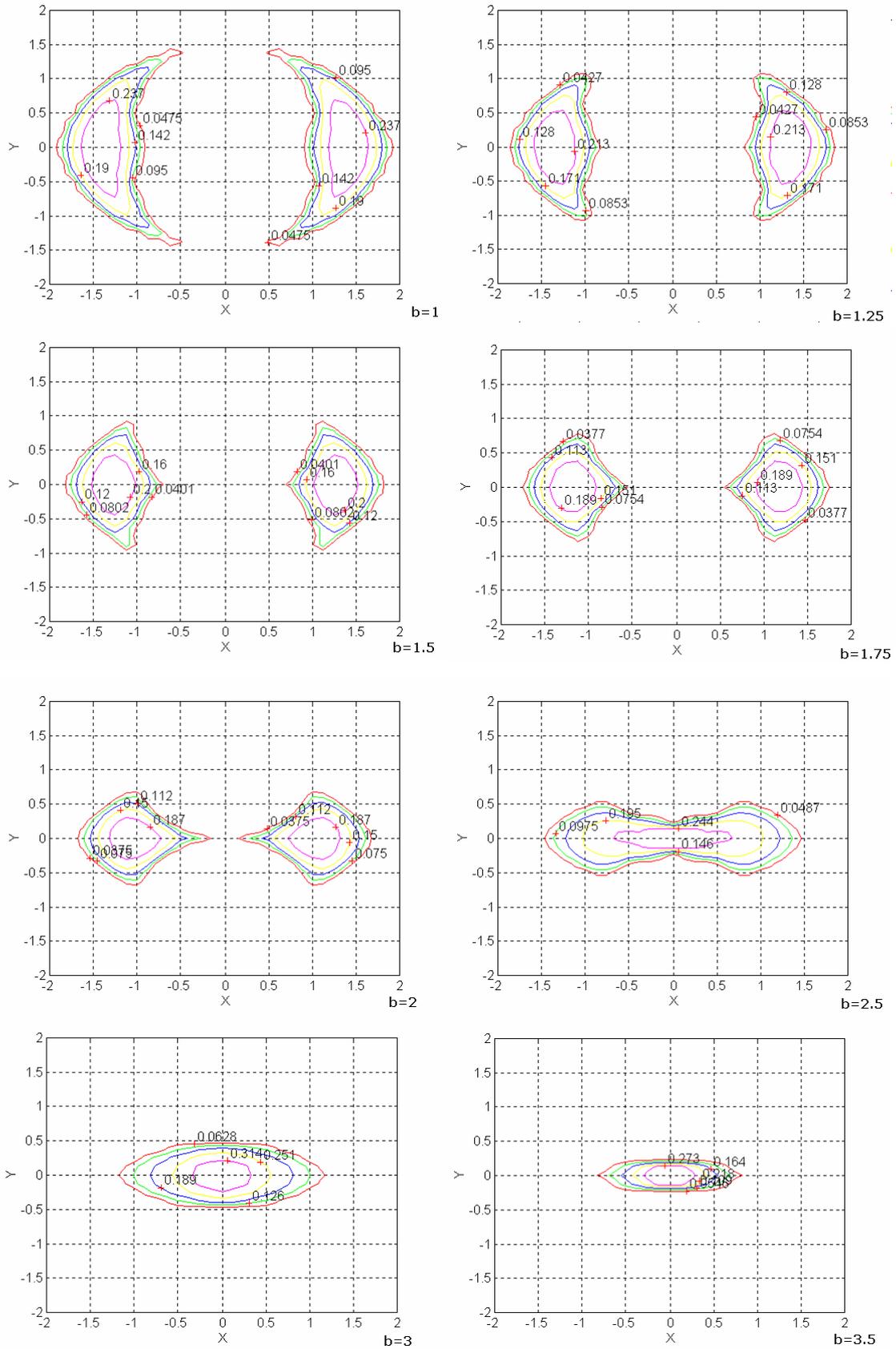
