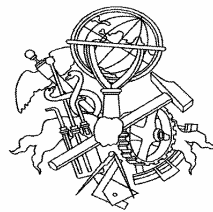


OPTIMIZAÇÃO DOS PARÂMETROS DE UM ROBÔ HEXÁPODE ATRAVÉS DE ALGORITMOS GENÉTICOS

Sérgio Paulo Oliveira Carvalho



Departamento de Engenharia Electrotécnica

Instituto Superior de Engenharia do Porto

2008

Candidato: Sérgio Paulo Oliveira Carvalho, N° 1010903, 1010903@isep.ipp.pt

Orientação científica: Manuel Santos Silva, mss@isep.ipp.pt

Co-Orientação científica: Ramiro Sousa Barbosa, rsb@isep.ipp.pt



Departamento de Engenharia Electrotécnica

Instituto Superior de Engenharia do Porto

13 de Novembro de 2008

Agradecimentos

Desejo expressar a minha sincera e profunda gratidão às seguintes pessoas que contribuíram de forma directa ou indirecta para a realização deste trabalho:

- Professores Manuel Santos Silva e Ramiro Sousa Barbosa pelas suas valiosas orientações, disponibilidade e dedicação.
- Companheiros de turma, Tiago Baptista, José Pinto, Jorge Ribeiro, Décio Esteves, Paulo Melo, Edgar Raimundo e Mateus Semedo, pela sincera e duradoura amizade estabelecida.
- Colegas de trabalho, Sérgio Meira, Sara Santos, Ângela Cunha, Anabela Paiva, Nuno Cunha, Moisés Sampaio e Mário Ramos, por todo o apoio oferecido e pelo convívio gratificante.
- Um agradecimento especial aos meus pais, que sem o seu apoio e esforço, este trabalho nunca teria sido realizado.

Resumo

Os robôs com pernas apresentam vantagens significativas quando comparados com os veículos tradicionais que apresentam rodas e lagartas. A sua maior vantagem é o facto de permitirem a locomoção em terrenos inacessíveis a outro tipo de veículos uma vez que não necessitam de uma superfície de suporte contínua. No entanto, no estado de desenvolvimento em que se encontram, existem vários aspectos que têm que ser necessariamente melhorados e otimizados.

Tendo esta ideia em mente, têm sido propostas e adoptadas diferentes estratégias de optimização a estes sistemas, quer durante a fase de projecto e construção, quer durante a sua operação. Entre os critérios de optimização seguidos por diferentes autores podem-se incluir aspectos relacionados com a eficiência energética, estabilidade, velocidade, conforto, mobilidade e impacto ambiental. As estratégias evolutivas são uma forma de “imitar a natureza” replicando o processo que a natureza concebeu para a geração e evolução das espécies.

O objectivo deste trabalho passa por desenvolver um algoritmo genético, sobre uma aplicação de simulação de robôs com pernas já existente e desenvolvida em linguagem C, que permita optimizar diferentes parâmetros do modelo do robô e do seu padrão de locomoção para diferentes velocidades de locomoção.

Palavras-Chave

Robôs com Pernas, Optimização de Parâmetros, Algoritmos Genéticos, Algoritmos Evolutivos, Padrão de Locomoção.

Abstract

Legged robots have significant advantages when compared with traditional vehicles using wheels and tracks. Their biggest advantage is that they allow the locomotion on terrains inaccessible to other type of vehicles because they don't need a continuous support surface. However, in their actual stage of development, there are several aspects that must necessarily be improved and optimized.

With these ideas in mind, different strategies have been proposed and adopted for the optimization of these systems, either during their design phase and construction, or during their operation. Among the different optimization criteria followed by different authors, it is possible to find issues related to energy efficiency, stability, speed, comfort, mobility and environmental impact. Evolutionary strategies are a way to "imitate nature" replicating the process that nature designed for the generation and evolution of species.

The objective of this project is the development of a genetic algorithm, running over a simulation application of legged robots, already developed in C, which allows the optimization of various parameters of the robot model and of its gaits for different locomotion speeds.

Keywords

Legged Robots, Parameter Optimization, Genetic Algorithms, Evolutionary Algorithms, Gait.

Índice

AGRADECIMENTOS	I
RESUMO	III
ABSTRACT	V
ÍNDICE	VII
ÍNDICE DE FIGURAS	IX
ÍNDICE DE TABELAS	XIII
ACRÓNIMOS	XV
1. INTRODUÇÃO	1
1.1. CONTEXTUALIZAÇÃO	3
1.2. OBJECTIVO DA TESE	4
1.3. PLANO DA TESE.....	4
1.4. ROBÓS MÓVEIS.....	5
1.5. LEIS DA ROBÓTICA	6
1.6. TIPOS, FUNÇÕES E CLASSIFICAÇÃO	7
1.7. CIRCUITOS DE CONTROLO	10
1.8. SENSORES E ACTUADORES.....	11
1.9. PROBLEMAS NA CONCEPÇÃO DE ROBÓS	12
2. PADRÕES DE LOCOMOÇÃO DOS INSECTOS	15
2.1. DEFINIÇÕES GERAIS	15
2.2. OS MECANISMOS DA MARCHA DOS INSECTOS	22
2.3. CONCLUSÃO.....	23
3. ROBÓS HEXÁPODES	25
3.1. DIFERENTES IMPLEMENTAÇÕES DE ROBÓS HEXÁPODES	26
3.2. MODELOS MATEMÁTICOS DO ROBÔ HEXÁPODE ENTRE OS ROBÓS CAMINHANTES	31
3.3. ARQUITECTURA DOS SISTEMAS DE CONTROLO	35
3.4. CONCLUSÃO.....	39
4. INTELIGÊNCIA ARTIFICIAL	41
4.1. PERSPECTIVA HISTÓRICA	43
4.2. APLICAÇÕES DA IA	45

4.3.	MODELOS DE IA.....	46
4.4.	INTELIGÊNCIA E CONHECIMENTO	47
4.5.	TÉCNICAS DE IA.....	48
4.6.	CONCLUSÃO	50
5.	ALGORITMOS GENÉTICOS	51
5.1.	EVOLUÇÃO HISTÓRICA	52
5.2.	CARACTERÍSTICAS GERAIS	53
5.3.	OPERADORES GENÉTICOS	55
5.4.	REPRESENTAÇÃO REAL	59
5.5.	APLICAÇÕES DOS AGs	64
5.6.	PARÂMETROS GENÉTICOS	65
5.7.	ASPECTOS AVANÇADOS.....	66
5.8.	CONCLUSÃO	74
6.	ALGORITMOS GENÉTICOS APLICADOS À ROBÓTICA.....	77
6.1.	CONTROLO DA LOCOMOÇÃO	80
6.2.	PROJECTO DE ROBÔS	82
6.3.	PROJECTO DE CONTROLADORES	84
6.4.	PLANEAMENTO DE TRAJECTÓRIAS.....	87
6.5.	CONCLUSÃO	89
7.	DESENVOLVIMENTO	91
7.1.	MODELO DO SISTEMA DE LOCOMOÇÃO	91
7.2.	ESTRUTURA DO CROMOSSOMA UTILIZADO	99
7.3.	ESTRUTURA BASE DO AG DESENVOLVIDO	102
7.4.	ALGORITMO GENÉTICO DESENVOLVIDO.....	109
7.5.	CONCLUSÃO	116
8.	ENSAIOS E RESULTADOS DAS SIMULAÇÕES	117
8.1.	ENSAIOS DO AG	118
8.2.	OBTENÇÃO DOS PARÂMETROS ÓPTIMOS DO SISTEMA DE LOCOMOÇÃO	123
8.3.	INFLUÊNCIA DA VELOCIDADE NOS PARÂMETROS DO ROBÔ	137
8.4.	OPTIMIZAÇÃO DOS PARÂMETROS E DA VELOCIDADE DE DESLOCAMENTO DO ROBÔ.....	155
9.	CONCLUSÕES E PERSPECTIVAS DE TRABALHO FUTURO	159
	REFERÊNCIAS DOCUMENTAIS.....	163

Índice de Figuras

Figura 1	Robôs Thibodeaux (esquerda), Dogbot (centro), Halluc II (direita)	2
Figura 2	Robô Spirit – missão no planeta Marte (NASA)	2
Figura 3	Robôs BIP (esquerda), AIBO (centro) e Robot III (direita)	3
Figura 4	Isaac Asimov (1920-1992)	6
Figura 5	Braço robótico KR1000 Titan (KUKA)	7
Figura 6	Robô Sojourner – missão Pathfinder (NASA)	7
Figura 7	Robô PackBot EOD, desactiva engenhos explosivos	8
Figura 8	Robô cirurgião Da Vinci	8
Figura 9	Robô Cog (MIT).....	9
Figura 10	Robô ASIMO (Honda)	9
Figura 11	Robôs voador (esquerda), submarino (centro) e micro robô (direita)	10
Figura 12	Robô hexápode Hannibal	13
Figura 13	Descrição do corpo de um insecto	16
Figura 14	Descrição da pata de um insecto	16
Figura 15	Insecto-pau.....	17
Figura 16	Barata (esquerda), caranguejo (centro) e gafanhoto (direita).....	17
Figura 17	Arquitectura do robô Sir Arthur	26
Figura 18	Robô hexápode Robot I.....	27
Figura 19	Robô hexápode Robot II.....	27
Figura 20	Robô hexápode Robot III.....	28
Figura 21	Robô hexápode Robot IV.....	28
Figura 22	Robô hexápode Robot V.....	29
Figura 23	Pata dianteira do robô Robot V.....	29
Figura 24	Robô Rhex	30
Figura 25	Robô desenvolvido no laboratório de automação de <i>Besançon</i>	30
Figura 26	Modelo geométrico da pata de um robô hexápode	32
Figura 27	Postura de referência de um robô hexápode	32
Figura 28	Movimentos do corpo de um robô hexápode	34
Figura 29	Robô hexápode Genghis	36
Figura 30	Robô hexápode Genghis II.....	36

Figura 31	Os cinco tipos de movimentos efectuados por um robô hexápode	37
Figura 32	Estrutura de controlo da arquitectura de subordinação.....	38
Figura 33	Computadores Colossus (esquerda) e ENIAC (direita)	44
Figura 34	Charles Darwin (1809-1882).....	52
Figura 35	Representação de um ciclo de um AG	54
Figura 36	Operador selecção de um AG.....	55
Figura 37	Operador cruzamento de um ponto de um AG	57
Figura 38	Operador cruzamento de dois pontos de um AG	57
Figura 39	Operador cruzamento uniforme de um AG	58
Figura 40	Operador mutação de um AG	59
Figura 41	Cruzamento BLX- α	61
Figura 42	Cruzamento BLX- α com β variável	62
Figura 43	Posição do cromossoma vs. pressão de selecção	70
Figura 44	Gráfico de escalonamento da aptidão	71
Figura 45	Frente Óptima de Pareto.....	72
Figura 46	Conjunto Óptimo de Pareto	73
Figura 47	Ordenamento com base na não-dominação dos indivíduos	73
Figura 48	Método Óptimo de Pareto com recurso a nichos de população	73
Figura 49	Robô Whegs	78
Figura 50	Robô Phony Pony	80
Figura 51	Robô PINO	84
Figura 52	Robô Robug III	85
Figura 53	Robô de resgate criado na Universidade da Flórida do Sul.....	87
Figura 54	Modelo cinemático do robô utilizado [164].....	92
Figura 55	Numeração das pernas do sistema robótico de locomoção [164].....	93
Figura 56	Modelo dinâmico do corpo do robô [164]	95
Figura 57	Arquitectura de controlo do sistema de locomoção [164]	97
Figura 58	Simulação da locomoção do sistema robótico.....	97
Figura 59	Interface gráfica de simulação do sistema robótico	99
Figura 60	Parâmetros do robô utilizados no AG	100
Figura 61	Estrutura do cromossoma utilizado	102
Figura 62	Estrutura base do Algoritmo Genético desenvolvido	103
Figura 63	Função para iniciar a população.....	103
Figura 64	Função para avaliar a população.....	105
Figura 65	Função para analisar a convergência do AG	106

Figura 66	Função de análise do número de gerações criadas pelo AG.....	106
Figura 67	Função de selecção dos cromossomas	106
Figura 68	Função de cruzamento dos cromossomas.....	107
Figura 69	Função de mutação dos cromossomas	108
Figura 70	Seleccção de execução do AG ou verificar valores	109
Figura 71	Introdução dos parâmetros do robô.....	110
Figura 72	Seleccção do parâmetro(s) que se pretende(m) minimizar	110
Figura 73	Seleccção do padrão de locomoção que o robô irá apresentar	111
Figura 74	Seleccção do método de geração da população inicial	112
Figura 75	Introdução do tamanho da população inicial	112
Figura 76	Escolha do método de selecção dos cromossomas	113
Figura 77	Seleccção do método de cruzamento.....	113
Figura 78	Seleccção do método de substituição das gerações.....	114
Figura 79	Seleccção do método de mutação.....	115
Figura 80	Seleccção do número de cromossomas elite.....	115
Figura 81	Comportamento dos diversos métodos do AG.....	119
Figura 82	Influência dos intervalos de pressão no comportamento do AG.....	120
Figura 83	Influência do método de substituição no AG.....	121
Figura 84	Influência do elitismo no AG	122
Figura 85	Soluções obtidas pelo AG – PLO.....	124
Figura 86	Simulação do sistema de locomoção – PLO	124
Figura 87	Soluções obtidas pelo AG – PLFIMC	126
Figura 88	Simulação do sistema de locomoção – PLFIMC	127
Figura 89	Soluções obtidas pelo AG – PLFICC	128
Figura 90	Simulação do sistema de locomoção – PLFICC.....	128
Figura 91	Soluções obtidas pelo AG – PLOI.....	129
Figura 92	Simulação do sistema de locomoção – PLOI.....	130
Figura 93	Soluções obtidas pelo AG – PLFIMCI	132
Figura 94	Simulação do sistema de locomoção - PLFIMCI	133
Figura 95	Soluções obtidas pelo AG – PLFICCI	134
Figura 96	Simulação do sistema de locomoção – PLFICCI.....	134
Figura 97	Comprimento do passo (L_s) em função da velocidade	138
Figura 98	Factor de ocupação (β) em função da velocidade	139
Figura 99	Altura do corpo ao solo (H_b) em função da velocidade.....	140
Figura 100	Altura do pé ao solo (F_c) em função da velocidade	141

Figura 101	Massa do corpo (M_b) em função da velocidade	142
Figura 102	Massa do primeiro elo das pernas dianteiras (M_{11}) em função da velocidade.....	143
Figura 103	Massa do segundo elo das pernas dianteiras (M_{12}) em função da velocidade.....	143
Figura 104	Comprimento do primeiro elo das pernas dianteiras (L_{11}) em função da velocidade	144
Figura 105	Comprimento do segundo elo das pernas dianteiras (L_{12}) em função da velocidade	145
Figura 106	Massa do primeiro elo das pernas do meio (M_{21}) em função da velocidade	146
Figura 107	Massa do segundo elo das pernas do meio (M_{22}) em função da velocidade.....	146
Figura 108	Comprimento do primeiro elo das pernas do meio (L_{21}) em função da velocidade	147
Figura 109	Comprimento do segundo elo das pernas do meio (L_{22}) em função da velocidade	148
Figura 110	Massa do primeiro elo das pernas traseiras (M_{31}) em função da velocidade.....	149
Figura 111	Massa do segundo elo das pernas traseiras (M_{32}) em função da velocidade.....	149
Figura 112	Comprimento do primeiro elo das pernas traseiras (L_{31}) em função da velocidade	150
Figura 113	Comprimento do segundo elo das pernas traseiras (L_{32}) em função da velocidade....	151
Figura 114	<i>Offset</i> dos pés dianteiros em relação às ancas (O_1) em função da velocidade.....	152
Figura 115	<i>Offset</i> dos pés do meio em relação às ancas (O_2) em função da velocidade.....	152
Figura 116	<i>Offset</i> dos pés traseiros em relação às ancas (O_3) em função da velocidade.....	153
Figura 117	Valor do índice E_{av} em função da velocidade.....	154
Figura 118	Valor do índice ε_{xyH} em função da velocidade.....	155

Índice de Tabelas

Tabela 1	Configurações dos diversos ensaios do Algoritmo Genético	118
Tabela 2	Ensaio do AG para diversos intervalos de pressão	120
Tabela 3	Parâmetros otimizados do robô (padrões de locomoção directos).....	125
Tabela 4	Parâmetros otimizados do robô (padrões de locomoção inversos)	131
Tabela 5	Valores mínimos do índice E_{av} , para cada padrão de locomoção, e correspondentes valores de ϵ_{xyH} e da distância percorrida pelo robô	135
Tabela 6	Valores mínimos do índice $\epsilon_{xyH'}$ para cada padrão de locomoção, e correspondentes valores de E_{av} e da distância percorrida pelo robô.....	136
Tabela 7	Resumo dos valores óptimos para cada padrão de locomoção.....	136
Tabela 8	Comparação dos valores óptimos dos índices E_{av} , ϵ_{xyH} e velocidade para cada padrão de locomoção	156
Tabela 9	Parâmetros e velocidade otimizados do robô (padrões de locomoção directos) ...	157
Tabela 10	Parâmetros e velocidade otimizados do robô (padrões de locomoção inversos) ..	158

Acrónimos

- AG – Algoritmo Genético
- gdl – Graus de Liberdade
- IA – Inteligência Artificial
- IGU – Interface Gráfica com o Utilizador
- MIT – *Massachusetts Institute of Technology*
- NASA – *National Aeronautics and Space Administration*
- RN – Redes Neurais
- SWAN – *Simulated Walking ANimat*
- PLO – Padrão de Locomoção Ondulatório
- PLFIMC – Padrão de Locomoção Fase Igual e Meio Ciclo
- PLFICC – Padrão de Locomoção Fase Igual e Ciclo Completo
- PLOI – Padrão de Locomoção Ondulatório Inverso
- PLFIMCI – Padrão de Locomoção Fase Igual e Meio Ciclo Inverso
- PLFICCI – Padrão de Locomoção Fase Igual e Ciclo Completo Inverso

1. INTRODUÇÃO

Os robôs de diferentes tipos (robôs manipuladores, robôs submarinos, robôs com rodas, robôs com pernas) e a robótica em geral desempenham um papel muito importante no sector industrial e estão cada vez mais não só no sector dos serviços, mas também em sectores como o entretenimento, a exploração de locais remotos, a medicina, etc. As estruturas, formas e funcionalidades destes robôs devem ser adaptadas ao ambiente com o qual interagem.

Os robôs manipuladores podem ser utilizados para tarefas de triagem, de montagem (nomeadamente no sector automóvel) ou mais recentemente noutras operações que exigem precisão como, por exemplo, em cirurgias.

Os robôs submarinos são utilizados para explorar e filmar a vida submarina, trabalhar nos fundos marinhos, procurar ou recuperar objectos muito importantes (como a caixa preta de um avião que caiu no mar), ou vestígios históricos como por exemplo os destroços de navios naufragados.

Os robôs com rodas, podem ter um número variável de rodas. Na Figura 1, por exemplo, são apresentados o robô Thibodeaux (4 rodas) [1], o robô Dogbot (6 rodas) [2] e o robô

Halluc II (8 rodas) [3]; o número de rodas varia de acordo com a carga a transportar e também de acordo com a sua aplicação. Estes robôs são utilizados nas mais diversas actividades, essencialmente pela sua capacidade de se deslocarem com uma velocidade elevada e pelo fácil controlo. Contudo, sentem maior dificuldade na deslocação sobre terrenos acidentados.



Figura 1 Robôs Thibodeaux (esquerda), Dogbot (centro), Halluc II (direita)

Em certos casos, os modelos de concepção geométrica, mecânica ou electrónica devem ser estudados especificamente, como é o caso dos robôs enviados pela *National Aeronautics and Space Administration* (NASA) para a exploração do planeta Marte (Figura 2).