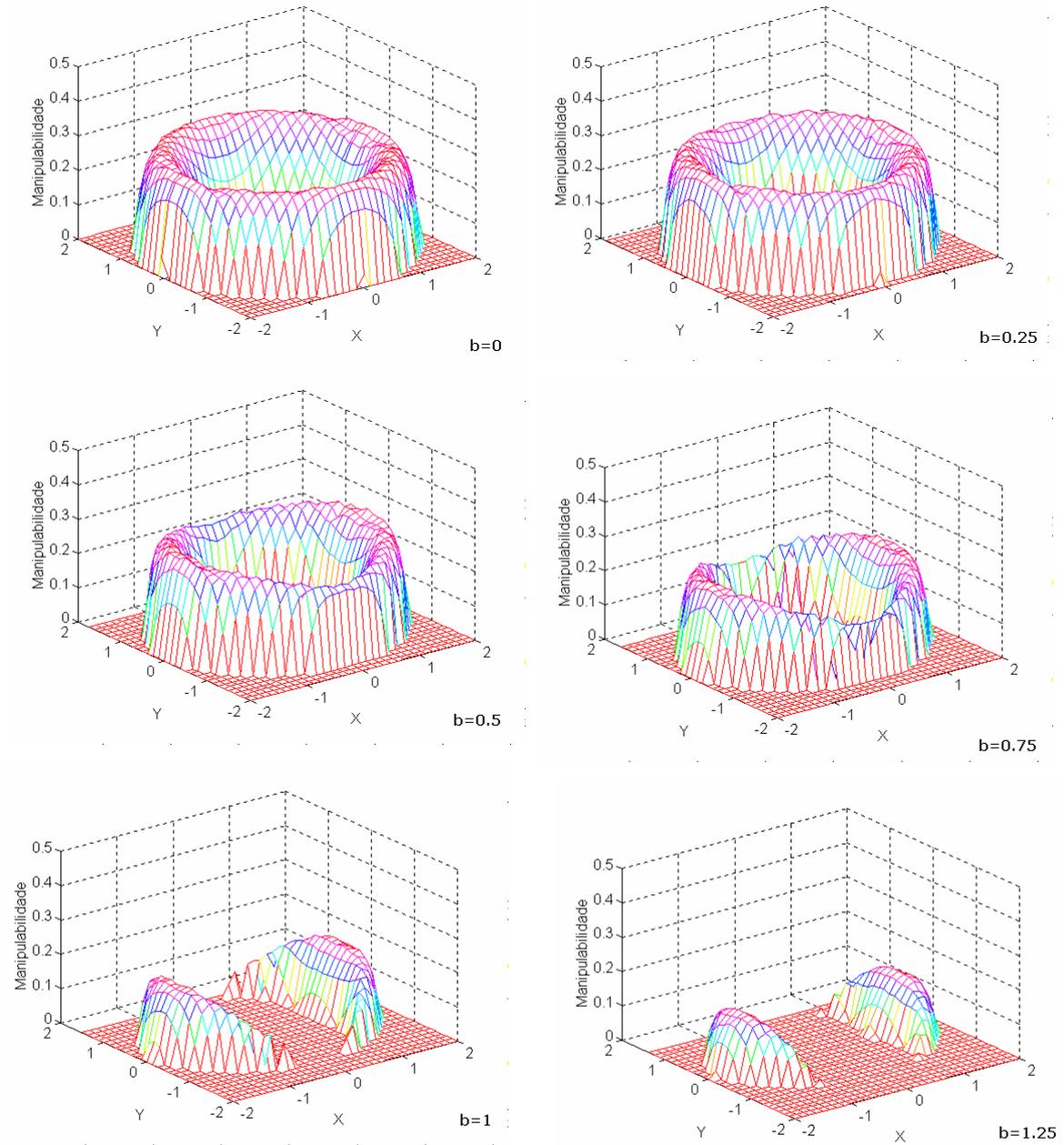


Figura 5.28 – Curvas de nível da medida de manipulabilidade de dois robôs RR do espaço operacional de  $b \in [0, 4[$ .

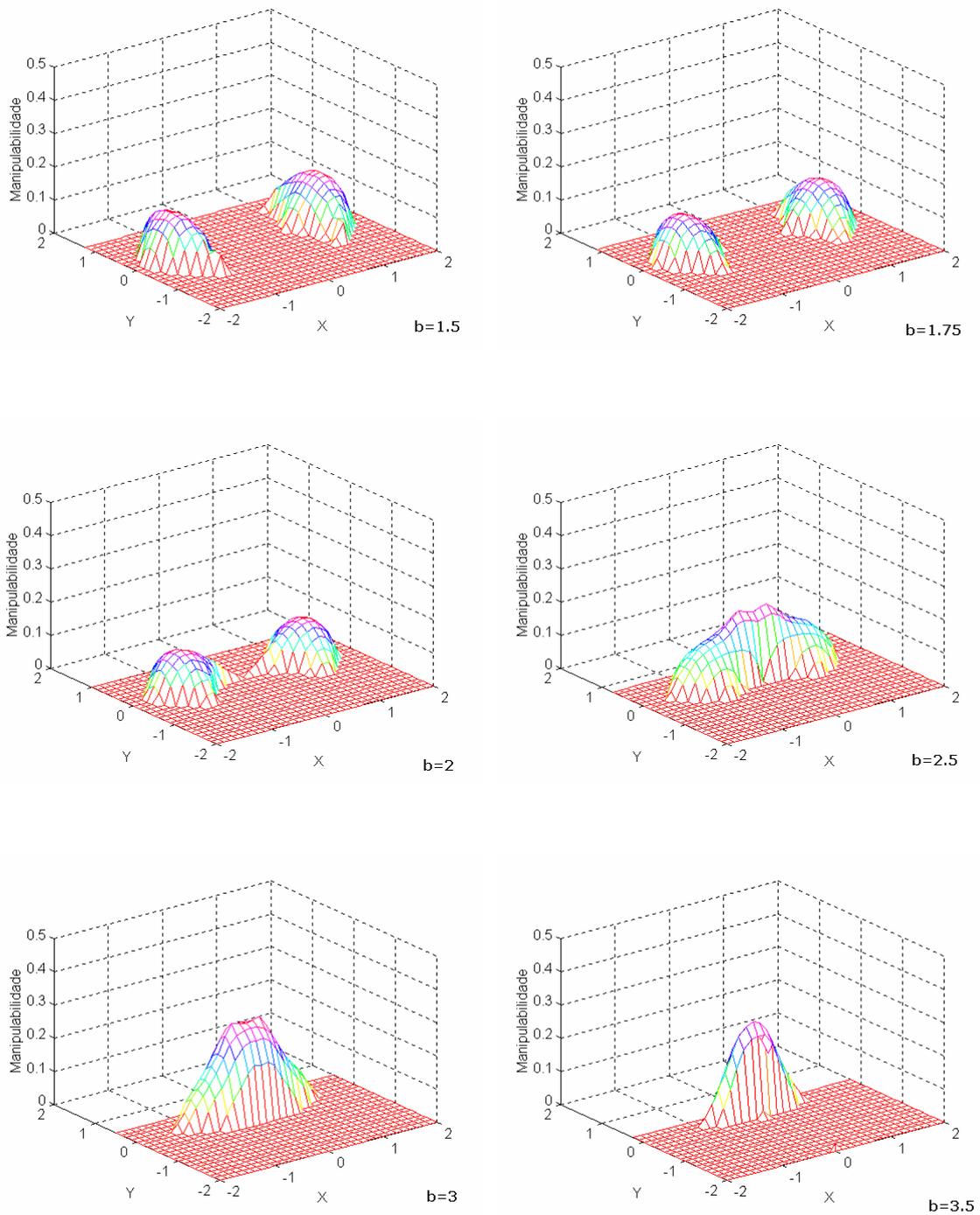


**Figura 5.29** – Curvas de nível da medida de manipulabilidade de dois robôs RR do espaço operacional de  $b \in [0, 4[$ .