Figura 2 Robô Spirit – missão no planeta Marte (NASA)

Os robôs com pernas, podem ser muito diferentes, quer pela sua forma, assim como pelo seu número de patas (Figura 3). Existem robôs com duas patas como o robô bípede BIP [4], quatro patas como o robô AIBO da Sony [5], que imita um cão, robôs com seis patas como o robô Robot III [6] e robôs com oito patas, como o robô escorpião desenvolvido por Kirchner [7], que tem como objectivo atravessar 40 quilómetros do deserto do

Mojave, na Califórnia, até um determinado ponto e, seguidamente, voltar ao seu ponto de partida.



Figura 3 Robôs BIP (esquerda), AIBO (centro) e Robot III (direita)

1.1. CONTEXTUALIZAÇÃO

A robótica é uma área que une diversos conhecimentos, como por exemplo: engenharia mecânica (como a estática e a dinâmica), matemática (com a descrição do movimento no espaço), engenharia elétrica (com o projecto de sensores e actuadores para os robôs), teoria do controlo (com o projecto de algoritmos para que o robô realize os movimentos desejados) e ciência da computação (com a programação de todos os algoritmos desenvolvidos por forma a realizar a tarefa desejada).

Os robôs com pernas apresentam vantagens significativas quando comparados com os veículos tradicionais que apresentam rodas e lagartas. A sua maior vantagem é o facto de permitirem a locomoção em terrenos inacessíveis a outro tipo de veículos uma vez que não necessitam de uma superfície de suporte contínua. No entanto, no estado de desenvolvimento em que se encontram, existem vários aspectos que têm que ser necessariamente melhorados e otimizados.

Este trabalho surgiu da necessidade de otimizar as características mecânicas dos robôs com pernas de forma a obter-se um consumo energético menor e também a diminuir o erro nas trajectórias das ancas. O objectivo é diminuir algumas das desvantagens que este tipo de robôs apresenta quando comparado com os robôs com rodas e lagartas.

1.2. OBJECTIVO DA TESE

O objectivo deste trabalho passa por desenvolver um AG, sobre uma aplicação de simulação de robôs com pernas já existente e desenvolvida em linguagem C, que permita otimizar diferentes parâmetros do modelo do robô e do seu padrão de locomoção para diferentes velocidades de locomoção.

O robô que irá servir de base a este trabalho tem seis pernas (hexápode), pesa 100 kg, o comprimento de cada uma das pernas é de 1 m, cada uma é composta por dois segmentos, o fémur (ligação entre anca e joelho) e o tibiotarso (ligação entre o joelho e o pé), e o espaçamento entre elas é igualmente de 1 m. Estes parâmetros não podem sofrer alterações, mas outros parâmetros como massa do corpo e pernas, comprimento do passo, altura do corpo ao solo, etc., irão ser alterados de forma a serem obtidos os valores óptimos.

Entre os critérios de optimização seguidos por diferentes autores podem-se incluir aspectos relacionados com a eficiência energética, estabilidade, velocidade, conforto, mobilidade e impacto ambiental. As estratégias evolutivas são uma forma de “imitar a natureza” replicando o processo que a natureza concebeu para a geração e evolução das espécies.

1.3. PLANO DA TESE

Este manuscrito encontra-se estruturado de acordo com os capítulos seguintes:

- Capítulo 1: neste primeiro capítulo, será efectuada uma introdução ao trabalho a ser desenvolvido, e serão apresentadas algumas definições e alguns tipos diferentes de robôs.
- Capítulo 2: no capítulo dois, são apresentadas as noções básicas sobre os insectos caminantes, as suas estruturas biológicas e os seus padrões de locomoção na natureza.
- Capítulo 3: o terceiro capítulo, permite uma transição da biologia à robótica. São apresentadas algumas realizações conhecidas de robôs hexápodes e são descritos

diferentes modelos deste tipo de robôs caminhantes.

- Capítulo 4: no quarto capítulo, é apresentado um estado da arte sobre a aprendizagem com os diferentes métodos e algoritmos de aprendizagem.
- Capítulo 5: no quinto capítulo, são apresentadas as abordagens teóricas sobre o AG proposto para a resolução deste trabalho.
- Capítulo 6: no sexto capítulo, são apresentados alguns exemplos de algoritmos genéticos aplicados à robótica.
- Capítulo 7: o sétimo capítulo é dedicado aos procedimentos efectuados.
- Capítulo 8: no oitavo capítulo são apresentados os resultados obtidos.
- Conclusão Geral e Perspectivas: por último são recapitulados os resultados obtidos e indicadas perspectivas para desenvolvimentos futuros.

1.4. ROBÔS MÓVEIS

Existem duas alternativas na definição de robôs móveis [8]. A primeira diz que eles devem ser completos, auto-contidos, autónomos e necessitam de instruções do seu criador apenas ocasionalmente. A segunda é aquela que os define como sendo qualquer equipamento que se mova pelos seus próprios meios, com o objectivo de executar tarefas semelhantes aos humanos.

Um robô móvel é capaz de se mover livremente no ambiente em que se encontra, realizando a função predestinada, desviando-se de possíveis obstáculos. A investigação dos movimentos e controlo têm recebido muita atenção nas duas últimas décadas. Duas abordagens surgiram destas pesquisas: a baseada em planeamento e a baseada em reacção.

A primeira utiliza um modelo global para gerar a sequência de acções mais apropriada para o agente encontrar o seu objectivo. Executando o planeamento livre podem-se gerar sequências óptimas num ambiente estático, porém complexo. A segunda incorpora

reações instantâneas, nos motores do robô, a qualquer informação recebida. Ele pode responder imediatamente e de forma fiável a qualquer alteração no ambiente. No entanto, perde no conhecimento global do ambiente e a sequência de operações pode não ser a melhor numa análise macro.

1.5. LEIS DA ROBÓTICA

O termo *robot* deve-se a Karol Capek que o utilizou pela primeira vez em 1917 para denominar uma máquina construída pelo homem e dotada de “inteligência”. É derivado de *robota*, termo que significa trabalho escravo.

Em 1942, Isaac Asimov (Figura 4) começou a utilizar o termo nas suas obras e acabou por popularizá-lo. São do mesmo autor, as leis da robótica [9]:

- Primeira Lei: um robô não pode prejudicar um ser humano, ou através da sua falta de acção, permitir que seja feito um dano ao ser humano.
- Segunda Lei: um robô deve obedecer às ordens dadas pelos seres humanos, excepto quando tais ordens estejam contra a primeira lei.
- Terceira Lei: um robô deve proteger a sua própria existência, sempre que esta protecção não entre em confronto com a primeira e a segunda lei.
- Lei Zero: um robô não pode prejudicar a humanidade ou através da sua falta de acção permitir que algum dano seja feito à humanidade.

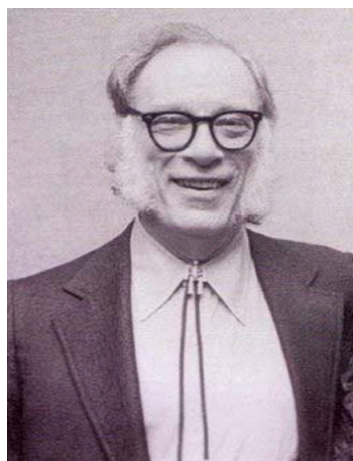


Figura 4 Isaac Asimov (1920-1992)

1.6. TIPOS, FUNÇÕES E CLASSIFICAÇÃO

As funções que os robôs executam, actualmente, são as mais variadas possíveis:

- Os robôs utilizados actualmente na indústria [10], como o da Figura 5, executam tarefas com maior exactidão, velocidade e são incansáveis, realizam trabalhos em ambientes perigosos e até inacessíveis ao homem e são mais económicos que a alternativa humana. Esta questão é um problema social enfrentado no mundo inteiro.



Figura 5 Braço robótico KR1000 Titan (KUKA)

- O Sojourner (Figura 6), foi enviado ao planeta Marte, em 1997, inserido na missão espacial Pathfinder (NASA), e tinha vários objectivos científicos para realizar [11].



Figura 6 Robô Sojourner – missão Pathfinder (NASA)

- O robô PackBot EOD (Figura 7) é utilizado para desactivar bombas e localizar minas evitando sequelas e mortes nos seres humanos [12].



Figura 7 Robô PackBot EOD, desactiva engenhos explosivos

- Os robôs cirurgiões auxiliam os médicos nas operações [13], como é o caso do robô Da Vinci (Figura 8).



Figura 8 Robô cirurgião Da Vinci

- O *Cog* (Figura 9), criado pelo *Massachusetts Institute of Technology* (MIT), é baseado na premissa de que a inteligência nos robôs requer a interacção destes com o mundo. Tem um conjunto de sensores e actuadores que tentam aproximar o seu funcionamento, aos sentidos e graus de liberdade do corpo humano [14].



Figura 9 Robô Cog (MIT)

- A Honda desenvolveu o robô humanóide ASIMO, parecido com um ser humano (Figura 10). Ele é capaz de subir e descer escadas e pode ser utilizado para fazer companhia aos seres humanos [15].



Figura 10 Robô ASIMO (Honda)

- Outros modelos são os robôs voadores, submarinos e micro robôs (Figura 11).



Figura 11 Robôs voador (esquerda), submarino (centro) e micro robô (direita)

Ao observar a forma e o funcionamento dos diferentes tipos de robôs, pode-se deduzir que todos têm algo em comum. A estrutura do chassis que está encarregue de lhe dar forma e sustentar os seus componentes, pode ser constituída por vários materiais, como por exemplo plásticos ou metais e ter várias formas diferentes.

1.7. CIRCUITOS DE CONTROLO

Os circuitos de controlo são o cérebro do robô e, actualmente, são constituídos por componentes electrónicos complexos e directamente proporcionais às funções desempenhadas por este. O controlador é o dispositivo que se encarrega de regular o movimento dos elementos do braço ou das pernas e de todo tipo de cálculos e processamento de informações. Existem várias categorias de controladores:

- Posição, só intervém no controlo de posição do elemento terminal, podendo actuar ponto a ponto ou em modo contínuo.
- Cinemático, além da posição, controla a velocidade do braço.
- Dinâmico, trata das propriedades dinâmicas do manipulador, motores e elementos associados.
- Adaptativo, além das indicações anteriores, também considera a variação das características do manipulador, ao variar a posição.

O controlador pode ser de malha aberta ou de malha fechada. No primeiro caso envia-se um sinal de controlo, mas não se verifica se este é executado com exactidão ou se houve um erro ao executar. No segundo caso existe uma realimentação da saída na entrada, cuja informação é comparada com o sinal enviado. Existindo um erro, procura-se ajustar a trajectória, de maneira adequada, para atingir o ponto pretendido.

1.8. SENSORES E ACTUADORES

Para um robô se relacionar com o ambiente é necessário que ele receba informações sobre as características do mesmo. Isto é feito através de sensores que lhes permitem mover com certa inteligência ao trocar informações com o meio. Os sensores são componentes que detectam e percebem certos fenómenos e situações e pretendem, de certa forma, imitar os sentidos que os seres vivos possuem. Entre os diferentes tipos que se pode encontrar, estão os fotodíodos, os microfones, os sensores de toque, de temperatura, de ultra-sons, câmeras de vídeo, etc.

Podem ainda existir outros tipos de sensores como, por exemplo, os de proximidade, que podem ser ópticos ou através de ultra-sons. Os sensores ópticos são simples, baratos e eficientes na detecção de presença. Não são robustos quanto à iluminação ambiente e a calibração depende da textura da superfície. Os sensores baseados em ultra-sons funcionam com a aplicação de impulsos de 40 a 60 kHz com duração de 1 ms e a sua precisão chega a 1% do valor máximo. No entanto, um ângulo de emissão superior a 30° causa reflexões indesejadas.

Os actuadores encarregam-se de mover as articulações através de cabos, polias, correntes, engrenagens e caracterizam-se pelo tipo de energia que utilizam:

- Pneumáticos: utilizam ar comprimido como fonte de energia e são utilizados para controlar movimentos rápidos, porém não de muita precisão.
- Hidráulicos: são utilizados quando se necessita de uma grande capacidade de carga, juntamente com um controlo de velocidade preciso.
- Eléctricos: são os mais utilizados, pelo seu controlo fácil e preciso, e pelas vantagens do funcionamento usando energia eléctrica.

1.9. PROBLEMAS NA CONCEPÇÃO DE ROBÔS

As perguntas seguintes colocam-se ao conceber um robô hexápode que pode evoluir num ambiente desconhecido:

- Qual o modelo mecânico de robô hexápode que se deve adoptar, qual o número de graus de liberdade e qual a forma do corpo e das patas para evoluir em terrenos variados?
- Entre os padrões de locomoção observados nos insectos, quais são os que podem ser aplicados aos robôs hexápodes?
- Como tratar simultaneamente os problemas de deslocação, manutenção do equilíbrio e postura, detecção de obstáculos e navegação. Qual a estratégia de controlo mais adaptada para satisfazer estes objectivos?
- Como decompor um sistema com grande número de graus de liberdade, como um robô hexápode, em vários subsistemas mais simples de controlar?
- Como pode um robô hexápode, que evolui num ambiente desconhecido, adaptar-se e aprender interagindo com o ambiente?

Pensando nestas questões, a equipa de Rodney Brooks (MIT) efectuou vários projectos de construção de robôs hexápodes como o robô Hannibal (Figura 12). Estes trabalhos comportam estudos sobre padrões de locomoção (inspirados na locomoção dos insectos), modelos dinâmicos, arquitecturas de controlo e a realização de robôs reais, capazes de evoluir em terrenos próximos dos terrenos naturais [16].



Figura 12 Robô hexápode Hannibal

2. PADRÕES DE LOCOMOÇÃO DOS INSECTOS

Neste capítulo são apresentadas definições e noções relativas aos padrões de locomoção dos insectos e também alguns resultados de estudos biológicos que têm como objectivo compreender os princípios utilizados pelos insectos para andar em diferentes ambientes. Assim são apresentadas as regras que os insectos parecem seguir para coordenar as patas, manter o equilíbrio e adaptar-se ao terreno. Diferentes termos como o movimento, a manutenção, a estabilidade e a postura, são utilizados para definir e caracterizar o andar de um insecto ou de um robô hexápode.

2.1. DEFINIÇÕES GERAIS

2.1.1. OS INSECTOS

Os insectos pertencem ao grupo dos artrópodes e são dotados de três ou quatro pares de patas. Os insectos apresentam uma dimensão variável, sendo que os mais pequenos medem menos de 0,5 cm e os maiores atingem a dimensão de pequenos mamíferos.

Uma espécie de escaravelho (*Dynastes*) excede os 16 cm de comprimento e certas borboletas podem atingir os 30 cm de envergadura.

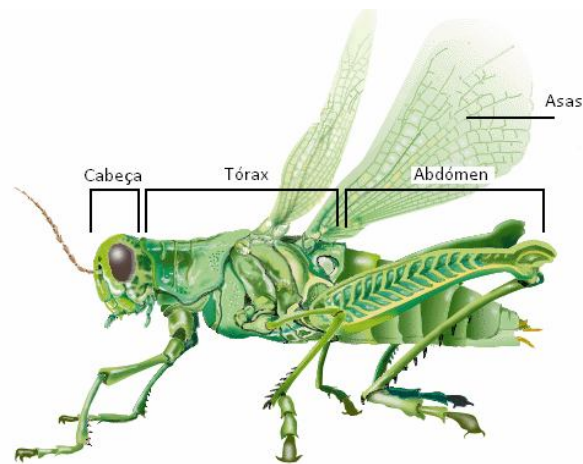


Figura 13 Descrição do corpo de um insecto

O corpo dos insectos, assim como o dos outros artrópodes, é formado por diversos elementos chamados segmentos (Figura 13). Em todos os insectos, o tórax é formado de três segmentos distintos. Cada segmento comporta um par de patas. A forma das patas varia em função da utilização que é feita. Podem servir para andar, para saltar, para nadar ou, ainda, para escavar no solo. Podem igualmente servir para actividades muito mais especializadas, como na abelha, para a colheita e o transporte do pólen. Em alguns casos, a pata é constituída por cinco peças chamadas segmentos. Estes segmentos, articulados entre si, receberam os nomes de coxa, "*trochanter*", fémur, tibia e de tarso (Figura 14). A coxa assegura a articulação da pata com o tórax.

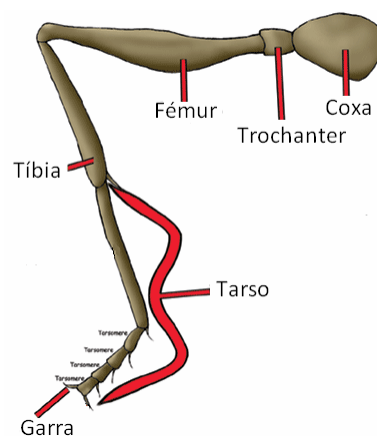


Figura 14 Descrição da pata de um insecto

Nos insectos do tipo caminhante as coxas são de dimensão importante, aumentando assim a força das patas. De notar ainda que em certas espécies, a tibia e o tarso podem ser unidos num só, o que dá origem ao tibiotarso, um exemplo é o insecto-pau (*stick insect*) (Figura 15), que é caracterizado pela sua capacidade de se confundir com o meio que o rodeia. Geralmente chamado "vara do diabo" este insecto vive principalmente nas regiões quentes e húmidas, equatoriais e tropicais.

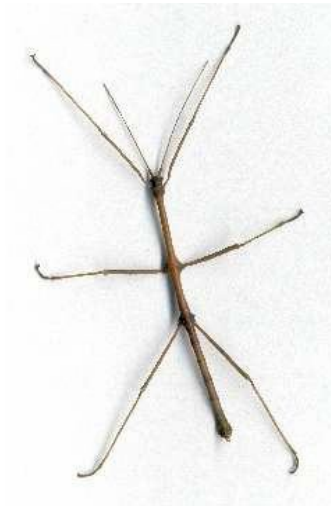


Figura 15 Insecto-pau

O insecto-pau serve de inspiração para o desenvolvimento de robôs com patas, mas não é caso único, outros animais (Figura 16) também são fonte de inspiração para o desenvolvimento de robôs com patas como a barata, o caranguejo, o gafanhoto, etc.



Figura 16 Barata (esquerda), caranguejo (centro) e gafanhoto (direita)

2.1.2. MOVIMENTO

De acordo com Massion [17], o movimento é uma fonte de desequilíbrio que altera a geometria do corpo e altera a posição do seu centro de gravidade. O corpo é composto por segmentos como a cabeça, o tronco e os membros, que são igualmente compostos por vários segmentos ligados pelas articulações. Para estabilizar o centro de gravidade, aquando da deslocação de um segmento, é necessário que outros segmentos participem no movimento. O movimento de um segmento perturba assim a posição dos outros segmentos.

Distinguem-se duas origens do movimento, o movimento passivo ou imposto que é produzido por uma força externa (empurrão, carga adicional) e um movimento activo que é associado a um controlo exercido pelo sistema nervoso.

Entre os movimentos activos distinguem-se os movimentos voluntários, os movimentos reflexos e os movimentos automáticos. Os movimentos voluntários são movimentos que aparecem sem estímulo externo aparente, com base numa instrução interna. Os movimentos reflexos caracterizam-se numa reacção desencadeada por um ou vários estímulos sensoriais bem identificados. O movimento automático é um movimento cujo desenrolar é estereotipado e reprodutível, e que é gerado por uma rede nervosa de maneira inata ou após uma aprendizagem.

2.1.3. ESTABILIDADE

Negligenciando qualquer fenómeno de adesão ou atrito entre os pés e o solo, a estabilidade depende da posição relativa do centro de gravidade em relação aos pontos de contacto com o solo, que definem o polígono de sustentação. A estabilidade é obtida se a vertical que passa pelo centro de gravidade do corpo atravessa o polígono de sustentação.

Podemos distinguir três tipos de estabilidade [18]:

- A estabilidade quasi-estática, neste caso a vertical que passa pelo centro de gravidade passa dentro do polígono de sustentação;

- A estabilidade quasi-dinâmica, é caracterizada por fases em que a vertical que passa pelo centro de gravidade se encontra no limite do polígono de sustentação;
- A estabilidade dinâmica, é caracterizada por fases em que não existem mais do que dois contactos com o solo, não se podendo, assim, formar o polígono de sustentação.