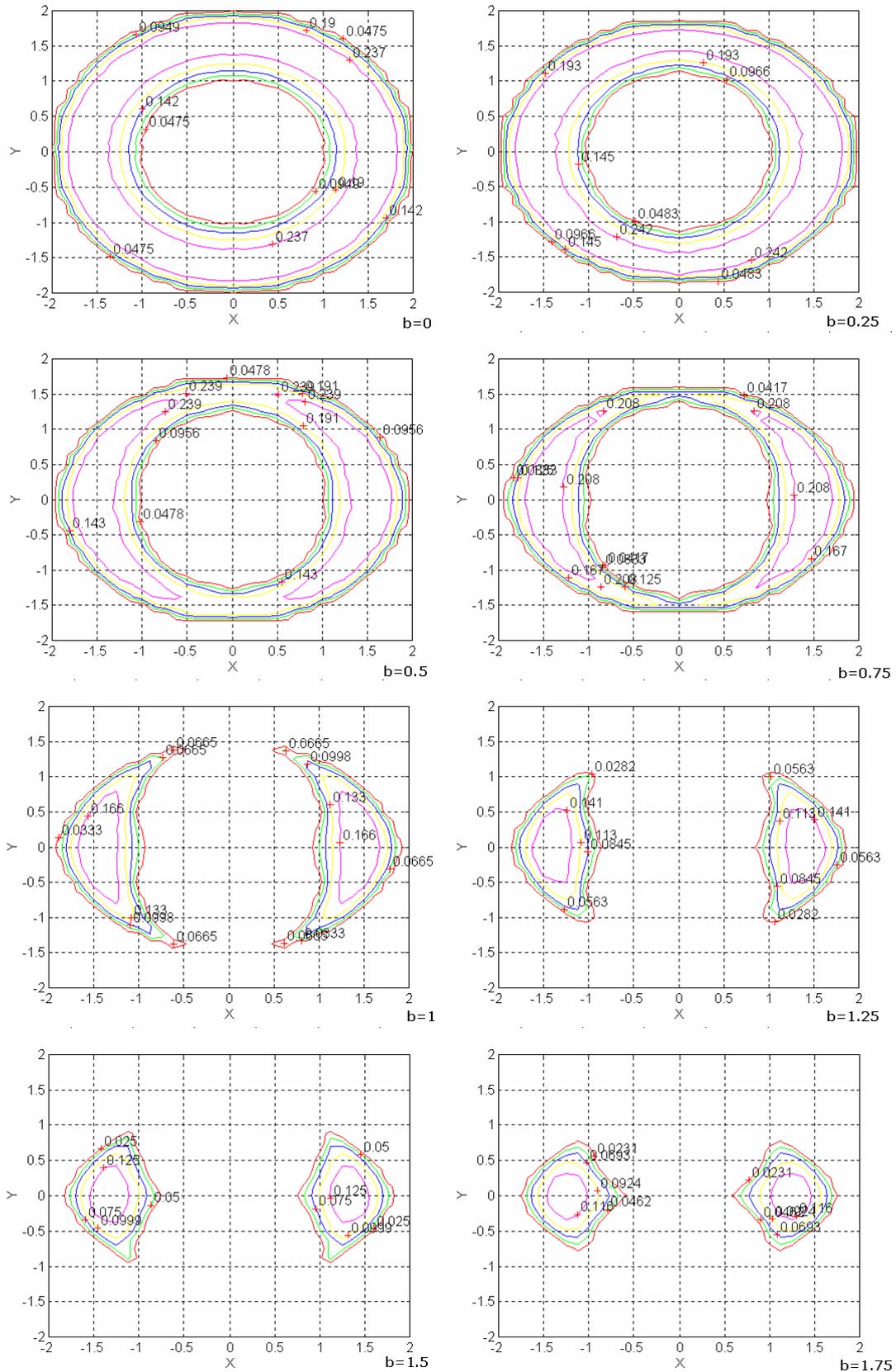
Os gráficos seguintes apresentam a manipulabilidade considerando os parâmetros do robô A e B são :  $l_1=0,5$  e  $l_2=1$ , respectivamente.