2.1.4. POSTURA

A postura é definida pela posição dos diferentes segmentos num dado momento. A postura assegura essencialmente duas funções [17]:

- Uma função anti-gravidade: consiste em dispor os segmentos de forma a oporem-se à força da gravidade;
- Uma função de interface com o mundo exterior para percepção e acção.

Celaya e Porta propõem um modelo da postura dos robôs caminantes hexápodes [19]. Este modelo apoia-se sobre a noção de polígono de configuração. O polígono de configuração é o polígono cujos limites correspondem às extremidades que têm por papel suportar o corpo (as patas). Não é necessariamente um polígono plano porque o solo pode ser irregular.

A postura é o conjunto das posições dos pés em relação ao corpo. Uma postura é definida pela forma do polígono de configuração e pela posição deste em relação ao corpo. Duas posturas são compatíveis se admitirem o mesmo polígono de configuração. O controlo da postura consiste então a deslocar o corpo em relação aos pontos de contacto ao solo de forma a melhorar a estabilidade e a mobilidade do robô caminhante [19]. A postura desempenha um papel primordial na organização dos movimentos.

2.1.5. LOCOMOÇÃO

A locomoção designa a actividade de deslocação do conjunto do corpo para um ponto preciso do espaço. Nos insectos distinguem-se vários tipos de locomoção: a marcha, o andar, o correr, o salto, o voo e, eventualmente, a natação. Confunde-se geralmente a

actividade locomotora com a do andar. O insecto deve apoiar o seu corpo num número variável de pontos de apoio e utilizar estes apoios para exercer uma força sobre o solo.

2.1.6. PADRÃO DE LOCOMOÇÃO

O andar é um dos principais meios utilizados para assegurar a deslocação do corpo para um ponto preciso do espaço e deve responder simultaneamente a várias exigências. Deve, em primeiro lugar, impulsionar o corpo de acordo com um eixo principal, ou seja, para a frente ou para trás. Esta propulsão deve poder ser modulada em velocidade, de maneira a responder às exigências ligadas à motivação e ao ambiente.

A deslocação do corpo durante a locomoção deve poder ser orientada para um objectivo (é necessário poder curvar a trajectória à direita ou à esquerda para se dirigir para esse objectivo).

O ciclo de marcha (de uma perna ou pata) define o conjunto dos acontecimentos articulares e musculares que se produzem entre dois apoios sucessivos no solo. Este ciclo compreende duas fases, a fase de transferência e a fase de apoio.

Aquando de uma deslocação, a fase de transferência ou "avanço" é a fase durante a qual o membro se desloca no ar para a frente do corpo e a fase de apoio ou "retracção" corresponde à fase durante a qual o membro é colocado em contacto com o solo. Esta fase decompõe-se em dois tempos: um tempo de contacto com o solo e o tempo de empurrão, ou seja, a propulsão do corpo para a frente.

2.1.7. MARCHA-ATRÁS

Graham [20] estudou a marcha-atrás do insecto-pau, e observou que a marcha-atrás pode ser provocada quando a antena, situada na cabeça do insecto, choca com um obstáculo. Durante a marcha, a sincronização das patas posteriores e anteriores é invertida e a marcha não é coordenada de igual forma.

De acordo com as experiências efectuadas sobre a marcha-atrás dos insectos, é claro que o controlo neural da marcha para a frente ou para atrás do insecto-pau não é inteiramente simétrica.

2.1.8. MUDANÇA DE VELOCIDADE

A velocidade da locomoção está directamente ligada à força produzida pela retracção [20] e a velocidade média da retracção de uma pata é igual à velocidade de locomoção em frente. De acordo com Cruse [18] e Graham [20], a velocidade é praticamente constante durante todo o avanço.

2.1.9. TRANSPOSIÇÃO DOS OBSTÁCULOS E ORIENTAÇÃO

Perante um obstáculo, há três comportamentos possíveis: o primeiro é passar sobre o obstáculo, o segundo é fazer marcha-atrás e o terceiro é contorná-lo.

Para alterar a orientação, os insectos utilizam cinco mecanismos [21]:

- O primeiro mecanismo é fazer variar a frequência dos passos do lado direito e/ou esquerdo, enquanto a manutenção da coordenação é assegurada;
- No segundo mecanismo as patas de um lado do insecto avançam enquanto simultaneamente as patas do outro lado recuam. Este movimento provoca uma rotação sobre o lugar;
- Terceiro mecanismo: certos insectos utilizam uma das patas do meio para empurrar o corpo em direcções diferentes (especialmente as baratas que podem alterar assim rapidamente a orientação durante uma fuga) [22], [23];
- Quarto mecanismo: as patas situadas de cada lado do corpo do insecto são desacopladas, e as patas situadas fora da curva movimentam-se com uma frequência mais elevada;
- Quinto mecanismo: a amplitude dos passos das patas que se encontram fora da curva é aumentada enquanto a amplitude dos passos das patas que se encontram dentro da curva é diminuída, o que provoca uma marcha arqueada [18], [24].

2.1.10. NAVEGAÇÃO

A navegação do insecto é baseada num método de integração de caminho, pelo qual as informações sobre o ambiente e sobre a distância de deslocação são combinadas para calcular a posição. A integração de caminho foi postulada para ser o mecanismo principal que permite aos insectos reencontrar o seu caminho de regresso. Para utilizar este mecanismo de navegação, as duas informações de distância e de direcção devem estar disponíveis [25].

A integração de caminho exige uma memória que permite ao insecto memorizar durante a navegação uma avaliação corrente da direcção e da distância em relação ao ninho. Tal memória, normalmente curta, deve-se ao processo de actualização, mas pode persistir durante várias horas se o insecto for confinado [26].

2.2. OS MECANISMOS DA MARCHA DOS INSECTOS

2.2.1. AS DIFERENTES MARCHAS DOS INSECTOS

Na marcha em tripé (ou trípode) há sempre três patas em contacto com o solo. Esta marcha é utilizada se o insecto quer andar rapidamente e com pouca carga. Na marcha tetrapode há sempre quatro patas em contacto com o solo e esta marcha é utilizada se o insecto quer andar lentamente com uma carga considerável. Randall [21] estabeleceu várias regras que permitem caracterizar as marchas, geralmente, observadas nos insectos sobre uma superfície plana:

- Os avanços propagam-se da parte de trás para a parte da frente. Nenhuma pata começa uma fase de avanço antes que as patas vizinhas estejam na fase de apoio;
- As patas opostas em relação ao eixo longitudinal movem-se em fase alternada (não há avanços simultâneos);
- A duração do avanço pode ser considerada como constante;
- Para aumentar a velocidade de deslocação, a duração da fase de apoio diminui;

- O intervalo de tempo que separa um passo da pata posterior e um passo da pata do meio é igual ao intervalo de tempo que separa um passo da pata do meio e um passo da pata anterior. No entanto, o intervalo entre um passo da pata anterior e um passo da pata posterior varia de forma inversa à velocidade da marcha.

2.2.2. MODELOS DE COORDENAÇÃO

Vários investigadores na área da biologia observaram que o controlo da marcha do insecto-pau não é centralizado, mas decomposto em vários subsistemas independentes e ligados entre si (para cada pata há uma rede de neurónios) [21], [27], [28]. Assim Ferrell [16] estudou os sistemas neuronais que controlam a marcha dos insectos e desenvolveu modelos para explicar o controlo individual de uma pata bem como a coordenação entre as patas.

Cruse [29], [30], [31], estudou a locomoção de vários insectos e desenvolveu dois modelos de locomoção do insecto-pau. O primeiro é um modelo de controlo de uma pata individual e o segundo é um modelo de coordenação entre as patas.

De acordo com Cruse, a coordenação dos movimentos das patas [32] resulta das interacções locais dos seus sistemas de controlo. A interacção implica que as posições extremas da pata dependem das posições do ponto de origem (ou seja, o ponto de contacto com o solo) e do estado das outras patas que são "vizinhas".

Assim, uma pata pode apenas ser influenciada pela pata anterior ou posterior do mesmo lado e pela pata simétrica do lado oposto em relação ao eixo longitudinal. O movimento de cada pata pode assim ser influenciado por duas ou três patas.

2.3. CONCLUSÃO

Foram apresentados, neste capítulo, resultados importantes obtidos por estudos efectuados sobre a marcha dos insectos. Foram apresentadas diferentes definições relativas à locomoção, ao movimento, à postura, à estabilidade, à marcha e à navegação.

Foi observado que a marcha de um insecto é muito complexa, porque põe em jogo funções de locomoção, controlo da postura, manutenção do equilíbrio e depende de factores como a carga transportada ou a natureza do terreno.

Com a finalidade de construir um sistema de controlo da marcha de um robô hexápode, impõem-se as seguintes perguntas:

- Qual deve ser a complexidade do modelo mecânico de cada pata?
- É possível separar as diferentes funções (manutenção do equilíbrio, controlo da postura, controlo do ciclo de locomoção, navegação) e como?
- Para cada função, qual a arquitectura de controlo a utilizar?
- Que tipo de ligações devem ser estabelecidas entre estas funções?

A partir dos resultados conhecidos sobre os insectos, equipas de investigação desenvolvem várias vias para responder às perguntas precedentes. Cada equipa baseou os seus trabalhos sobre um modelo específico de robô hexápode, mas todos os modelos tiram a sua inspiração dos princípios da marcha dos insectos.

Descrevemos no capítulo seguinte diferentes modelos de robôs hexápodes e certas arquitecturas de controlo.

3. ROBÔS HEXÁPODES

Após ter apresentado os princípios da marcha dos insectos, será abordado, neste capítulo, a passagem da biologia (insecto) à robótica, ou seja, a concepção dos robôs caminantes com seis patas. Várias vias de investigação foram exploradas, de acordo com a escolha da estrutura mecânica do robô e a natureza dos modelos utilizados para o controlo.

Os robôs de seis patas (robôs hexápodes) apresentam, em geral, semelhanças com insectos capazes de se deslocarem em terrenos naturais, uma vez que os investigadores procuram inspiração na biologia para desenvolverem os seus robôs. Apresentam um bom compromisso entre complexidade de concepção, estabilidade e mobilidade. Podem ser úteis para aceder a zonas perigosas para o Homem como os locais com resíduos nucleares ou o espaço.

A necessidade de dominar as interacções do robô com o seu ambiente, de coordenar os movimentos durante a marcha, de controlar o equilíbrio e a postura, torna estes sistemas difíceis de controlar, sobretudo quando estes robôs hexápodes possuem grande número de graus de liberdade (por exemplo o robô R-V possui 24 graus de liberdade [33]).

3.1. DIFERENTES IMPLEMENTAÇÕES DE ROBÔS HEXÁPODES

Existem, em laboratório, diversas realizações de robôs hexápodes construídos para aplicações ou objectivos específicos de cada equipa. Neste capítulo irão ser apresentados alguns exemplos conhecidos, para ilustrar a variedade das abordagens.

O robô hexápode Sir Arthur, desenvolvido por Kirchner [34], é composto por três segmentos, com duas patas por segmento. As articulações dos segmentos apresentam, entre si, dois graus de liberdade (gdl). Estes podem ser controlados para levantar ou fazer girar sobre um eixo o segmento anterior e o segmento posterior do robô. As patas são comandadas por dois servomotores, formando os eixos dos motores um ângulo de 90°. Cada pata pode fazer quatro movimentos (para a parte superior, parte inferior, frente e atrás). A combinação dos seus quatro movimentos dá origem aos dois movimentos principais, o avanço e a retracção da pata. A arquitectura do robô é apresentada na Figura 17.

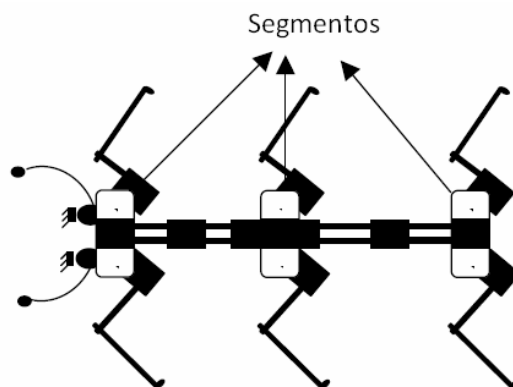


Figura 17 Arquitectura do robô Sir Arthur

A equipa de investigação do laboratório *Biologically Inspired Robotics Laboratory*, da *Case Western Reserve University* nos Estados Unidos [35], cujo director é o professor R.D. Quinn [36] [37], trabalha em colaboração com biólogos como R.D. Beer, H.J. Chiel e R.E. Ritzmann. Esta equipa desenvolveu vários tipos de robôs hexápodes que se diferenciam quer pela forma, quer pelo número de graus de liberdade, quer pela tecnologia mecânica utilizada.

O primeiro robô hexápode, o Robot I, foi criado para estudar o controlo da locomoção recorrendo a redes neuronais [38]. Cada pata deste robô tem dois graus de liberdade. A

sua marcha é inspirada na marcha dos insectos, e são aplicadas as leis estabelecidas por Wilson, em 1966 [39], para evoluir sobre uma superfície plana. Este robô é representado na Figura 18.

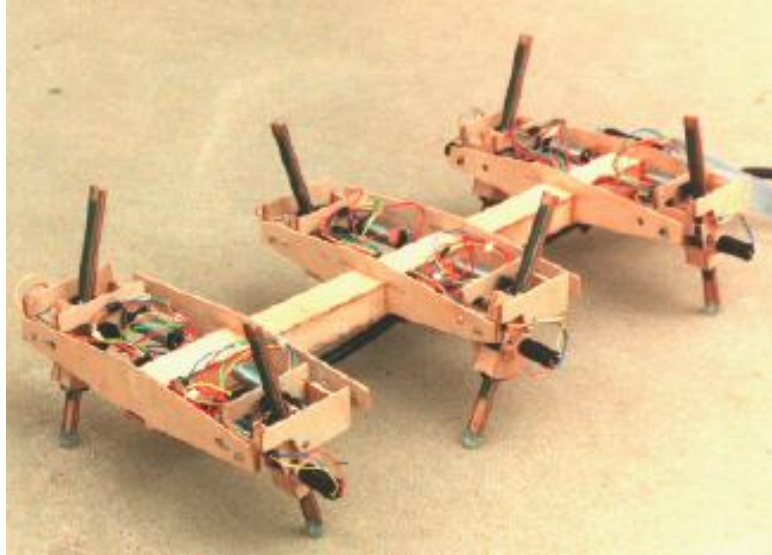


Figura 18 Robô hexápode Robot I

O segundo robô hexápode é o Robot II (Figura 19). Este robô tem uma estrutura similar ao insecto-pau. Cada pata deste robô tem três graus de liberdades. A estratégia e as leis da marcha também são inspiradas na marcha dos insectos, como o robô Robot I. O Robot II também tem a faculdade de andar para atrás, de girar sobre o lugar, de se deslocar lateralmente e de evoluir sobre uma superfície não plana [40], [41].



Figura 19 Robô hexápode Robot II

O terceiro robô, Robot III, é inspirado na barata, apresentando a forma deste insecto. Este robô tem um total de 24 graus de liberdade. Devido a diferenças funcionais, as patas anteriores têm cinco graus de liberdade, as patas do meio quatro e as patas posteriores três. As patas do meio elevam o corpo quando o robô quer trepar um obstáculo. As patas do robô são fabricadas à base de cilindros pneumáticos. O protótipo foi construído para fazer testes de estabilidade, deslocação e adaptação às mudanças do ambiente. Uma versão deste robô é apresentada na Figura 20.

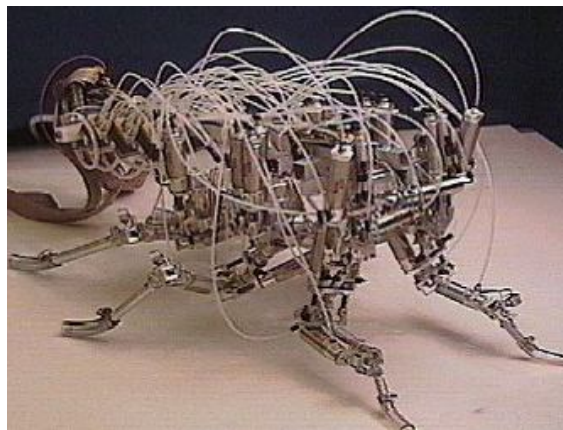


Figura 20 Robô hexápode Robot III

O quarto robô, o Robot IV, foi concebido de modo a que a locomoção do robô se faça economizando a energia. A cinemática das suas patas é idêntica à do robô Robot III, e as patas deste robô têm articulações que apresentam uma rigidez passiva que pode ser regulada para aproveitar o armazenamento de energia durante um ciclo de movimento das patas [37]. O robô Robot IV é representado na Figura 21.



Figura 21 Robô hexápode Robot IV

Kingsley [33] concebeu um robô hexápode inspirado numa barata, como o robô Robot III e Robot IV. O seu nome é Robot V (ou Ajax) (Figura 22). Este robô tem 24 graus de liberdade, cinco graus de liberdade para as duas patas da frente, quatro graus de liberdade para as patas do meio e três graus de liberdade para as patas de trás.

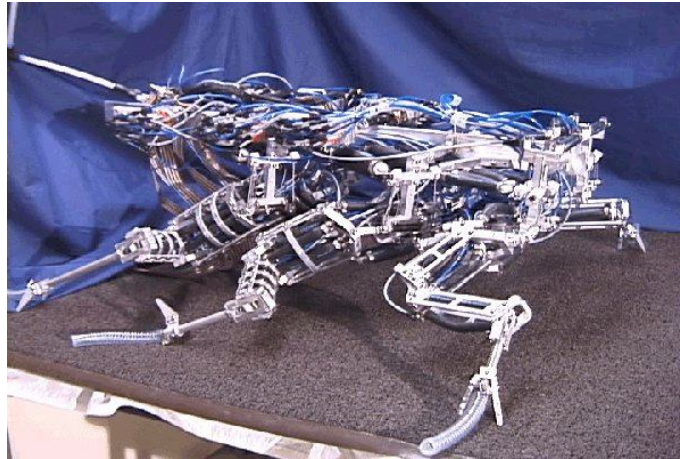


Figura 22 Robô hexápode Robot V

As patas dianteiras são relativamente pequenas e pouco potentes (Figura 23), mas podem manipular eficazmente objectos ou evoluir em terrenos difíceis. As patas do meio são maiores e desempenham um papel muito importante de apoio e levantamento do corpo para as operações de rotação e passagem sobre obstáculos. Como a barata, o robô Robot V utiliza principalmente as suas patas traseiras (mais potentes) para a locomoção. Os actuadores são músculos artificiais.

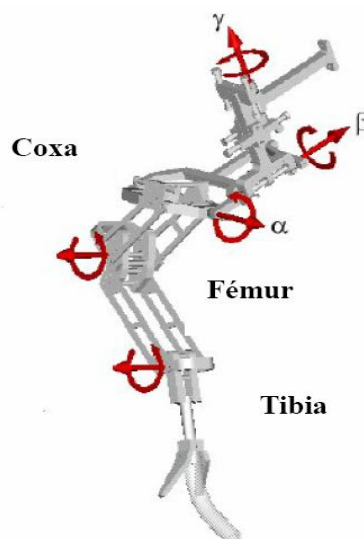


Figura 23 Pata dianteira do robô Robot V

O trabalho de Altendorfer *et al.* [42], [43], é também baseado numa inspiração biológica do modelo do robô hexápode (Figura 24), com um estudo dinâmico do movimento do robô. Usando-o como um modelo físico, os investigadores esperam isolar e entender as complexas redes neuronais e musculares dos insectos. Ao mesmo tempo, eles farão experiências biomecânicas e neurológicas em insectos para desenvolver modelos matemáticos para aperfeiçar o robô.

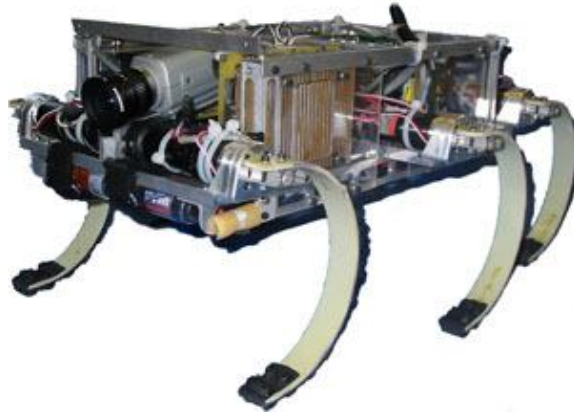


Figura 24 Robô Rhex

Certos trabalhos de investigação recentes têm por objectivo construir robôs hexápodes da dimensão de um insecto. Por exemplo, os trabalhos efectuados no laboratório de automação de *Besançon* [44] têm por objectivo integrar num volume de 2 a 3 cm³, não só as funções locomotoras, mas também os micro-sensores e as conexões necessárias para a transmissão da informação (Figura 25).

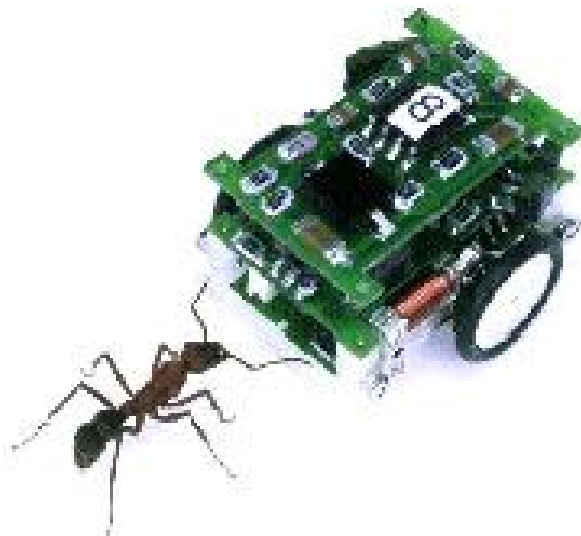


Figura 25 Robô desenvolvido no laboratório de automação de *Besançon*

3.2. MODELOS MATEMÁTICOS DO ROBÔ HEXÁPODE ENTRE OS ROBÔS CAMINHANTES

Os robôs hexápodes são dos mais estudados e mais utilizados pelos investigadores, porque a estrutura do robô hexápode realiza um compromisso interessante entre o custo de realização, complexidade do controlo (em especial do ponto de vista da estabilidade) e adaptação aos terrenos difíceis.

Na natureza, os insectos existem sob várias formas, que deram ideias para construir robôs hexápodes. Estes distinguem-se pela sua concepção mecânica (número de graus de liberdade, formas das patas e do corpo, tipos de actuadores e características dimensionais) e também pelos seus sistemas de percepção do ambiente e controlo.

Em função das aplicações visadas, foram desenvolvidos modelos mais ou menos complicados de robôs hexápodes, entre os quais encontramos modelos lógicos [45], [46], modelos geométricos [21], [47], [48] e modelos dinâmicos [49], [50].

3.2.1. MODELOS LÓGICOS

Neste tipo de modelos, os diferentes estados codificam as posições dos segmentos das patas e as acções que correspondem aos movimentos efectuados para passar de um estado para outro. Estes modelos são utilizados para estudar os mecanismos de coordenação entre as patas para aprender quer o andar [46], [51], [52], quer evitar os obstáculos [45], quer atingir um objectivo fixado no ambiente [53].

Mas a utilização destes modelos lógicos é limitada ao estudo da marcha sobre um solo plano e não tem em conta as irregularidades do terreno. Por outro lado, o número de estados e acções aumenta rapidamente quando, como no modelo de Kirchner [34], a quantificação do espaço se torna mais fina.

3.2.2. MODELOS GEOMÉTRICOS

No robô proposto por Pratt *et al.* [54] e Torres [55] cada pata tem três gdl (o ângulo do joelho e os ângulos de rotação do quadril). Observamos que os três gdl de cada pata abrem possibilidades de movimentos suplementares como, por exemplo, no robô Sir Arthur (apresentado na Figura 17). O modelo deste robô é representado na Figura 26.

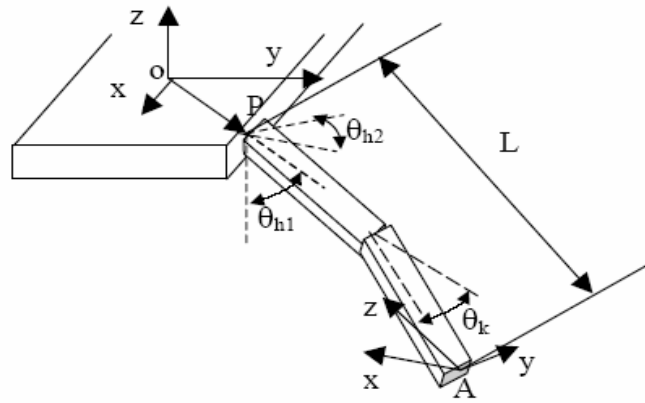


Figura 26 Modelo geométrico da pata de um robô hexápode

Entre os modelos geométricos, existe também o modelo proposto por Porta e Celaya [52]. Este modelo permite descrever a postura e a locomoção do robô hexápode em terrenos difíceis. A mobilidade do robô é melhorada elevando o corpo do robô em relação ao solo e afastando os pés dos limites do seu espaço de trabalho a fim de permitir uma grande variedade de movimentos em todas as direcções.

A postura de um robô é o conjunto das posições dos pés em relação ao corpo. A postura de referência é a que, em condições gerais, oferece melhor compromisso entre estabilidade e mobilidade.

Este modelo permite descrever a progressão do robô sobre qualquer terreno, de acordo com trajectórias rectilíneas ou curvas e também a posição relativa do corpo em relação ao solo. A adaptação ao terreno consiste em posicionar as patas e o corpo tendo em conta a topografia e a reacção a eventuais colisões com obstáculos. Apresenta-se na Figura 27 uma postura de referência.

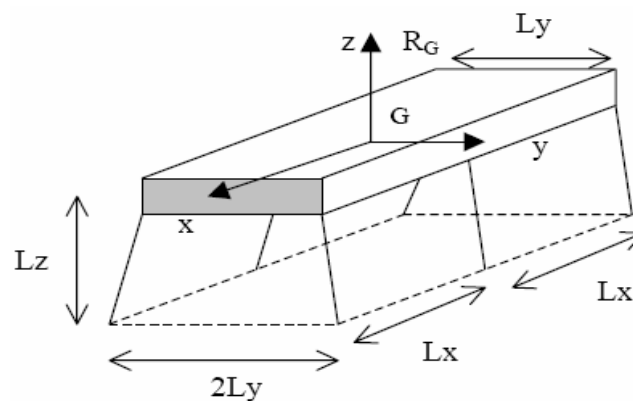


Figura 27 Postura de referência de um robô hexápode

3.2.3. MODELOS DINÂMICOS

Os modelos dinâmicos são baseados em equações matemáticas que descrevem a evolução do robô ao longo do tempo [49], [50], considerando as forças aplicadas e os seus efeitos. São estreitamente ligados ao modelo mecânico do robô (tipo de actuador, forma do corpo e patas).

Estes modelos consideram os efeitos de um empurrão ou resistência das patas ao solo. Uma pata já posta no solo exerce uma força de atrito. Além disso, no momento em que uma pata entra em contacto com o solo exerce uma força dirigida de acordo com o eixo longitudinal no sentido da marcha em frente ou da marcha atrás do robô (de acordo com o movimento da pata).

Entre os modelos dinâmicos apresentamos aqui um modelo de um simulador dinâmico da marcha de um animal (robô hexápode), o SWAN (Simulated Walking ANimat) que foi utilizado por Kodjabachian [56]. Este simulador dinâmico é empregue em simulações de evolução artificial pesadas do ponto de vista computacional. Há duas versões deste modelo de simulador.

Na primeira versão do simulador, chamado SWAN-1D, o robô hexápode desloca-se de acordo com uma só direcção [56], [57], [58] e cada perna tem dois gdl. Para cada perna, há um jogo de controlos que permite alterar a altura do pé e a posição angular da pata. O objectivo deste modelo é testar controladores que produzem diferentes padrões de marcha incluindo o padrão trípole.

A segunda versão do simulador, chamado SWAN-2D, foi explorada em [56]. Neste segundo modelo as patas têm três graus de liberdade. Com estes três graus de liberdade o robô hexápode pode fazer deslocações de acordo com duas dimensões. Estes modelos apresentam uma analogia biológica, ou seja, cada pata é controlada por dois pares de músculos antagónicos.

Nos dois modelos, o estado do robô é definido pela posição do corpo e pela posição das pernas. Uma perna é dita estendida se o seu comprimento excede um limiar dado, se não, diz-se retraída e não suporta pesos. O robô é localizado por três variáveis (x_G , y_G

ψ_G): x_G , y_G representam a posição do centro de gravidade do robô num plano ligado ao ambiente e ψ_G a orientação do robô em redor do eixo vertical concorrido por G , como indicado na Figura 28.

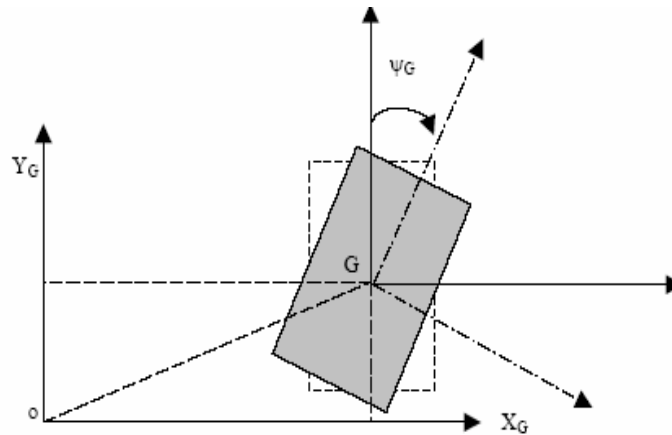


Figura 28 Movimentos do corpo de um robô hexápode

Este modelo dinâmico não permite, contudo, controlar a postura do robô.

Outro modelo de robô hexápode é proposto em trabalhos de Saranli e Buchler [48]. O corpo do robô tem seis graus de liberdade. As pernas são ligadas ao corpo rígido nos pontos de união. Cada perna apresenta um movimento esférico e é considerada de massa negligenciável excepto no avanço, onde a massa do pé (negligenciável em relação à do corpo) é tida em conta.

Este modelo foi utilizado para efectuar simulações da marcha de um robô hexápode sobre um terreno plano ou em declive [59] [60] [61]. Com efeito não há, para os robôs hexápodes, uma estrutura universal e as estruturas estudadas podem afastar-se das de um insecto.

3.2.4. CONCLUSÃO SOBRE OS MODELOS DE ROBÔS HEXÁPODE

De acordo com os modelos apresentados, observamos que não existe um modelo único geométrico, dinâmico ou mecânico para os robôs hexápodes. Cada equipa de investigação usa o seu modelo tendo em vista os seus próprios objectivos e a utilização do robô no seu ambiente. Preocupações comuns são:

- Assegurar o equilíbrio do robô

- Dominar a coordenação entre as patas durante a deslocação
- Adaptar-se às mudanças do ambiente

3.3. ARQUITECTURA DOS SISTEMAS DE CONTROLO

O controlo da locomoção de um robô hexápode é complexo. O sistema de controlo serve para efectuar as tarefas de manutenção do equilíbrio, controlo da postura e geração dos movimentos de locomoção. A solução proposta por Brooks [62] [63] é baseada na arquitectura de subordinação (*subsumption*).

Esta abordagem constrói progressivamente a solução de controlo/comando acrescentando sucessivamente níveis ou camadas funcionais; as camadas inferiores são consagradas às tarefas mais elementares e os níveis superiores realizam as funcionalidades mais complexas. A arquitectura de subordinação é uma abordagem alternativa às abordagens clássicas de controlo baseadas numa decomposição de tarefas complexas em fases sucessivas de planeamento das tarefas, de geração de trajectórias e acompanhamento de trajectórias.

Um aspecto importante da arquitectura de subordinação é que favorece um acoplamento estreito entre os sensores e os actuadores que promovem comportamentos reactivos, em detrimento dos comportamentos baseados em capacidades cognitivas como por exemplo o planeamento.

Os níveis mais baixos podem funcionar independentemente dos níveis mais elevados e os níveis superiores apoiam-se sobre as competências desenvolvidas nos níveis inferiores para acrescentar novas competências. Assim, a arquitectura de controlo do robô Hannibal (Figura 12) é composta de três níveis. O nível mais baixo assegura o acoplamento básico entre os sensores e os actuadores, o nível do meio assegura a locomoção sobre terreno plano e o nível mais elevado assegura a tarefa de locomoção sobre um terreno irregular [16].

Para assegurar o funcionamento correcto do sistema global (robô) é necessário que os subsistemas e os níveis cooperem entre si. Para cooperar, estes subsistemas e níveis

devem comunicar, quer directamente, pela troca de mensagens, quer indirectamente, pela modificação do estado do robô no seu ambiente. Brooks [62] propôs uma rede básica para controlar a locomoção sobre um terreno plano do robô hexápode Genghis (Figura 29), dispondo cada perna do robô de dois gdl.



Figura 29 Robô hexápode Genghis

Dois módulos centrais têm como papel, respectivamente, desencadear os movimentos de elevação e empurrão das patas em função das informações de estado recebidas de cada uma delas. Os outros movimentos são decididos a nível local.

Celaya e Porta [64] [65] [66] propuseram uma estrutura de controlo da marcha do robô hexápode Genghis II (Figura 30) sobre um terreno não regular e com obstáculos. Esta estrutura é baseada na decomposição das tarefas de acordo com a arquitectura de subordinação, na qual os níveis são independentes mas podem trocar dados e informações entre eles.

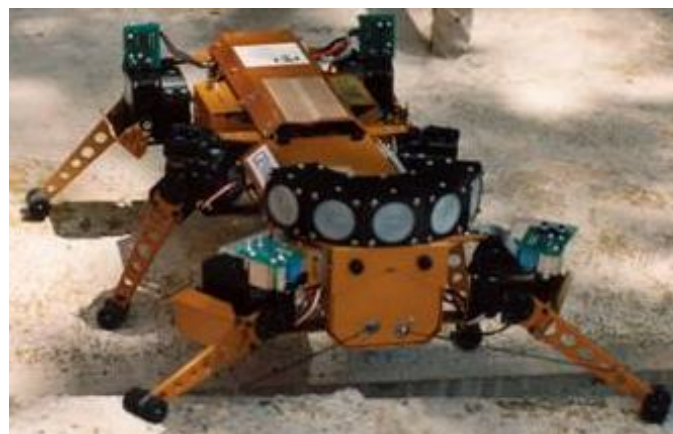


Figura 30 Robô hexápode Genghis II

Cada perna do robô hexápode Genghis II tem dois gdl. Cada perna tem dois movimentos possíveis, um movimento de avanço e um movimento de retracção controlados por dois motores e a estrutura de controlo proposta por Celaya e Porta [64] é composta de seis níveis. Cada nível representa uma competência que resulta de um ou vários comportamentos.

Os diferentes níveis são descritos do seguinte modo:

- Nível 1 (nível da manutenção): O primeiro nível controla directamente os motores para assegurar a manutenção de cada perna do robô na posição pedida. Por defeito esta posição é uma posição de referência que oferece uma boa estabilidade e a sua escolha é importante.
- Nível 2 (nível de controlo da postura): Este nível assegura uma posição correcta do corpo do robô em relação aos pés (controlo da postura). Este nível é composto de cinco comportamentos. Cada comportamento traduz-se quer numa translação, quer numa rotação do corpo. Estes diferentes comportamentos são representados na Figura 31.

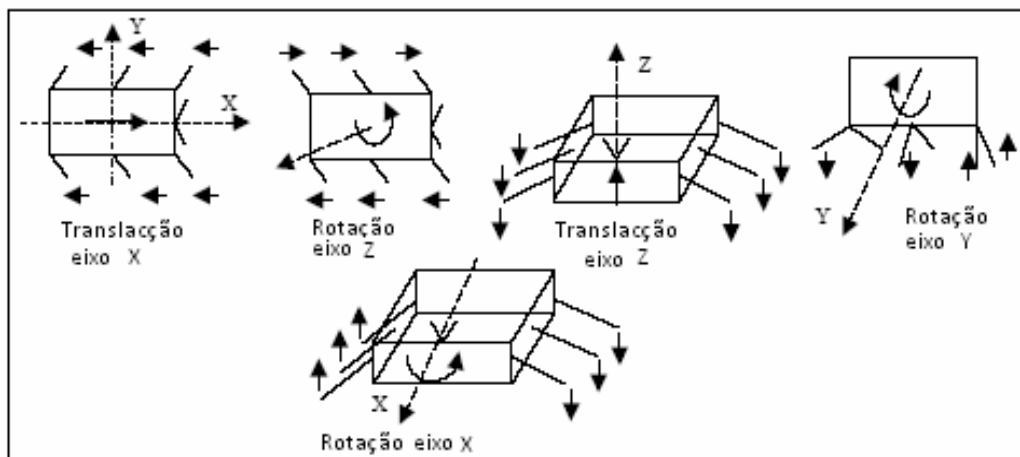


Figura 31 Os cinco tipos de movimentos efectuados por um robô hexápode

- Nível 3 (nível de adaptação ao ambiente): O objectivo deste nível é alterar as instruções de postura geridas pelo nível 2 para adaptar-se às condições do terreno como são percebidas pelos sensores. Assim, de acordo com as necessidades, o corpo pode ser avançado ou recuado relativamente às posições dos pés em contacto com o solo.

- Nível 4 (nível das forças de adaptabilidade): A este nível o robô adapta a posição de cada pé à altitude do solo. Para esse efeito o esforço suportado por cada perna é avaliado e se é inferior a um certo valor, o pé é baixado. Assim, no caso de sobrecarga sobre uma das pernas, as pernas que se encontram sob carga têm tendência a descer para diminuir essa sobrecarga. Esta estratégia, combinada aos efeitos de controlo de postura, constitui uma forma de adaptabilidade activa.
- Nível 5 (nível da marcha): A este nível o robô é capaz de andar sobre um terreno irregular, tentando sistematicamente passar pelos obstáculos. Para cada perna há um comportamento de adiantamento e de retrocesso. O comportamento de adiantamento desencadeia um avanço logo que as duas pernas vizinhas emitam uma mensagem indicando que estão em contacto com o solo. O comportamento de retrocesso produz-se quando a perna encontra um obstáculo durante o movimento de avanço. Então, retorna ligeiramente para atrás e levanta-se antes de retomar o movimento de avanço.
- Nível 6 (nível de navegação): Este nível aplica um comportamento de contorno elementar, afastando o robô dos obstáculos detectados pelos sensores tácteis, de tipo "bigode". A mudança de trajectória é obtida por rotação do corpo, através do movimento das patas de cada lado do corpo em sentido oposto.

Os seis níveis previamente descritos são representados na Figura 32.

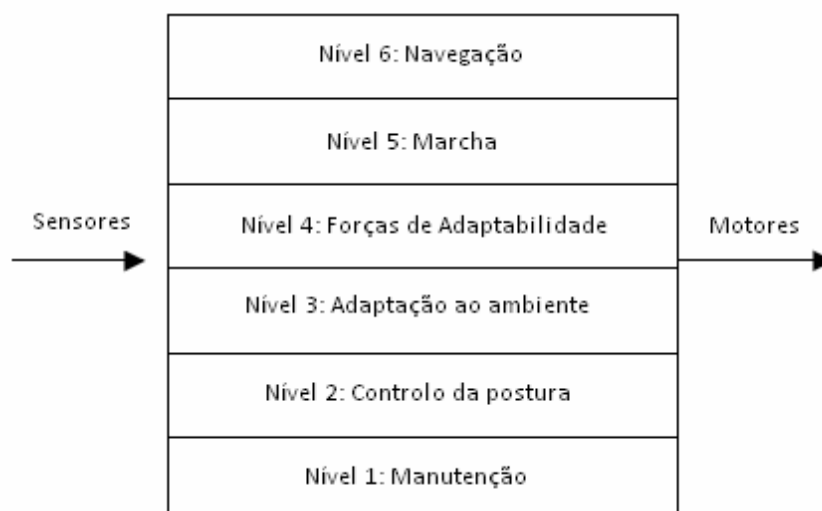


Figura 32 Estrutura de controlo da arquitectura de subordinação

3.4. CONCLUSÃO

Apresentamos, neste capítulo, diferentes exemplos de modelos de robôs hexápodes, em especial os modelos lógicos utilizados para estudar os ritmos de marcha sobre terreno plano, os modelos geométricos (com as suas alternativas de acordo com o número de graus de liberdade e a forma do robô) e os modelos dinâmicos que consideram forças de tracção e forças de torção exercidas sobre as pernas ou sobre o corpo para descrever nomeadamente fenómenos de adaptabilidade necessários para a adaptação a terrenos irregulares.