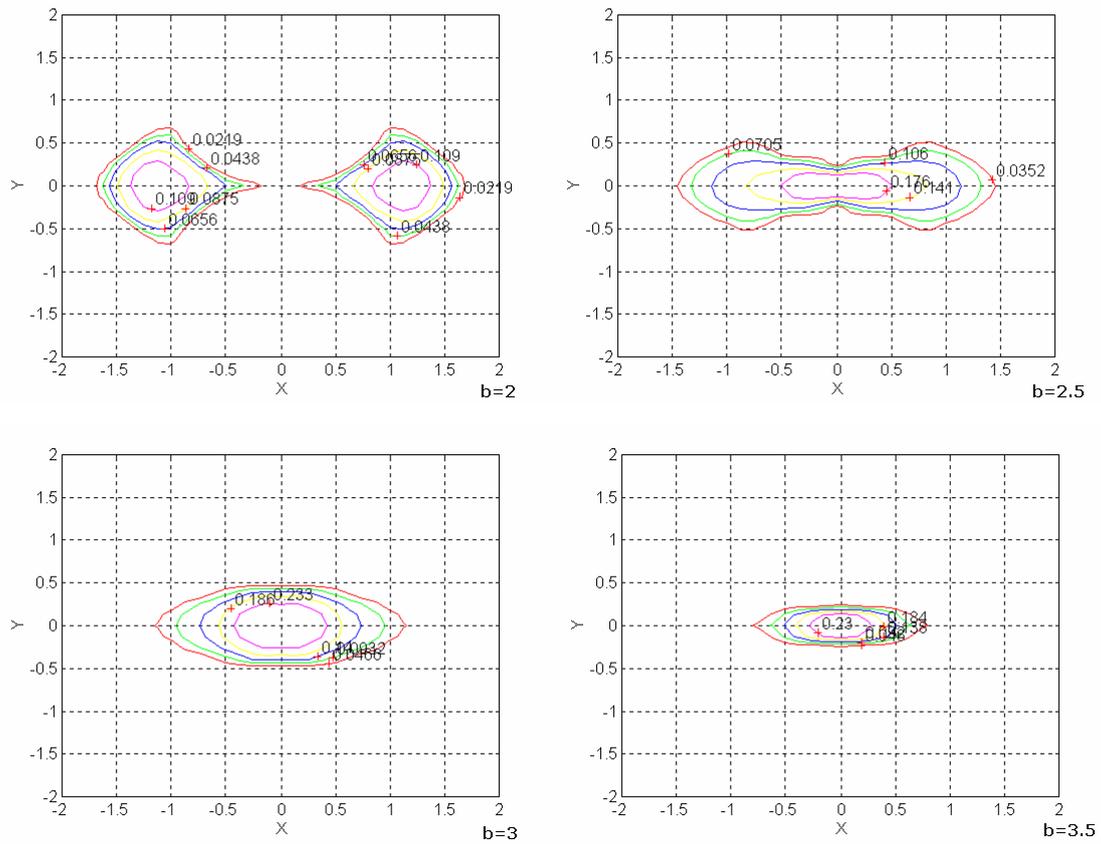
**Figura 5.30** – Medida de manipulabilidade conjunta de dois robôs RR no espaço operacional para vários valores de  $b \in [0, 4[$ .



**Figura 5.31** – Medida de manipulabilidade conjunta de dois robôs RR no espaço operacional para vários valores de  $b \in [0, 4[$ .



**Figura 5.32** – Curvas de nível da medida de manipulabilidade de dois robôs RR do espaço operacional de  $b \in [0, 4[$ .

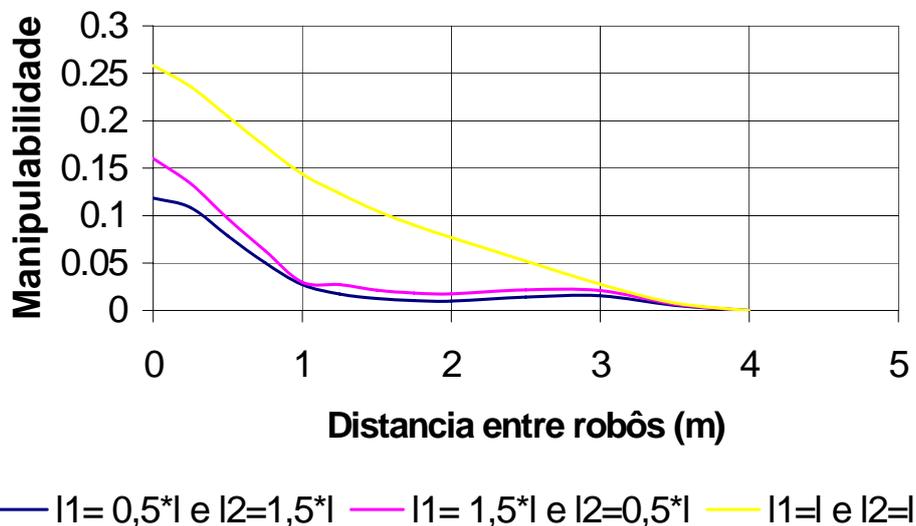


**Figura 5.33** – Curvas de nível da medida de manipulabilidade de dois robôs RR do espaço operacional de  $b \in [0, 4[$ .

Os gráficos que se apresentam, nas figuras 5.34 a 5.36, mostram a manipulabilidade média, a manipulabilidade máxima e o volume médio, em função da distância  $b$  entre braços. Observa-se, na figura 5.34, que se obtém uma maior manipulabilidade média para distâncias perto  $b=0$ , já que a área de trabalho é máxima, nesse caso.

Na figura 5.35, observa-se que (na manipulação de pequenos objectos), a manipulabilidade máxima encontra-se para distâncias perto de  $b=0$  e para distâncias próximas de  $b=3$ . Comparando as figuras 5.34 e 5.35 pode-se afirmar que, para manipular pequenos objectos, a distância entre braços deverá ser próxima de zero.