Estes modelos são utilizados para gerir, de acordo com estratégias variadas, diferentes tipos de marcha, a mudança de trajectória, a transposição de obstáculos e a adaptação a terrenos irregulares.

As estratégias desenvolvidas para o controlo da locomoção de um robô hexápode são frequentemente inspiradas na observação da locomoção dos insectos. Parece então útil decompor o problema do controlo do robô caminhante em funções de complexidade crescente que se apoiam sobre funções mais simples já adquiridas. Assim, as funções elementares de manutenção em posição dos segmentos recebem as suas instruções de um nível superior, responsável pelo controlo da postura, e este obedece às ordens do nível da marcha, que é influenciado pela tarefa de navegação, que responde às ordens de deslocação.

Uma abordagem baseada na aquisição ou adaptação de comportamentos de complexidade crescente que se apoiam sobre os já adquiridos, é uma alternativa que permite construir progressivamente uma solução satisfatória para o controlo de robôs caminantes. Dado que as soluções não são conhecidas à priori, estes comportamentos devem ser adquiridos por aprendizagem. É por isso que nos interessamos, no capítulo seguinte, pela Inteligência Artificial, nomeadamente sobre os tipos de aprendizagem existentes nesta área.

4. INTELIGÊNCIA ARTIFICIAL

Vimos, no capítulo precedente, que certos trabalhos de investigação sobre robôs hexápodes têm como objectivo a melhoria das capacidades mecânicas (geometria das pernas e corpo, número de gdl, etc.), outros a melhoria da percepção do ambiente (por exemplo adição de sensores mais eficazes) e outros a melhoria do modelo de controlo do robô. Este trabalho foi orientado sobre o primeiro eixo de investigação, relativo à melhoria das capacidades mecânicas de um robô com pernas. Neste quadro as técnicas de aprendizagem assumem particular importância.

Ao longo da história iniciou-se o estudo da inteligência através dos filósofos. Actualmente extrapolou-se o campo filosófico e outros campos do saber humano começaram a estudá-la: a engenharia, a psicologia, a pedagogia, a neurologia, a linguística, a computação, etc.

A Inteligência Artificial (IA) tem-se destacado com o objectivo prático de simular a inteligência [67]. Pelas características de processar uma informação e juntamente com a complexidade de calcular esta simulação, a IA encontrou nos computadores o seu aliado principal [68]. O mesmo autor afirma que a IA procura entender a mente humana e imitar o seu comportamento. As questões que surgem são muitas:

- Como é que ocorre o pensamento, como surgem as ideias?
- Como é que a memória, os sentidos e a linguagem ajudam no desenvolvimento da inteligência?
- Como é que a mente processa informações e tira conclusões?

As técnicas que se apoiam na IA necessitam de uma grande quantidade de conhecimentos e de manipulação de símbolos segundo Ganascia [67]. Esta base deve ter as seguintes características: representação, modificação e ampliação.

O que o homem chama de inteligência, é o resultado de milhões de anos de evolução. A complexidade envolvida na formação, no funcionamento e na tentativa de imitação é muito grande. Muitas conclusões a este respeito são relevantes, mas há muito a ser desvendado. Não existe uma teoria completa sobre a mente humana e os processos de raciocínio. Para Simons [69] o objectivo da IA é estudar a inteligência tratando-a como um fenómeno e tentando modelá-la. A expressão que trata de máquinas que são capazes de “comportamentos inteligentes” não quer dizer que se chega ao nível da inteligência humana, mas sim que possui características de uma capacidade própria de tomar decisões, “imitando” uma pessoa.

Muitos investigadores afirmam que, embora estudos e implementações tenham sido realizados nesta área e já seja possível reproduzir algumas características da inteligência humana através de programas de computador, não se pode afirmar com certeza que um dia seja possível construir uma máquina que pense como o ser humano [67].

A capacidade de interagir com o ambiente por meio de capacidades cognitivas e conotativas, ou seja, movimentar-se, reconhecer imagens e objectos e reconhecer sons, são as características principais da inteligência [69]. Existe um esforço, principalmente no campo da robótica, no sentido de implementar essas capacidades nas máquinas inteligentes, de modo a propiciar uma maior interacção com o meio e desenvolver padrões de inteligência envolvidos na aquisição do conhecimento, reconhecimento e aprendizagem.

A IA desenvolveu-se num estágio inicial da investigação sobre a inteligência. Ainda existe muita discussão contra, ou a favor da possibilidade (ou não), de se criarem máquinas com uma inteligência igual à do homem. Deve-se levar em consideração que a mesma é uma ciência ainda em fase de estruturação e, a cada ano, surgem novos campos de investigação. Não é possível afirmar até onde a IA poderá chegar no propósito de criar máquinas capazes de pensar.

As características envolvidas como, por exemplo, consciência, inconsciência, factores cognitivos e até mesmo instintivos estão envolvidos no processo de pensar. Alguns aspectos da inteligência já podem ser implementados e utilizados para auxiliar ou substituir o homem na realização de tarefas. E mesmo quando se chegar à modelagem e ao entendimento dos factores envolvidos nesse processo, ainda haverá o desafio de encontrar formas de implementá-los no computador e desenvolver equipamentos adequados para otimizar essa implementação [67].

4.1. PERSPECTIVA HISTÓRICA

Na década de 40 do século passado, época marcada pela II Guerra Mundial, surgiram os primeiros estudos sobre IA [70]. A necessidade básica da guerra era a produção de artefactos para análise de balística, quebra de códigos e máquinas de cálculos capazes de suportar o projecto de uma bomba atómica. Neste período a tecnologia evoluiu até surgirem os primeiros projectos de construção de computadores, assim chamados por serem máquinas utilizadas para fazer cálculos. Após o término da Guerra o computador deixou de ser objecto militar e invadiu empresas, indústrias, universidades.

Desde o aparecimento dos primeiros computadores electrónicos (1943 - Colossus, na Inglaterra e 1946 - ENIAC, nos Estados Unidos) (Figura 33) até ao aparecimento dos microcomputadores (na década de 70 do século XX), esta área recebeu grandes investimentos.

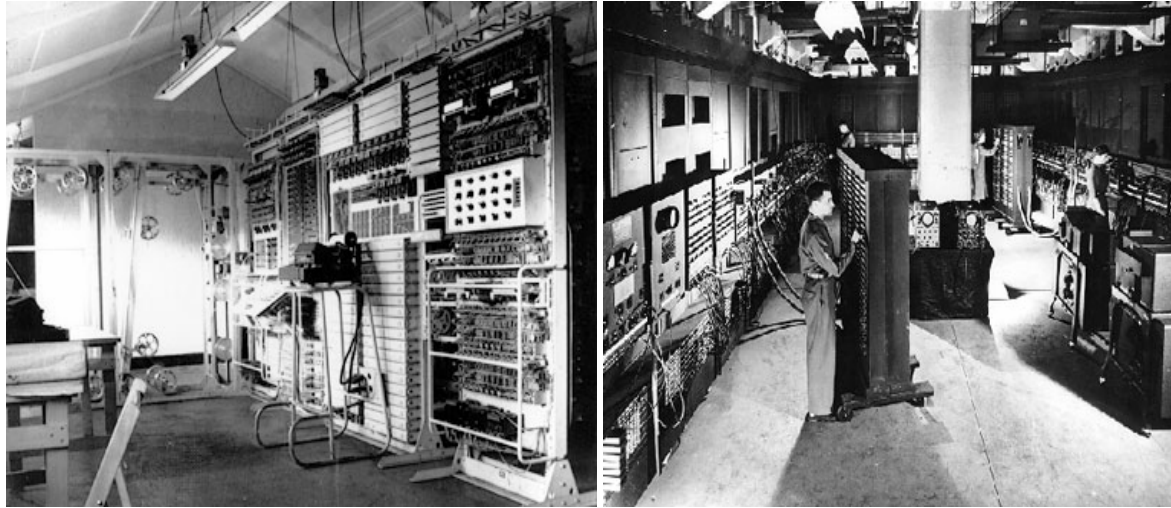


Figura 33 Computadores Colossus (esquerda) e ENIAC (direita)

Em 1956, nos Estados Unidos, John McCarthy reuniu, numa conferência proferida ao *Dartmouth College*, na Universidade de *New Hampshire*, vários investigadores de renome para estudar o que foi denominado de Inteligência Artificial, expressão utilizada para designar um tipo de inteligência construída pelo homem para dotar a máquina de comportamentos inteligentes [71].

Com a abertura e estruturação desse novo campo do conhecimento, o fenómeno da inteligência começou a ser investigado de forma variada e intensa. Os métodos de raciocínio utilizados pelo ser humano foram modelados e o próximo passo era implementá-los no computador utilizando a IA.

Isso tornou-se possível em grande parte graças ao desenvolvimento dos sistemas especialistas, da lógica "fuzzy" e das redes neuronais. É uma actividade que envolve investigadores de instituições governamentais, militares, industriais e universitárias de todo o mundo e muito investimento tem sido feito neste sentido [67].

Apesar das pesquisas nesta área não terem parado, foi apenas na década de 80, que o físico e biólogo do Instituto de Tecnologia da Califórnia, John Hopfield, conseguiu recuperar a credibilidade da utilização das redes neuronais.

4.2. APLICAÇÕES DA IA

Existem vários campos de estudo dentro da IA com o intuito de prover algum raciocínio no computador, bem como certa capacidade de aprendizagem e de auto-ajuste. A seguir são apresentados alguns campos de aplicação [72].

O processamento de linguagem natural, é uma área de estudo que visa a construção de programas capazes de compreender a linguagem natural (interpretação) e gerar textos. A geração de linguagem natural consiste na produção de textos por um programa a partir de um conteúdo semântico representado internamente no próprio programa. O principal objectivo é aperfeiçoar a comunicação entre as pessoas e os computadores.

O reconhecimento de padrões, é uma área de pesquisa que possui grandes aplicações já implementadas e a funcionar. A capacidade de reconhecimento de padrões permite ao programa reconhecer a fala em linguagem natural, os caracteres digitados e a escrita (por exemplo, assinatura). Os *scanners*, por exemplo, utilizam programas de reconhecimento óptico desenvolvidos tendo por base pesquisas em IA. A visão computacional procura desenvolver formas do computador trabalhar com a visão bidimensional e tridimensional.

Os jogos computadorizados são um grande sucesso, ainda mais quando exibem um tipo de inteligência capaz de desafiar as habilidades do jogador. O jogo de xadrez, por exemplo, foi utilizado para as primeiras experiências em programação do raciocínio artificial, através do qual o computador se tornou capaz de analisar milhões de jogadas por segundo para tentar derrotar o adversário. Além de analisar as jogadas, os programas utilizam um método heurístico que consiste na utilização de uma árvore de procura, que possui ramificações a partir de certos nós representando pontos de decisão no caminho a tomar, com um certo número de etapas, para chegar a um objectivo.

A robótica, que é o campo de estudo voltado para desenvolver métodos de construção de máquinas que possam interagir com o meio, como por exemplo: ver, ouvir, reagir aos estímulos sensoriais, realizar tarefas em ambientes hostis ao homem e tarefas repetitivas.

4.3. MODELOS DE IA

São considerados como os principais modelos de IA os seguintes: os algoritmos genéticos, a programação evolutiva, os sistemas difusos, os sistemas baseados em conhecimento, o raciocínio baseado em casos, a programação genética e as redes neuronais [67]. De seguida é apresentada uma breve descrição de cada um destes modelos.

Os Algoritmos Genéticos são um modelo usado para a aprendizagem da máquina, inspirado no livro *Origem das Espécies*, através da Selecção Natural, escrito pelo naturalista inglês Charles Darwin (1809-1882), criador da teoria evolucionista segundo a qual somente os mais aptos sobrevivem.

Este é um método utilizado pelos Algoritmos Evolutivos, que inclui o estudo dos algoritmos genéticos, estratégia de evolução, programação evolutiva e sistemas classificatórios. Foram criados por John Holland (1975), e têm como objectivo emular operadores genéticos específicos, como cruzamento, mutação e reprodução, da mesma forma como é observado na natureza. Isso é feito criando-se, dentro da máquina, uma população de indivíduos representados por cromossomas. Os indivíduos passam por um processo simulado de evolução, selecção e reprodução, gerando uma nova população.

É com este método que desenvolvemos o nosso trabalho, pelo que uma explicação mais pormenorizada irá ser dada no próximo capítulo.

A Programação Evolutiva é o campo da IA concebido por Lawrence J. Fogel (1960). Assemelha-se aos algoritmos genéticos, sendo dada maior ênfase na relação comportamental entre os parentes e os seus descendentes. As soluções para os problemas são obtidas por meio de tentativas e transmitidas para a nova população (simulada em programas).

Os Sistemas Difusos ou Lógica Nebulosa foram estruturados por Lofti Zadeh, da Universidade da Califórnia, no ano de 1965. É uma metodologia que serve para representar, manipular e modelar informações incertas.

Os Sistemas Baseados em Conhecimento implementam comportamentos inteligentes de especialistas humanos. O Raciocínio Baseado em Casos é o campo de estudo da IA que

utiliza uma grande biblioteca de casos para consulta e resolução de problemas. Os problemas actuais são resolvidos através da recuperação e consulta de casos já solucionados e da conseqüente adaptação das soluções encontradas. Como exemplo, refira-se o Sistema CASEY que faz o diagnóstico em pacientes cardíacos baseado na consulta de arquivos de pacientes com o mesmo diagnóstico.

A Programação Genética é um campo de estudo da IA voltado para a construção de programas que visam imitar o processo natural da genética. Trabalha com métodos de procura aleatória.

As redes neuronais possuem várias denominações [71], de entre elas, redes neurais, modelo conexionista, neurocomputação, modelo de processamento paralelo distribuído, sistemas neuromórficos e computadores biológicos. As redes neuronais artificiais são definidas como redes massivamente paralelas e interligadas de elementos simples, com organização hierárquica. Estes elementos devem interagir com objectos do mundo real, da mesma maneira que o sistema nervoso biológico.

Diversas áreas de investigação, dos mais variados assuntos, estão a ser influenciadas pelo desenvolvimento da IA [69]. Por ser uma ciência relativamente nova possui um potencial muito grande e ainda inexplorado, pelo que certamente muitas outras áreas de investigação e de aplicações deverão surgir nas próximas décadas.

As implementações que utilizam a IA caracterizam-se pelo facto de objectivar numa interacção com o utilizador ou o seu ambiente (sistema) por meio da simulação de atitudes e reacções humanas que envolvem a compreensão, análise, planeamento, tomada de decisão e aprendizagem. Estes já são utilizados com significativa eficiência, substituindo ou auxiliando o homem, em tarefas de aprendizagem, diagnóstico médico, predições económicas, tradução, reconhecimento de padrões de imagens e voz, automação de processos industriais e controlo de qualidade.

4.4. INTELIGÊNCIA E CONHECIMENTO

Para implementar a inteligência numa máquina, torna-se necessário que sejam analisados todos os aspectos relativos à maneira como o ser humano adquire e desenvolve esta

característica. Dentro desse contexto, o conhecimento é o que faz com que seja possível o encadeamento e desenvolvimento da inteligência. Para incorporar estes conhecimentos devem ser observados alguns detalhes [72].

O conhecimento é volumoso, isto quer dizer, possui diversos aspectos, características e detalhes. À medida que se procura aprofundar mais um determinado assunto, mais a pessoa obtém informações. Muitas vezes tem-se apenas o conhecimento, mas não se sabe como foi adquirido, ou então não se sabe explicá-lo. De facto, muitas vezes não há consciência do conhecimento que se possui. Ele é também uma aquisição do indivíduo, ou seja, duas pessoas não adquirem o mesmo conhecimento de uma forma idêntica. Cada uma faz a sua própria interpretação. Pode-se afirmar que duas pessoas possuem o mesmo conhecimento genérico, mas não com conceitos idênticos.

Para se utilizar o conhecimento no computador é necessário estruturar o que será utilizado pelo sistema, trabalhando com uma representação do conhecimento. Ao contrário do conhecimento em si, a representação tem as seguintes características, segundo Rich e Knight [72]: é generalizável, ao contrário do conhecimento em si, que é individual. Nem todas as aplicações de IA envolvem conhecimento estruturado, mas necessitam, porém, de pré-processar o conhecimento existente.

Na aplicação deste trabalho é importante a aquisição automática do conhecimento que permitiria a um sistema a sua auto-evolução. O facto deste sistema aprender com exemplos significa não haver necessidade de pessoas estruturarem o conhecimento para colocá-lo no computador, executando, com isso, mais uma tarefa humana. Os Sistemas de IA podem ser utilizados em programas tutoriais, destinados a ensinar ao utilizador um determinado raciocínio em relação à computação ou a outras ciências.

4.5. TÉCNICAS DE IA

Uma técnica de IA é um método que explora o conhecimento e deve ser representado de tal forma que capture generalizações, seja compreendido pelas pessoas que o fornecem, seja utilizado em inúmeras situações e seja facilmente modificado. De entre as técnicas de IA, destacam-se duas: a procura heurística e o uso do conhecimento.

A heurística é um procedimento para resolver problemas através de um enfoque intuitivo muitas vezes racional. É através da interpretação da própria estrutura do problema que se obtém uma solução adequada. Para os investigadores da IA, as heurísticas são critérios, métodos ou princípios para decidir, entre vários cursos de acção alternativos, aquele que parece mais efectivo para atingir algum objectivo [73].

Para solucionar os problemas encontrados na IA precisa-se de uma grande quantidade de conhecimento e também de alguns mecanismos para a manipulação deste, a fim de criar soluções para novos problemas [72].

A geração de certa inteligência pressupõe a aquisição, armazenamento e inferência de conhecimento. Para armazená-lo é essencial que se possa representá-lo. A maior parte do esforço em IA tem-se concentrado em procurar ou aperfeiçoar formalismos para essa representação. A representação através de casos é uma tecnologia emergente para representação e processamento de conhecimento. Usa experiência passada, acumulando casos e tentando descobrir, por analogia, soluções para outros problemas [71]. A construção de um modelo deste tipo segue os seguintes passos:

- Aquisição e Representação: seleccionar os melhores dentro de um conjunto de casos disponíveis;
- Indexação: selecciona características da descrição dos casos que permitam a recuperação eficiente dos casos relevantes ao problema;
- Recuperação: mecanismos para medir o grau de similaridade entre casos e problemas;
- Adaptação: estratégias para adaptar a solução ou soluções a casos recuperados e mecanismos para avaliar a qualidade da solução;
- Aprendizagem: um problema resolvido e com solução validada pode ser acrescentado à base de casos [74].

4.6. CONCLUSÃO

Neste capítulo, apresentaram-se, resumidamente, alguns métodos de IA como os métodos de Algoritmos Evolutivos, que inclui o estudo dos algoritmos genéticos, que foram criados por John Holland, a Programação Evolutiva que foi o campo da IA concebido por Lawrence J. Fogel, os Sistemas Difusos ou Lógica Nebulosa estruturados por Lofti Zadeh, entre outros.

Seguidamente focalizamo-nos sobre as técnicas da IA; uma técnica de IA é um método que explora o conhecimento e deve ser representado para que seja compreendido pelas pessoas que o fornecem. De entre as técnicas de IA, destacam-se três: a procura heurística, a abstracção e o uso do conhecimento.