Verifica-se que para valores  $l_1=l_2$ , se obtém uma maior manipulabilidade do que para  $l_1 \neq l_2$ .



**Figura 5.34** – Manipulabilidade média para a base considerada.

Por outro lado, a figura 5.36 apresenta o volume médio em função da distância  $b$ . Verifica-se, nesta figura e na figura 5.35, que para valores próximos de  $b=2$  (*i.e.* o comprimento total do braço) é que se encontra o valor mínimo.

Observando o corpo humano constata-se que este apresenta um comprimento do braço igual ao antebraço e a distância entre os ombros é de 1 a 1,5 vezes maior que o comprimento total de um braço.

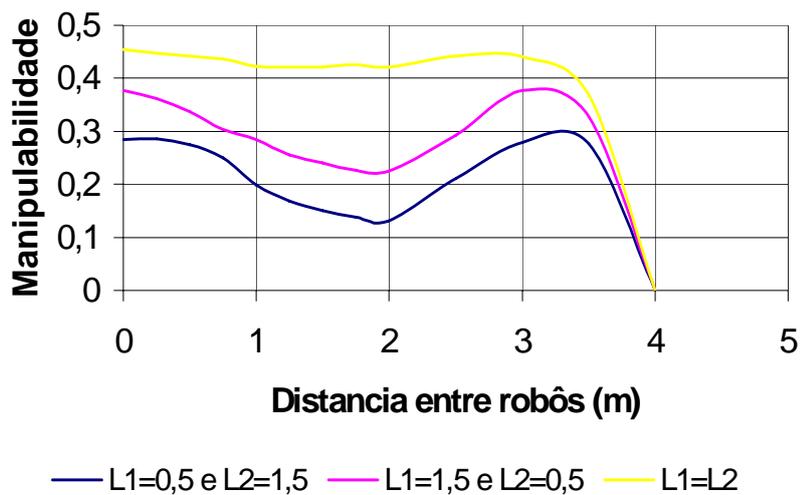


Figura 5.35 – Manipulabilidade máxima.

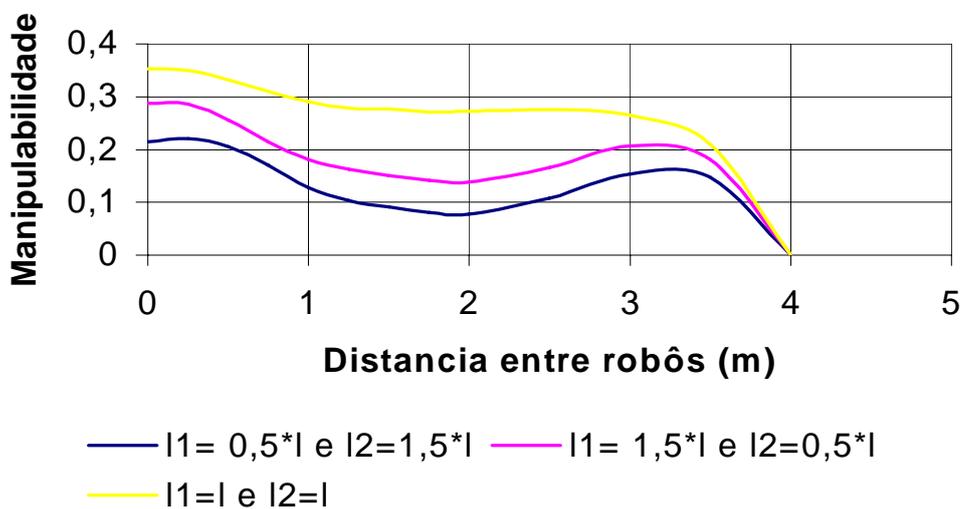


Figura 5.36 – Volume médio de manipulabilidade.

## 5.7 Conclusões

Conclui-se assim, que os robôs apresentam uma manipulabilidade máxima para elos iguais, tal como se apresenta no corpo humano. Todavia, considerou-se apenas a manipulação de pequenos objectos e portanto, não se pode comparar com o corpo humano, visto que este pode manipular objectos de dimensões variadas. Desta forma, deve-se continuar o estudo dos dois braços a trabalhar em cooperação, e efectuar simulações considerando objectos com várias dimensões.