No capítulo seguinte apresenta-se uma descrição das funcionalidades e características dos Algoritmos Genéticos. É neste algoritmo que se baseia o desenvolvimento do nosso trabalho.

5. ALGORITMOS GENÉTICOS

Os AGs fazem parte da área dos Sistemas Inspirados na Natureza, simulando os processos naturais e aplicando-os em problemas reais. São métodos generalizados de procura e optimização que simulam os processos naturais de evolução, aplicando a ideia "darwiniana" de selecção.

Estes algoritmos são baseados nos processos genéticos dos organismos biológicos, codificando uma possível solução de um problema num "cromossoma", composto por uma cadeia de bits e caracteres. Estes cromossomas representam indivíduos que são levados ao longo de várias gerações, na forma similar aos problemas naturais, evoluindo de acordo com os princípios de selecção natural de sobrevivência dos mais aptos, descritos pela primeira vez por Charles Darwin (Figura 34) no seu livro "Origem das Espécies" [75].

Na natureza os indivíduos competem entre si por recursos como comida, água e refúgio. Adicionalmente, entre os animais de uma espécie, aqueles que não obtêm êxito tendem a ter um número reduzido de descendentes, tendo portanto menor probabilidade dos seus genes serem propagados ao longo de sucessivas gerações. A combinação entre os genes

dos indivíduos que perduram na espécie, podem produzir um novo indivíduo, melhor adaptado às características de seu meio ambiente.

Aos mais aptos é dada a oportunidade de se reproduzirem mediante cruzamentos com outros indivíduos da população, produzindo descendentes com características de ambas as partes. Se um AG for desenvolvido correctamente, a população (conjunto de possíveis respostas) irá convergir para uma solução óptima (ou quase óptima) do problema proposto.

Os AGs são muito eficientes para a procura de soluções óptimas, numa grande variedade de problemas, pois não impõem muitas das limitações encontradas nos métodos de procura tradicionais.

5.1. EVOLUÇÃO HISTÓRICA

No Século XIX, os naturalistas acreditavam que cada espécie havia sido criada separadamente por um ser supremo ou através de geração espontânea. O trabalho do naturalista Carolus Linnaeus levou a acreditar na existência de uma certa relação entre as espécies. Por outro lado, Thomas Robert Malthus propôs que factores ambientais tais como doenças e carência de alimentos limitavam o crescimento de uma população.

No Século XX após anos de observações e experiências, Charles Darwin (Figura 34) apresentou em 1858 a sua teoria de evolução através da selecção natural. Por volta de 1900, a teoria da evolução moderna combina a genética e as ideias de Darwin e Wallace sobre a selecção natural, criando o princípio básico de Genética Populacional.

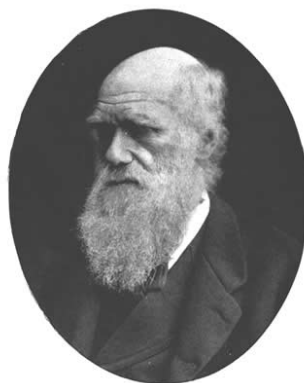


Figura 34 Charles Darwin (1809-1882)

A variabilidade entre indivíduos numa população de organismos que se reproduzem sexualmente é produzida pela mutação e pela recombinação genética. Este princípio foi desenvolvido durante os anos 30 e 40 do século XX, por biólogos e matemáticos de importantes centros de investigação. Nos anos 50 e 60 muitos biólogos começaram a desenvolver simulações computacionais de sistemas genéticos.

Foi John Holland [76] quem começou, seriamente, a desenvolver as primeiras investigações no tema. Holland foi gradualmente refinando as suas ideias e, em 1975, publicou o seu livro "*Adaptation in Natural and Artificial Systems*", hoje considerado a "Bíblia" dos AGs. Nos anos 80, David Goldberg (aluno de Holland) consegue o primeiro sucesso numa aplicação industrial de AGs.

Desde então, estes algoritmos têm sido aplicados com sucesso nos mais diversos problemas de optimização [77], [78], [79] e aprendizagem das máquinas [80], [81], [82].

5.2. CARACTERÍSTICAS GERAIS

O AG é um algoritmo que tenta imitar a natureza no que toca à sobrevivência do mais apto. Foi apresentado pela primeira vez por John Holland [76] da Universidade de Michigan em 1975 e tem sido adaptado para a resolução de diversos problemas de optimização e aplicações de controlo [77], [80], [83]. Para se obter a máxima eficiência e/ou eficácia do AG é necessário codificar adequadamente os elementos envolvidos no problema [84], [85].

Os AGs são algoritmos de optimização global que utilizam uma estratégia de procura paralela e estruturada, mas aleatória, focada na procura de pontos de "alta aptidão", ou seja, pontos nos quais a função a ser minimizada (ou maximizada) tem valores relativamente baixos (ou altos). Os AGs exploram informações históricas para encontrar novos pontos de procura onde são esperados melhores desempenhos. Isto é feito através de processos iterativos, onde cada iteração é chamada de geração [86], [87], [88].

O ponto de partida para a utilização dos AGs, como ferramenta para a solução de problemas, é a representação destes problemas de maneira a que os AGs possam trabalhar adequadamente sobre eles. Tradicionalmente, os indivíduos são representados

geneticamente por vectores binários, onde cada elemento de um vector denota a presença (1) ou ausência (0) de uma determinada característica: o seu genótipo. Os elementos podem ser combinados formando as características reais do indivíduo: o seu fenótipo [85], [89], [90].

É necessário um conjunto de operações para que, dada uma população, se consigam gerar populações sucessivas que (espera-se) melhorem a sua aptidão ao longo do tempo. Estes operadores são: cruzamento (*crossover*) e mutação (*mutation*) [79], [88], [90]. Eles são utilizados para assegurar que a nova geração seja totalmente nova, mas possui, de alguma forma, características dos seus pais.

Para prevenir que os melhores indivíduos não desapareçam da população pela manipulação dos operadores genéticos, eles podem ser automaticamente colocados na próxima geração, através da reprodução elitista [91].

Esse ciclo é repetido um determinado número de vezes. Na Figura 35 é mostrado um exemplo de um AG. Durante esse processo, os melhores indivíduos, assim como alguns dados estatísticos, podem ser recolhidos e armazenados para avaliação.

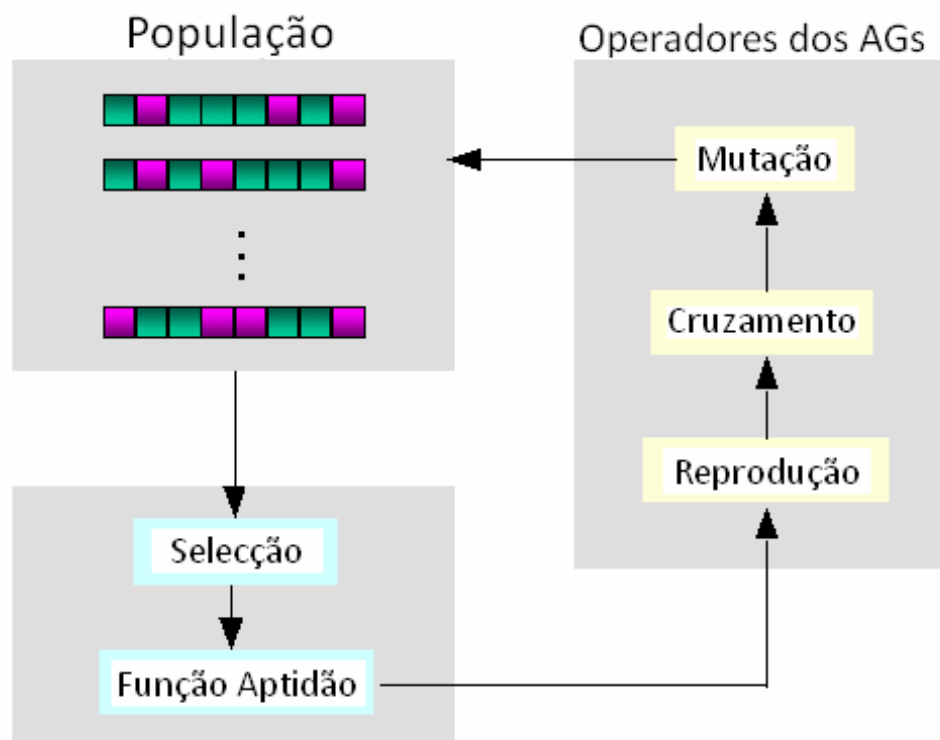


Figura 35 Representação de um ciclo de um AG

5.3. OPERADORES GENÉTICOS

Os operadores genéticos transformam a população através de sucessivas gerações, estendendo a procura até chegar a um resultado satisfatório. Estes operadores são necessários para que a população se diversifique e mantenha características de adaptação adquiridas pelas gerações anteriores [90] [92].

5.3.1. OPERADOR SELECÇÃO

A função do operador selecção (Figura 36) é fazer a selecção dos pais para aumentar a probabilidade de reproduzir membros da população que tenham bons valores da função aptidão. Uma vez seleccionados os pais, será utilizado um operador genético, como por exemplo o cruzamento [84], [87]. Às vezes a escolha de um mecanismo de selecção adequado permite um tratamento matemático mais rigoroso.

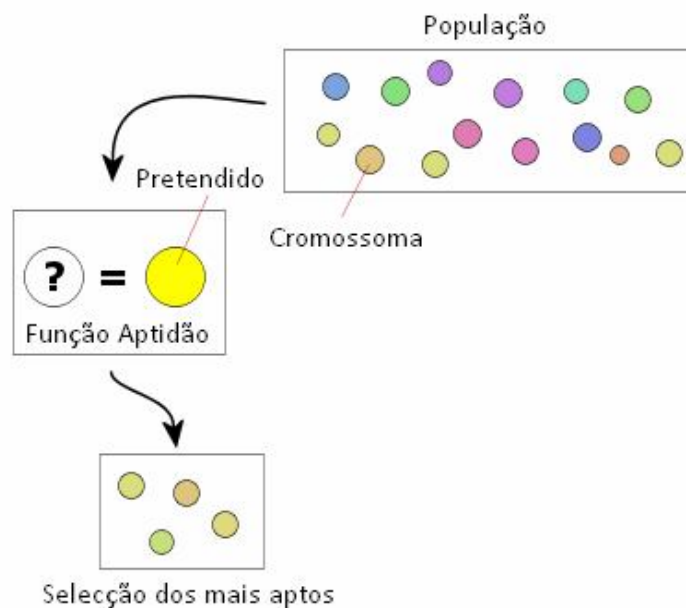


Figura 36 Operador selecção de um AG

A selecção pode ser feita pelo seguinte procedimento prático; primeiro calcula-se uma coluna de aptidões. Em seguida, gera-se um número aleatório n (retirado de uma distribuição normal) no intervalo $[0, Soma\ Total]$, onde *Soma Total* é a soma de todas as aptidões. Por fim, o cromossoma seleccionado é o primeiro (segundo a tabela de cima para baixo) que possui aptidão acumulada maior ou igual a n .

Os mesmos passos são repetidos até preencher a população intermédia com N cromossomas (tamanho igual à população inicial). Este procedimento é conhecido como a Roda da Roleta [93], [94].

Várias alternativas têm sido propostas para definir a aptidão, sendo que a mais simples iguala a aptidão ao valor da função objectivo. Mas a aptidão, definida desta forma, pode assumir valores negativos e o algoritmo Roda da Roleta não funciona com aptidões negativas. Além disso, pode gerar problemas tais como a convergência prematura [94], [95].

É possível, no entanto, dispensar o Algoritmo Roda da Roleta e utilizar a Selecção por Torneio. Neste caso, são escolhidos aleatoriamente (com probabilidades iguais) n cromossomas da população, e o cromossoma com maior aptidão é seleccionado para a população intermédia. O processo repete-se até preencher a população intermédia. Um valor geralmente utilizado é $n = 3$ [93].

Outra forma de definir a aptidão é pelo ordenamento dos cromossomas consoante o valor da função objectivo. O primeiro cromossoma (o mais próximo do objectivo) recebe uma aptidão arbitrária de x e o último cromossoma do ordenamento, vê-lhe atribuído o valor de y . Os restantes cromossomas da população recebem valores de aptidão obtidos através da interpolação dos dois extremos por uma recta.

5.3.2. OPERADOR CRUZAMENTO

O operador cruzamento é o responsável pela recombinação de características dos pais durante a reprodução (Figura 37), permitindo que as próximas gerações herdem essas características. Ele é considerado o operador genético predominante, por isso é aplicado com probabilidade dada pela taxa de cruzamento, que deve ser maior que a taxa de mutação [77] [91] [96].

O cruzamento é aplicado com uma dada probabilidade a cada par de cromossomas seleccionados. Na prática, esta probabilidade, denominada de taxa de cruzamento, varia entre 60% e 90% [94], [95]. Não ocorrendo o cruzamento, os filhos serão iguais aos pais (isto permite que algumas soluções sejam preservadas). Isto pode ser implementado,

gerando números pseudo-aleatórios no intervalo $[0, 1]$. Assim, o cruzamento só é aplicado se o número gerado for menor que a taxa de cruzamento.

5.3.2.1. CRUZAMENTO DE N PONTOS E UNIFORME

Os tipos de operadores cruzamento mais conhecidos para cadeias de bits são o de n pontos e o uniforme [94], [95]. No cruzamento de um ponto, tal como apresentado na Figura 37, é escolhido um ponto de cruzamento e a partir deste ponto as informações genéticas dos pais serão trocadas. As informações anteriores a este ponto num dos pais são ligadas às informações posteriores a este ponto no outro pai.

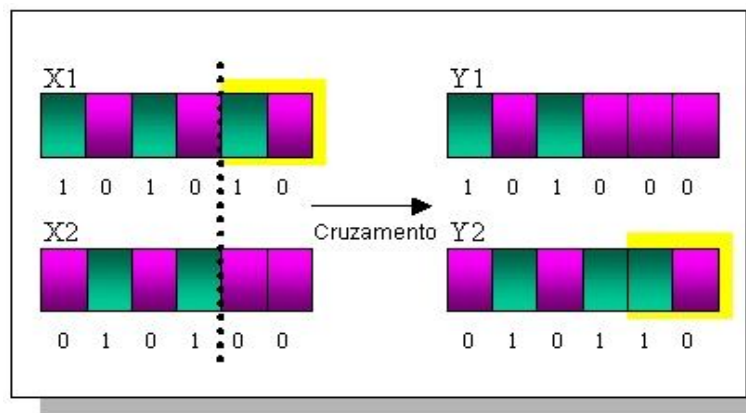


Figura 37 Operador cruzamento de um ponto de um AG

O cruzamento multi-ponto [94], [95] é uma generalização desta ideia de troca de material genético através de pontos, onde vários pontos de cruzamento podem ser utilizados. O de dois pontos é apresentado na Figura 38; os dois pontos de corte são escolhidos aleatoriamente, e as secções entre os dois pontos são trocadas entre os pais. O cruzamento de n pontos mais usado tem sido o de dois pontos.

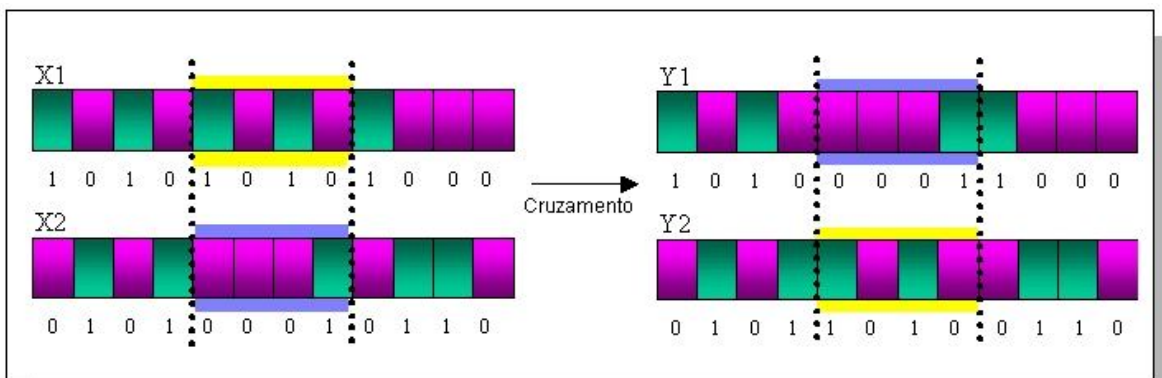


Figura 38 Operador cruzamento de dois pontos de um AG

O cruzamento uniforme [96] é apresentado na Figura 39. Para cada par de pais é gerada uma máscara de bits aleatórios. Se o primeiro bit da máscara possui o valor um, então o primeiro bit do pai1 é copiado para o primeiro bit do filho1. Caso contrário, o primeiro bit do pai2 é copiado para o primeiro bit do filho1. O processo repete-se para os restantes bits do filho1. Na geração do filho2 o processo é invertido, ou seja, se o bit da máscara é um, então será copiado o bit do pai2.

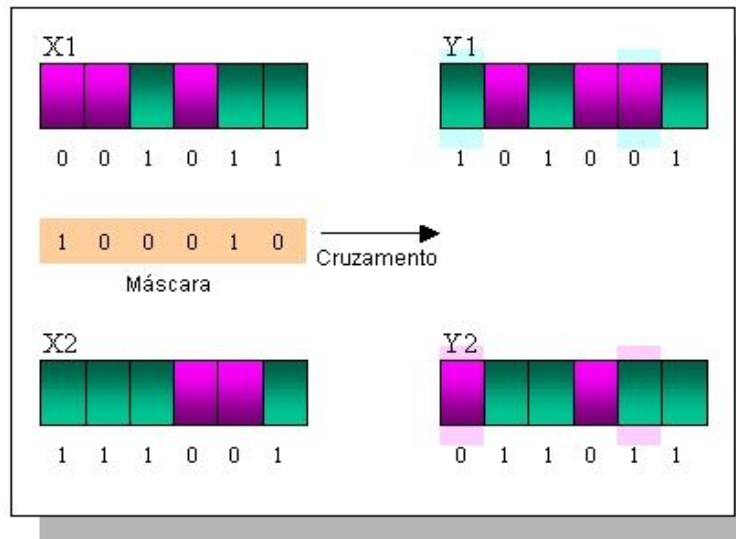


Figura 39 Operador cruzamento uniforme de um AG

5.3.3. OPERADOR MUTAÇÃO

Os operadores de mutação são necessários para a introdução e manutenção da diversidade genética da população, alterando arbitrariamente um ou mais componentes de uma estrutura escolhida, fornecendo assim meios para a introdução de novos elementos na população (Figura 40). Desta forma, a mutação assegura que a probabilidade de se chegar a qualquer ponto do espaço de procura nunca será zero.

O operador de mutação [88] é aplicado aos indivíduos com uma probabilidade dada pela taxa de mutação. Geralmente é utilizada uma taxa de mutação pequena, pois este é um operador genético secundário.

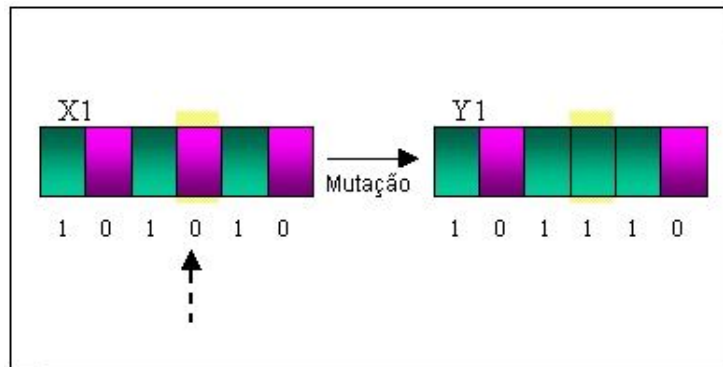


Figura 40 Operador mutação de um AG

Após a operação de cruzamento, é aplicado o operador de mutação, com uma dada probabilidade, em cada bit dos filhos. O operador de mutação inverte os valores de bits, ou seja, muda o valor de dado bit de um para zero ou de zero para um.