# CAPÍTULO 6

## Conclusão

### 1.1 Introdução

Este último capítulo é constituído por duas secções. Na primeira secção são apresentadas as conclusões finais relativas ao trabalho implementado. Na segunda secção apontam-se possíveis melhoramentos a efectuar no programa e as perspectivas de desenvolvimento futuro

### 1.2 Conclusões do trabalho

Neste trabalho fez-se um estudo de manipuladores robóticos, com estruturas RR e RP, com vários fenómenos dinâmicos nas juntas bem como de vários algoritmos de controlo. Nesta perspectiva, desenvolveu-se o programa *RobLib* no ambiente Windows, como uma ferramenta útil e de fácil utilização que permite simular a cinemática e dinâmica dos robôs a simular, visualizar a sua região de trabalho e um planeamento de trajectórias rectilíneo, quer no espaço cartesiano quer no espaço das juntas. O *RobLib* permite também visualizar inúmeras variáveis tais como a posição, a velocidade, a aceleração, os erros de seguimento da trajectória em ambos os espaços, as forças e os binários a aplicar nos actuadores.

Além disso, o simulador tem uma base de dados integrada que permite armazenar as simulações, para efectuar uma análise dinâmica dos robôs e estabelecer uma comparação de vários algoritmos de controlo de manipuladores robóticos.

O *RobLib* constitui uma plataforma a partir da qual se pode efectuar numerosas experiências, para várias estratégias de controlo, tipos de planeamento de trajectórias, diferentes fenómenos dinâmicos do sistema e da superfície de restrição no caso dos algoritmos de controlo de posição/força.

Nesta perspectiva, os objectivos deste trabalho consistiram em :

- Fazer uma análise dinâmica de robô industriais, de modo a efectuar um estudo dos fenómenos dinâmicos que aparecem neste tipo de sistemas;
- Desenvolver um programa intitulado *RobLib* (*Robot Library*), que permite a simulação de robôs e de algoritmos de controlo. O programa disponibiliza robôs de duas juntas (com estruturas RR e RP) ideais ou com fenómenos dinâmicos nas juntas, assim como um conjunto de algoritmos de controlo de posição e de força;
- Construir uma base de dados integrada no programa *RobLib* de modo a facilitar o estudo dos robôs e dos diversos algoritmos de controlo;
- Criar uma ferramenta numérica que permita o cálculo e a visualização da manipulabilidade de sistemas robóticos com um ou com dois braços no seu espaço de trabalho.

### **1.3 Perspectivas de Desenvolvimento Futuro**

A partir do trabalho desenvolvido, os seguintes aspectos correspondem a hipótese de melhoria ou de aumento das capacidades do *RobLib*:

- alargar o número de modelos de robôs disponíveis, de modo a ampliar a biblioteca;

- implementar as cinemáticas directa e inversa por outros métodos, nomeadamente, através dos ângulos de Euler, de Bryant, náuticos, dos parâmetros de Euler ou das transformações homogéneas;
- implementar outras funções de interpolação no planeamento de trajectórias;
- implementar estratégias automáticas de desvios de obstáculos;
- acrescentar mais fenómenos dinâmicos nas juntas;
- implementar novos algoritmos de controlo de manipuladores robóticos, tais como, algoritmos adaptativos, preditivos, com aprendizagem ou com lógica fuzzy entre outros, e, para o controlo de posição/força, o algoritmo de impedância;
- implementar o controlo dinâmico de dois robôs a trabalhar em cooperação, com a possibilidade de estes serem de estruturas diferentes, e com diferentes fenómenos dinâmicos nas juntas;
- utilizar uma linguagem e uma plataforma de trabalho eventualmente mais eficiente, tal como a linguagem C++ e o sistema operativo Unix;
- implementar uma linguagem de programação, para os diversos robôs existentes, de modo a possibilitar uma programação de tarefas.
- Desenvolvimento de um manual do utilizador do *RobLib*.

A introdução de novas funcionalidades no *RobLib* é facilitada pela forma estruturada como o programa foi desenvolvido. Assim, a implementação numa nova linguagem de programação só acarretará algum trabalho adicional, no tocante à nova plataforma de trabalho, se se pretender criar um ambiente de trabalho distribuído, como por exemplo para uma sala de aula, onde o servidor terá a base de dados, que poderá ser consultada pelos diversos clientes. Da mesma forma, poderá também ser criada uma página WEB para a consulta de simulações.



## Referências Bibliográficas

- [1] Megahed, Said M.;" Principles of Robot Modelling and Simulation", John Wiley & Sons,1993.
- [2] John j. Craig, "Introduction to Robotics Mechanics and Control", Addison-Wesley Publishing Company, 1989.
- [3] Fu, K. S.;Gonzalez, R.C.; Lee, "Robotics: Control, Sensing, Vision and Intelligence", McGraw Hill International Editions, 1987.
- [4] Sciavicco, Lorenzo, Siciliano, Bruno, "Modeling and Control of Robot Manipulators", McGraw-Hill International Editions,1996.
- [5] Machado, J.A T., Tese de Doutoramento:"Gestão dos Recursos Estruturais no Controlo de Robots Manipuladores" Faculdade de Engenharia da Universidade do Porto, 1988.
- [6] Machado, J. A T.; Gadanho, A M. S. F., "Evaluation of Manipulator Direct Dynamics using Customized Runge –Kutta Methods", SAMS,Vol17, pp.229-239,1995.
- [7] Readman, MarK C; "Flexible Joints Robots", CRC press,1994.
- [8] Wit, Carlos Canudas; Siciliano, Bruno; Bastin, Georges, "Theory of Robot Control", Springer,1996.
- [9] Dawson, Darren M.; Bridges Michael M.; Qu, Zhihua; "NonLinear Control of Robotic systems for Envirmental Waste and Restoration", Prentice Hall PTR,1995.
- [10] Carvalho, J.L. Martins;" Dynamical Systems and Automatic Control", Prentice Hall International Series in Systems and Control Engineering,1993.

- [11] Chae H, An; Atkerson, Christopher D.; Hollerbach, John M.; “Model Based Control of Robot Manipulator”, MIT press series in artificial Intelligence,1988.
- [12] Machado, J. A T.; Carvalho, J. L. M., Gadanho, A M. S. F., “Analysis of Robot Dynamics and Compensation Using Classical and Computed Torque Techniques”, ”, IEEE transactions on education,36,No. 4, pp372-379, 1993.
- [13] Machado, J.A T; Carvalho, J. L. M., “Smooth Variable Structure Control Algorithm for Robot Manipulators”, IEE International Conference on Control Oxford, UK, 1988.
- [14] Machado, J. A. T., “Controlo de Estrutura Variável de Manipuladores com Folgas e Flexibilidade nos Eixos”, 4<sup>as</sup> Jornadas Luso-Espanholas de Engenharia Electrotécnica, Porto, Portugal, pp 1531-1538, 1995.
- [15] Machado, J. A T.; “Variable Structure Control of Manipulators with Joints Having Flexibility and Backlash” Proceedings of 7th International Conference on Advanced Robotics, Spain, pp 953-959, 1995.
- [16] Azenha, Abílio Joaquim G. Oliveira, Tese de Doutoramento :”Análise Dinâmica e Controlo de Força em Manipuladores Robóticos”, Faculdade de Engenharia da Universidade do Porto, 1998.
- [17] Azenha, A; Machado, J. A. T.; “Controlo Híbrido de Manipuladores Robóticos”, 4<sup>as</sup> Jornadas Luso- Espanholas de Engenharia Electrotécnica, Porto, Portugal , pp. 1523-1530,1995.
- [18] Machado, J. A T.; Gadanho, A M. S. F., “WinRob: Na Educational Program for Robotics” Journal of Robotics Systems,Vol 34,pp.37-47,1997.
- [19] Cordeiro, Carla Sofia, Tese de Mestrado: “RobLib uma Biblioteca de Robôs Industriais”, Faculdade de Engenharia da Universidade do Porto, 1997.
- [20] Cantù, Marco; “Dominando o Delphi 3 A Biblia”; Makron Books,1998.

[21] <http://www.interbase.com>

[22] Richard M. Murray; Zexiang Li; S. Shankar Sastry, “ A Mathematical Introduction to Robotic Manipulation”, Library of Congress Cataloging-in-Publication Data, 1993.

[23] Asada, Haruhiko; Slotine, Jean- Jacques E., “Robot Analysis and Control”, John Wiley & Sons, 1989.

[24] Yoshikawa, Tsuneo; “Manipulability of Robotic Mechanisms”, Automation Research Laboratory Kyoto University, Uji, Kyoto 611, Japan, 1987.

[25] Homsup, W; Anderson J. N. ; “Performance Evaluation of Robotic Mechanisms”, Center for Manufacturing Research Tennessee Technological University Cookeville, Tennessee, 1988.

[26] Galhano Alexandra M.S.F; Machado J.A Tenreiro; Carvalho J.L. Martins; “Modelização de manipuladores robóticos”; Revista de Engenharia, Março 1992.