A mutação melhora a diversidade dos cromossomas na população. No entanto, por outro lado, destrói a informação contida no cromossoma; logo, deve ser utilizada uma taxa de mutação pequena (normalmente entre 0,1% e 5%) [94], [95], mas suficiente para assegurar a diversidade.

5.3.4. ELITISMO

Vale a pena observar que o melhor cromossoma pode ser perdido de uma geração para outra devido ao corte do cruzamento ou à ocorrência de mutação. Portanto, é interessante transferir o melhor cromossoma de uma geração para outra sem alterações. Esta estratégia é denominada Elitismo, sendo muito comum nos AGs tradicionais. O elitismo foi proposto por DeJong [97], num dos trabalhos pioneiros sobre AGs.

5.4. REPRESENTAÇÃO REAL

5.4.1. REPRESENTAÇÃO BINÁRIA VS. REAL

A representação binária é historicamente importante, uma vez que foi utilizada nos trabalhos pioneiros de Holland [98]. É a representação tradicional, sendo fácil de utilizar e manipular, sendo também simples de analisar teoricamente. Contudo, se um problema tem parâmetros contínuos e o utilizador quer trabalhar com boa precisão numérica, será

necessário armazenar cromossomas longos na memória, uma vez que, para cada ponto decimal acrescentado na precisão, é necessário adicionar bits na cadeia. Quando há muitos parâmetros, obtêm-se longas cadeias de bits que podem fazer o algoritmo convergir lentamente. Além disso, não existe uniformidade nos operadores. Por exemplo, a mutação nos primeiros bits do gene afecta mais a aptidão do que a mutação dos últimos bits do gene.

A representação real (i.e., com ponto flutuante) gera cromossomas menores e é compreendida mais naturalmente pelo ser humano do que a cadeia de bits. Outra vantagem da representação real é a facilidade de criar novos operadores. Existe um grande número deles conforme será mostrado mais à frente.

Vários autores têm discutido qual a melhor representação, a binária ou a real, e muitos deles têm mostrado experiências favoráveis à representação real [99],[100],[101].

5.4.2. CRUZAMENTO PARA REPRESENTAÇÃO REAL

Os operadores convencionais (cruzamento de n pontos e uniforme) são o resultado das adaptações dos operadores utilizados para a representação binária. Os operadores convencionais funcionam bem na representação binária, mas na representação real eles basicamente trocam os valores dos genes e, portanto, não criam informações novas (i.e., novos números reais). Assim, o melhor será usar operadores aritméticos, uma vez que estes realizam algum tipo de combinação linear entre os cromossomas pais.

5.4.2.1. CRUZAMENTO MÉDIA

Dados dois cromossomas $pai1$ e $pai2$, é produzido um cromossoma filho da seguinte forma [102]:

$$filho_i = \frac{(pai1_i + pai2_i)}{2}$$

Uma variação deste cruzamento, o cruzamento média geométrica, é apresentada no ponto seguinte.

5.4.2.2. CRUZAMENTO MÉDIA GEOMÉTRICA

Neste caso cada gene f_i do *filho1* é dado por [102]:

$$f_i = \sqrt{pai1_i \times pai2_i} \quad \text{para } i = 1..l$$

O cruzamento média geométrica tende a levar os genes para o meio do intervalo permitido $[a, b]$ causando perda de diversidade. Este aspecto pode ser melhorado com o cruzamento BLX- α , apresentado no ponto seguinte.

5.4.2.3. CRUZAMENTO BLX- α OU CRUZAMENTO MISTURA

Dados dois cromossomas $pai1$ e $pai2$, é produzido um cromossoma filho da seguinte forma [103]:

$$filho1_i = pai1_i + \beta(pai2_i - pai1_i) \quad \text{onde } \beta \in [-\alpha, 1 + \alpha]$$

O cromossoma BLX- α é ilustrado na Figura 41, na qual foi escolhido um único valor de β para todos os pares de genes. Quando $\alpha=0$ o filho situa-se sobre o intervalo I entre os dois pontos que representam os pais. Por exemplo se $\alpha=0,5$, o intervalo I é estendido (αI) em ambos os lados. O BLX-0,5 balanceia a tendência de gerar filhos próximos do centro do intervalo I evitando a perda de diversidade. O BLX- α tem sido usado com sucesso em muitos trabalhos e é talvez o operador mais utilizado na representação real.

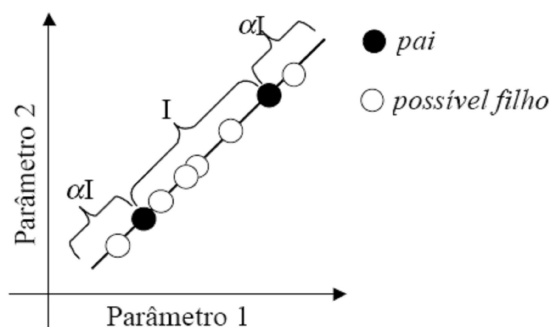


Figura 41 Cruzamento BLX- α

Se o filho gerado for irrealizável, então gerar-se-á um outro filho com um novo valor de β . O processo é repetido até se obter um filho exequível.

Neste exemplo, foi utilizado apenas um β para todos os genes, em alternativa pode-se usar um β diferente para cada par de genes. Neste caso, um possível filho situa-se num qualquer lugar de uma área limitada por um rectângulo (Figura 42).

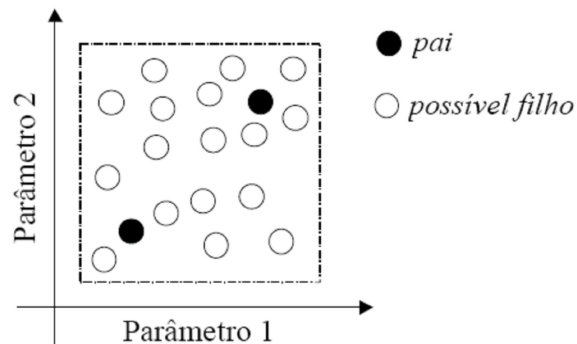


Figura 42 Cruzamento BLX- α com β variável

5.4.2.4. CRUZAMENTO LINEAR

Uma alternativa ao cruzamento BLX- α é o cruzamento linear. Neste caso, dados dois pais, $pai1$ e $pai2$ obtêm-se três filhos $filho1$, $filho2$ e $filho3$ da seguinte forma [103]:

$$filho1_i = 0,5 \times pai1_i + 0,5 \times pai2_i$$

$$filho2_i = 1,5 \times pai1_i - 0,5 \times pai2_i$$

$$filho3_i = -0,5 \times pai1_i + 1,5 \times pai2_i$$

Dos três filhos gerados, apenas o melhor é escolhido, sendo os outros dois descartados.

5.4.2.5. CRUZAMENTO ARITMÉTICO

Neste caso, dados dois cromossomas $pai1$ e $pai2$, são produzidos dois cromossomas $filho1$ e $filho2$ da seguinte forma [104]:

$$filho1_i = \beta \times pai1_i + (1 - \beta) \times pai2_i$$

$$filho2_i = (1 - \beta) \times pai1_i + \beta \times pai2_i \quad \text{onde } \beta \in [0,1]$$

Este operador difere do cruzamento BLX- α por não extrapolar o intervalo entre o $pai1$ e o $pai2$.

5.4.2.6. CRUZAMENTO HEURÍSTICO

Este tipo de cruzamento realiza uma extrapolação linear entre os pais usando a informação da aptidão. Dados dois cromossomas, $pai1$ e $pai2$, em que o $pai1$ é melhor que o $pai2$ em termos de aptidão, é produzido um cromossoma filho da seguinte forma [104]:

$$filho_i = pai1_i + r \times (pai1_i - pai2_i) \quad \text{onde } r \in [0,1]$$

Caso o cruzamento produza um filho irrealizável, gerar-se-á outro número aleatório r , e obtém-se um novo filho. Se em t tentativas o filho continuar irrealizável, então o cruzamento pára sem produzir o filho.

5.4.2.7. CRUZAMENTO SIMPLES

É uma variação do cruzamento convencional de um ponto adaptado para a representação real, ou seja, são apenas trocados genes entre os cromossomas, não existe nenhuma alteração do valor de nenhum gene [104].

5.4.3. MUTAÇÃO PARA REPRESENTAÇÃO REAL

5.4.3.1. MUTAÇÃO UNIFORME

A mutação uniforme efectua a simples substituição de um gene por um número aleatório [103], ou seja, dado um cromossoma pai com o n ésimo gene seleccionado para mutação, o valor do gene passa a ter um valor aleatório compreendido entre os limites do intervalo permitido para o gene seleccionado.

5.4.3.2. MUTAÇÃO GAUSSIANA

Este tipo de mutação efectua a substituição de um gene por um número aleatório de uma distribuição normal [103]. Dado um cromossoma pai com o n ésimo gene seleccionado para mutação, é produzido um cromossoma filho tendo em conta a média da distribuição normal (valor antes de aplicar a mutação) e o desvio padrão. Como alternativa, pode-se diminuir o valor do desvio padrão à medida que o número de gerações aumenta.

5.4.3.3. MUTAÇÃO CREEP

Neste caso é adicionado ao gene um pequeno número aleatório obtido de uma distribuição normal (com média zero e desvio padrão pequeno) ou de uma distribuição uniforme. Em alternativa, a mutação *creep* pode ser realizada multiplicando o gene por um número aleatório próximo de um [104].

O número aleatório deve ser suficientemente pequeno para que cause apenas uma ligeira perturbação no cromossoma, porque ao estar perto do ponto máximo, tal perturbação pode afastá-lo rapidamente do topo.

A taxa de mutação *creep* pode ser relativamente alta, visto que esta é usada para explorar localmente o espaço de busca (a mutação *creep* não é muito destrutiva).

5.4.3.4. MUTAÇÃO LIMITE

A mutação limite efectua a substituição do gene por um dos limites do intervalo permitido $[a, b]$. Assim, dado um cromossoma pai com o n ésimo gene seleccionado para mutação, o valor deste passa a ser o valor do limite inferior ou o valor do limite superior do intervalo de restrição do valor desse mesmo gene [104].

Note-se que este operador leva os genes para os limites dos intervalos permitidos. Isto é feito para evitar a perda de diversidade dos filhos gerados pelo cruzamento aritmético, que tende a trazer os genes para o centro dos intervalos permitidos.

5.4.3.5. MUTAÇÃO NÃO UNIFORME

Na mutação não uniforme é efectuada a simples substituição de um gene por um número extraído de uma distribuição não uniforme. Dado um cromossoma pai com o n ésimo gene seleccionado para mutação, esse gene é substituído por um número retirado de uma qualquer distribuição não uniforme [104].

5.5. APLICAÇÕES DOS AGs

Os sistemas adaptativos tentam resolver problemas acumulando conhecimentos sobre o problema e utilizando estas informações para gerar soluções aceitáveis. Estes problemas

encontram-se, tipicamente, nas áreas de configuração de sistemas complexos, gestão de tarefas, selecção de rotas e outros problemas de optimização e aprendizagem das máquinas [105], [106].

Entre as aplicações dos Algoritmos Genéticos, podemos citar:

- Descodificação de código;
- Projecto de sistemas de distribuição de água;
- Distribuição de topologias de redes de computadores;
- Projecto de circuitos electrónicos;
- Aprendizagem de comportamentos em robôs;
- Análise linguística;
- Optimização de infra-estruturas de comunicações móveis;
- Optimização de sistemas de compressão de dados;
- Distribuição de equipamentos num determinado espaço;
- Maximização de volumes de produção diminuindo o tempo de operação;
- Selecção do melhor modelo matemático que descreve sistemas biológicos;
- Software de engenharia;
- Optimização de rotas;
- Procura de erros de hardware.

5.6. PARÂMETROS GENÉTICOS

É também importante, analisar a maneira como alguns parâmetros influenciam o comportamento dos Algoritmos Genéticos, para que se possa estabelecê-los conforme as necessidades do problema e dos recursos disponíveis.

5.6.1. TAMANHO DA POPULAÇÃO

O tamanho da população afecta o desempenho global e a eficiência dos AGs. Uma população pequena oferece uma pequena cobertura do espaço de procura, causando uma queda no desempenho [107], [108], [109]. Uma grande população fornece uma melhor cobertura do domínio do problema e previne a convergência prematura para soluções locais. No entanto, com uma grande população tornam-se necessários recursos computacionais maiores, ou seja, um maior tempo de processamento do problema.

5.6.2. TAXA DE CRUZAMENTO

Quanto maior for esta taxa, mais rapidamente serão introduzidas novas estruturas na população. No entanto, isto pode gerar um efeito indesejado pois a maior parte da população será substituída podendo ocorrer perda de estruturas de alta aptidão. Com um valor baixo, o algoritmo pode tornar-se muito lento [84] [88].

5.6.3. TAXA DE MUTAÇÃO

Uma baixa taxa de mutação previne que uma dada posição fique estagnada num valor, além de possibilitar que se chegue a qualquer ponto do espaço de procura. Com uma taxa muito alta a procura torna-se essencialmente aleatória.

5.7. ASPECTOS AVANÇADOS

Neste item, serão comentados alguns aspectos práticos dos Algoritmos Genéticos considerados importantes por diversos autores [94],[95],[97],[99],[100].

5.7.1. POPULAÇÃO INICIAL

A população inicial pode ser gerada de várias maneiras. Se uma população inicial pequena for gerada aleatoriamente, provavelmente, algumas regiões do espaço de busca não serão representadas.

Este problema pode ser minimizado gerando a população inicial de maneira uniforme (i.e., com pontos igualmente espaçados, como se preenchessem uma grelha no espaço de

busca). Outra alternativa é gerar a primeira metade da população aleatoriamente e a segunda metade a partir da primeira, invertendo os bits. Isto garante que cada posição da cadeia de bits tenha um representante na população com os valores zero e um.

Pode ser interessante usar uma população inicial maior que a utilizada nas gerações subsequentes, visando melhorar a representação do espaço de busca. Uma técnica denominada *seeding* pode ser útil em muitos problemas práticos. Consiste em colocar, na população inicial, soluções encontradas por outros métodos de optimização. Isto garante que a solução gerada pelo AG não seja pior que as soluções geradas por estes métodos.

5.7.2. FUNÇÃO OBJECTIVO

Em alguns problemas a função objectivo pode ser bastante complicada, exigindo um alto custo computacional. Por exemplo, existem problemas em que para avaliar um cromossoma, é necessária uma simulação completa do processo, o que pode chegar a consumir horas.

Para lidar com tais funções objectivo, alguns autores propõem cuidados a ser tomados para não avaliar cromossomas idênticos mais de uma vez, reutilizando, deste modo, a avaliação feita anteriormente. Isto pode ser feito de várias maneiras [100]:

- Evitando gerar cromossomas idênticos na população inicial;
- Verificando se foi aplicado o *crossover* ou mutação nos pais; caso não tenha sido, os filhos são iguais aos pais;
- Observando se o filho é igual a um dos pais;
- Mantendo a população com todos os cromossomas distintos entre si, o que também ajuda na manutenção da diversidade;
- Antes de avaliar um filho verificar se já existe um cromossoma igual a este filho na população.

Em situações mais extremas, dever-se-ão armazenar todos os cromossomas da geração actual e da anterior, verificando se algum deles é igual ao novo filho gerado.

É claro que tais abordagens também incorporam um custo computacional extra ao AG. Deve-se analisar se este custo extra compensa o tempo economizado na avaliação da função objectivo.

Uma outra abordagem é procurar uma maneira de simplificar a função objectivo. A versão simplificada da função objectivo seria utilizada nas gerações iniciais para acelerar a busca por regiões promissoras no espaço de busca. Nas gerações finais utiliza-se, a versão original da função objectivo para melhorar a precisão da solução.

5.7.3. CRITÉRIO DE PARAGEM

Alguns dos possíveis critérios de paragem para os AGs, são:

- Quando o AG atingir um dado número de gerações (ou avaliações);
- Attingir o valor óptimo da função objectivo, se este for conhecido;
- Convergência, isto é, quando não ocorrer melhoramento significativo no cromossoma de maior aptidão por um dado número de gerações.

Outras alternativas são também usadas, por exemplo, considerando que um gene converge se 90% da população tem o mesmo valor para este gene. Se, entre 90% e 95% dos genes convergirem, o AG convergiu [99].

5.7.4. SUBSTITUIÇÃO GERACIONAL E *STEADY-STATE*

Existem dois tipos básicos de substituição de cromossomas: substituição geracional e substituição *steady-state*.

Na substituição geracional toda a população é substituída em cada geração, ou seja, são criados N filhos para substituir N pais. Em alternativa, podem ser criados N filhos e os N melhores indivíduos da união de pais e filhos substituem a população actual. Na substituição geracional com elitismo, os k melhores pais nunca são substituídos por filhos piores. Geralmente é utilizado um valor de $k=1$. O aumento do valor de k , aumenta o risco da convergência prematura.

Na substituição *steady-state*, são criados apenas dois filhos em cada geração, que substituem os dois piores cromossomas da população. Em alternativa [102], os dois filhos podem substituir os pais ou os dois cromossomas mais velhos da população, tendo por base a suposição de que o cromossoma existente na população há muitas gerações, já transmitiu os seus genes à população. Na forma geral da substituição *steady-state*, são criados N filhos que substituem os N piores pais.

5.7.5. PROBLEMAS DE CONVERGÊNCIA

A convergência prematura é um problema conhecido dos AGs. Ocorre quando surgem cromossomas de alta aptidão (mas não com aptidão óptima), e os cromossomas realmente óptimos ainda não estão presentes na população. Tais cromossomas (apelidados de super-indivíduos) geram um número excessivo de filhos que dominam a população, uma vez que a mesma é finita.

Estes cromossomas espalham os seus genes por toda a população, enquanto outros genes desaparecem (tal desaparecimento de genes é denominado de *genetic drift*). Como consequência, o algoritmo converge para um máximo ou mínimo local.

Combate-se a convergência prematura, limitando o número de filhos por cromossoma. Esta limitação pode ser realizada através do escalonamento da aptidão, ordenamento e outras técnicas existentes nos AGs, nomeadamente a utilização dos tipos de cruzamento e mutação mais destrutivos ou o aumento das taxas de cruzamento e/ou mutação.

Manter a diversidade dos cromossomas na população também combate a convergência prematura, visto que é também causada pela perda de diversidade. O aumento da taxa de mutação também melhora a diversidade (mais genes são criados). Evitar a inserção de filhos duplicados na população é uma outra alternativa para melhorar a diversidade.

5.7.6. MAPEAMENTO DA FUNÇÃO OBJECTIVO

O valor da função objectivo nem sempre é adequado para ser utilizado como valor de aptidão. Por exemplo, a função objectivo pode assumir valores negativos (a roda da roleta não funciona), valores muito próximos (torna a selecção aleatória), alguns valores

podem ser muito elevados em relação ao resto da população (causa a convergência prematura), etc.

O mapeamento da função objectivo para o valor da aptidão pode ser feito de várias maneiras, sendo algumas delas discutidas nos pontos seguintes.

5.7.6.1. ORDENAMENTO LINEAR

No método de ordenamento linear [93], [110], a aptidão é dada por:

$$f_i = \text{Min} + (\text{Max} - \text{Min}) \frac{N - i}{N - 1} \quad i \text{ é o índice do cromossoma na população}$$

Normalmente é utilizado $1 < \text{Max} < 2$ e $\text{Max} + \text{Min} = 2$. De notar que deste modo a aptidão representa o número de filhos esperados do cromossoma e $\text{Max} - \text{Min}$ representa a pressão de selecção.

A Figura 43 mostra o controlo da pressão de selecção utilizando o ordenamento linear. À esquerda, a alta pressão de selecção favorece fortemente os melhores cromossomas, direccionando a busca para as melhores soluções encontradas até então. À direita, a baixa pressão de selecção favorece um pouco mais os cromossomas de baixa aptidão, direccionando a busca para regiões desconhecidas do espaço de busca.

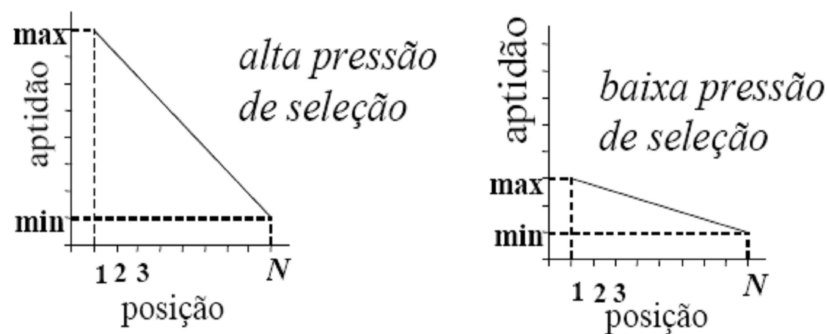


Figura 43 Posição do cromossoma vs. pressão de selecção

5.7.6.2. ORDENAMENTO EXPONENCIAL

No método de ordenamento exponencial [104] a aptidão é dada por:

$$f_i = q(1 - q)^{i-1} \quad q \in [0,1] \text{ e } i \text{ é o índice do cromossoma na população}$$

Em alternativa, a aptidão pode ser normalizada dividindo a equação anterior pelo factor $1-(1-q)^N$. O ordenamento exponencial permite uma maior pressão de selecção do que o ordenamento linear.

5.7.6.3. ESCALONAMENTO LINEAR

No método de escalonamento linear [79] a aptidão é obtida pela equação:

$$f = ag + b \quad \text{em que } g \text{ é o valor da função objectivo}$$

Os coeficientes a e b são determinados, limitando o número esperado de filhos dos cromossomas (filhos em excesso causam perda de diversidade). O escalonamento linear transforma as aptidões de tal modo que a aptidão se torna igual ao valor médio da função objectivo e a aptidão máxima é igual a C vezes a aptidão media (Figura 44). Tipicamente, o valor de C está entre 1,2 e 2,0.

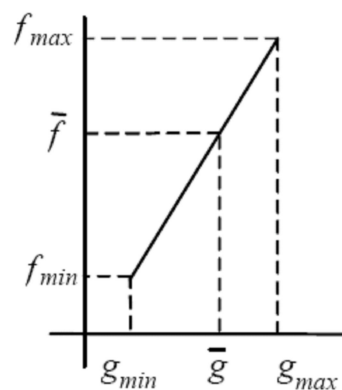


Figura 44 Gráfico de escalonamento da aptidão

5.7.7. OPTIMIZAÇÃO MULTI-OBJECTIVO

Muitos problemas do mundo real envolvem a optimização de mais do que uma função objectivo. Uma abordagem simples para tratar um conjunto de funções objectivo $f_1, f_2, f_3, \dots, f_n$, é formar uma nova função objectivo F , obtida através da soma ponderada das funções objectivo, ou seja:

$$F = \sum_{i=1}^n w_i f_i \quad \text{em que } w_i \text{ é um peso (que pode ser usado para dar mais relevância a uma função)}$$

A minimização de F minimiza todas as funções f_i . No entanto, as funções objectivo podem ser conflituosas, no sentido em que optimizar uma função pode degradar a optimização

de uma outra (e.g. minimizar os custos de um produto e maximizar a sua qualidade apresenta, em geral, objectivos conflituosos).

Uma melhor abordagem passa por usar o conceito de Ótimo de Pareto (devido a Vilfredo Pareto, um dos primeiros a usar tal conceito). Este conceito introduz a ideia de dominação e não-domação [111], [112]. Uma solução x_1 é dita não dominada por uma solução x_2 , quando ela, em todas as funções, é no mínimo tão boa quanto a solução x_2 , e em pelo menos numa das funções, x_1 é melhor que x_2 , ou seja:

Condição 1: $\forall i : f_i(x_1) \leq f_i(x_2)$

Condição 2: $\exists i : f_i(x_1) < f_i(x_2)$

A meta do AG é encontrar o conjunto Ótimo de Pareto (também denominado de Fronteira Ótima de Pareto), que é o conjunto de todas as soluções não-dominadas do espaço de busca (Figura 45).

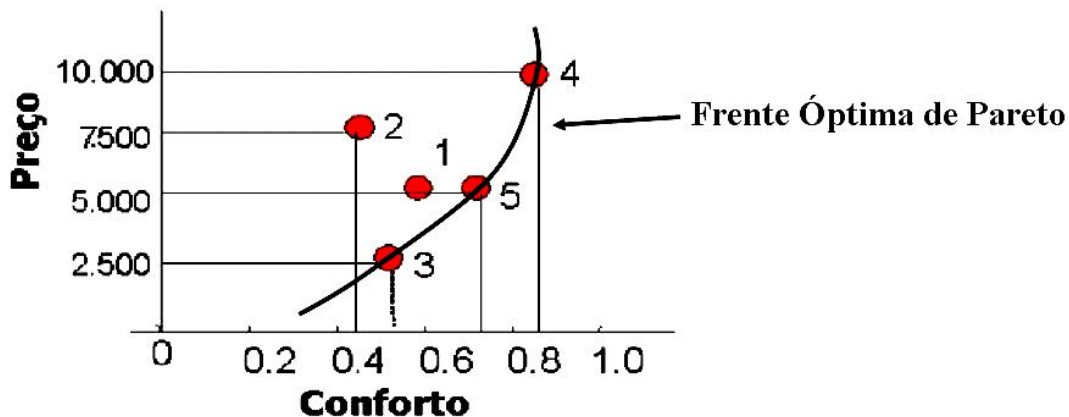


Figura 45 Frente Ótima de Pareto

Vale a pena notar que o AG retornará ao utilizador um conjunto de soluções ótimas ao invés de uma única solução (Figura 46). A aptidão pode ser calculada utilizando ordenamento com base na não-domação de indivíduos, conforme sugerido pelo seguinte procedimento: atribui-se a posição 1 a todos os indivíduos não dominados da população. Em seguida, tais indivíduos de posição 1 são removidos, temporariamente, da população. Na que restou, atribui-se a posição 2 aos indivíduos não dominados. Em seguida, retira-se temporariamente os indivíduos de posição 2 da população. Na restante,

repete-se o procedimento, atribuindo a posição 3 aos indivíduos não dominados, e assim por diante até a população se tornar vazia (Figura 47).

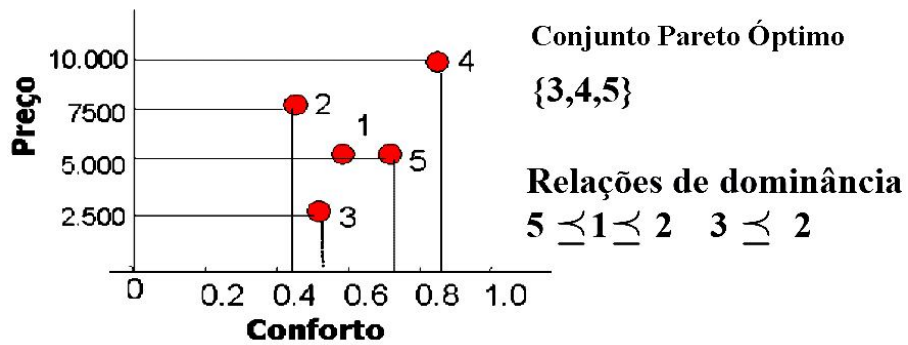


Figura 46 Conjunto Ótimo de Pareto

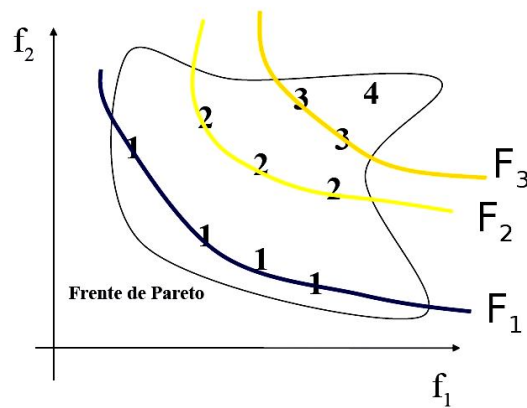


Figura 47 Ordenamento com base na não-dominância dos indivíduos

A otimização com o Ótimo de Pareto tem sido usada com nichos (Figura 48), visando estabilizar o grande número de subpopulações que surgem no conjunto Ótimo de Pareto [113].

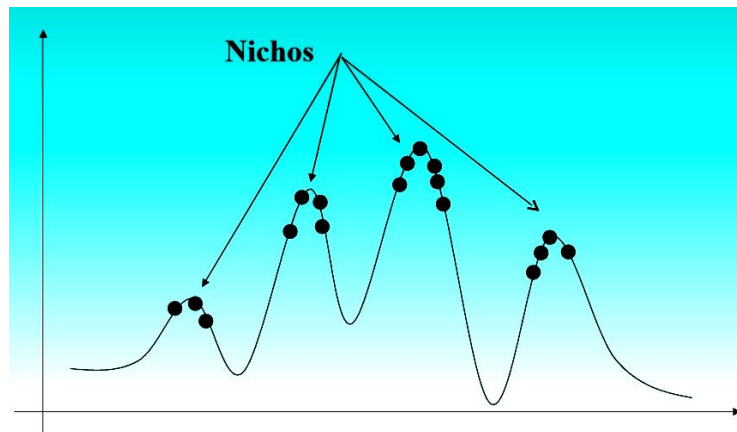


Figura 48 Método Ótimo de Pareto com recurso a nichos de população

5.7.8. RESTRIÇÕES

Vários problemas de optimização do mundo real contêm restrições. Em geral, um problema de optimização restrita pode ser declarado como:

Minimize $f(x)$

Sujeito a $g_1(x) \leq 0$

$g_2(x) \geq 2$

...

$g_m(x) \leq 1$

Uma abordagem para lidar com restrições em AG é simplesmente atribuir aptidão zero aos cromossomas irrealizáveis (i.e., aqueles que não satisfazem as restrições). No entanto, os cromossomas irrealizáveis próximos das regiões factíveis podem conter informações importantes para gerar filhos executáveis. Portanto, ao invés de atribuir o valor zero à aptidão desses cromossomas, pode-se apenas penalizar a aptidão. Para isto, usa-se uma função de penalização, que define o quanto a solução X viola a restrição J (i.e., o quão distante o cromossoma irrealizável está da região executável).

5.8. CONCLUSÃO

Os AGs são apropriados para problemas de optimização complexos, que envolvem muitas variáveis e um espaço de soluções de dimensão elevada. Abrangem um grande número de aplicações e o controlo sobre os parâmetros do algoritmo é de fundamental importância para uma convergência rápida. Para problemas específicos é aconselhável a utilização de algoritmos híbridos, que misturam as técnicas dos AGs com os métodos de optimização tradicionais.

Devido ao grande número de variáveis que um AG trata e às populações elevadas e alto número de gerações para a cobertura do espaço de soluções, os AGs possuem um custo computacional elevado.

Há dois aspectos importantes da programação de AGs que diferem um pouco da programação habitual. O primeiro é que é muito mais difícil detectar erros na programação, devido ao elevado número de linhas de código que se utilizarão,

unicamente como melhorias, ao algoritmo base. O segundo é que em muitos casos o programa funcionará e ainda fornecerá bons resultados, executando funções incorrectas.

No próximo capítulo iremos abordar a aplicação dos algoritmos genéticos à robótica, nomeadamente, no controlo do robô, na determinação das características físicas do robô, no cálculo de trajectórias, etc.

6. ALGORITMOS GENÉTICOS APLICADOS À ROBÓTICA

A utilização de algoritmos inspirados na evolução das espécies, conhecidos como Algoritmos Evolutivos, tem sido proposta na Robótica, tanto para a optimização da morfologia, como para o desenvolvimento de leis de controlo e de estratégias de navegação [114]. Os Algoritmos Evolutivos têm sido utilizados nesta área principalmente porque o projecto de robôs autónomos e dos seus controladores para ambientes não-estruturados, flexíveis, e/ou parcialmente desconhecidos é uma tarefa muito difícil para um projectista humano [115].

A dificuldade do projecto advém, principalmente, da impossibilidade de prever todas as situações com as quais os robôs serão confrontados. Utilizando Algoritmos Evolutivos, o ambiente e a tarefa a ser executada passam a ser os factores principais no desenvolvimento do robô e do seu controlador, tirando o trabalho que antes cabia à, muitas vezes, restrigente mente do projectista. Tal é a perspectiva do uso de Algoritmos Evolutivos em Robótica, que um termo foi proposto especialmente para designar os mecanismos robóticos por eles criados: Robôs Evolutivos [114].

Salienta-se que a conexão entre Robótica e Biologia não têm um sentido único neste caso: os robôs autônomos podem servir como uma importante ferramenta para o desenvolvimento e teste de modelos de comportamentos adaptativos, de habilidades cognitivas e de modelos evolutivos dos sistemas biológicos [116], como é o caso do robô Whegs (Figura 49).

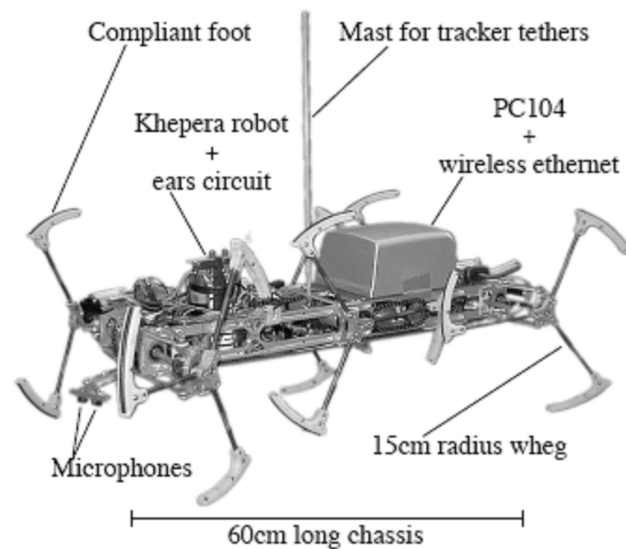


Figura 49 Robô Whegs

As pesquisas em Robôs Evolutivos têm-se concentrado quase exclusivamente em problemas de otimização estacionários. No entanto, a aplicação da robótica em ambientes flexíveis, não-estruturados, e/ou parcialmente desconhecidos exige que os robôs se consigam adaptar às mudanças no ambiente, fazendo com que os problemas de otimização se tornem problemas não-estacionários. Tais mudanças representam variações nas restrições das soluções ou na função de aptidão (*fitness*) associada ao problema que ocorre durante o processo de otimização.

São diversos os exemplos de mudanças que podem ocorrer em tais ambientes, como sejam: ocorrência de falhas [117], [118], mudanças nas características do ambiente [119], cooperação e competição entre robôs [114], mudança de tipo de robô [120] e mudanças de estratégias.

Algumas soluções foram propostas para tratar dos problemas não-estacionários especificamente em Robôs Evolutivos. Entre estes, podem ser citadas as seguintes

técnicas: adição de ruído [121], avaliação da função de aptidão para diferentes condições do ambiente [122], combinação de Algoritmos Evolutivos e métodos de aprendizagem [114] e manutenção do nível de diversidade por meio de Imigrantes Aleatórios [123]. As duas primeiras técnicas implicam o conhecimento, *à priori*, das características do ambiente, o que muitas vezes é inviável. Já quando se utilizam técnicas de aprendizagem associadas com Algoritmos Evolutivos, os custos evolutivos associados à aprendizagem dão uma vantagem selectiva para a assimilação das propriedades aprendidas para os indivíduos que estão a ser evoluídos [120], [124].

O AG com Imigrantes Aleatórios é inspirado no fluxo de indivíduos que entram e saem de uma população entre duas gerações na Natureza [125]. Este método é bastante simples e interessante. Em cada geração do processo de optimização, alguns indivíduos da população corrente são substituídos por indivíduos criados aleatoriamente. As estratégias de substituição, passam, por exemplo, por substituir indivíduos escolhidos aleatoriamente ou indivíduos com os menores valores de aptidão.

Através da introdução de novos indivíduos, o método dos Imigrantes Aleatórios tenta manter o nível de diversidade da população. A manutenção da diversidade da população pode ser bastante útil em problemas sujeitos a mudanças [126] para evitar que a população fique presa em óptimos locais (ou seja, soluções que são óptimas nas suas vizinhanças locais) representados pelas soluções adoptadas antes da mudança no problema.

Contudo, em alguns casos, como quando o número de genes de um indivíduo é alto e/ou quando os óptimos locais em que a população se encontra têm valores de aptidão muito maiores que os valores médios de todos os indivíduos possíveis do espaço de busca, a probabilidade que novos indivíduos sejam preservados é geralmente muito pequena. Isso ocorre porque o AG preserva, directa ou indirectamente, os melhores indivíduos da população corrente, e a probabilidade de que novos indivíduos tenham valores de aptidão próximos ou maiores que os valores dos indivíduos actuais é geralmente baixa.

6.1. CONTROLO DA LOCOMOÇÃO

O controlo da locomoção em robôs com pernas é um problema de busca num espaço de estados multi-dimensional que vem desafiando os investigadores ao longo das últimas décadas [127]. Este controlo requer a especificação e a coordenação dos movimentos de todas as pernas do robô, considerando factores como a estabilidade e o atrito em relação à superfície de contacto (solo) [128].

Esta é uma área de investigação que tem uma clara ligação com o controlo da locomoção realizada pelos animais, e muitas das pesquisas realizadas até o momento inspiram-se na locomoção efectuada por animais, como os mamíferos e os insectos [129].

O controlo da locomoção em robôs com pernas é uma área de investigação que vem sendo explorada à bastante tempo. Como trabalhos pioneiros nesta área podem-se destacar os primeiros robôs com pernas realmente independentes, como o *Phony Pony* (Figura 50) desenvolvido por Frank e McGhee [130], onde cada junta era controlada por uma simples máquina de estados finita, até ao muito bem sucedido controlo algorítmico de bípedes e quadrúpedes desenvolvido por Raibert [131].

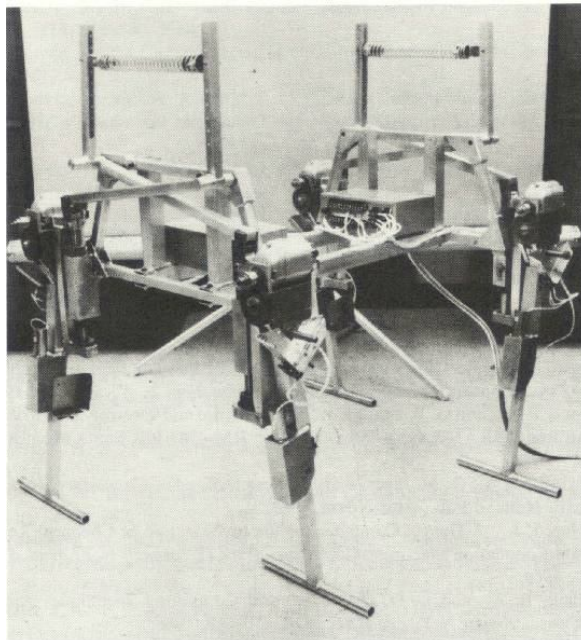


Figura 50 Robô Phony Pony

Na área do controlo inteligente de robôs dotados de pernas, os primeiros trabalhos datam do final dos anos 80 e início dos anos 90 do século XX. A título de exemplo refira-se o trabalho de Lewis *et al.* [132], que utilizaram os AGs para a evolução dos controladores de um robô de seis pernas. Neste trabalho, o controlador foi evoluído num robô cujo caminhar era inspirado no caminhar dos insectos. Através de vários estágios de evolução, o seu comportamento foi sendo modificado até atingir um caminhar razoavelmente satisfatório. Bongard e Pfeifer [133] evoluíram os parâmetros de uma Rede Neuronal Artificial dinâmica utilizada para controlar diversos tipos de robôs simulados.

Reeve e Hallam [129] utilizaram AGs para a evolução dos parâmetros de diversos modelos de Redes Neurais utilizadas para o controlo de diversos quadrúpedes simulados utilizando o DynaMechs [134]. Na maioria das abordagens descritas acima, a função de aptidão utilizada foi a distância percorrida pelo robô durante um certo período de tempo. Embora esta função de aptidão seja largamente utilizada, ela pode fazer com que a evolução privilegie formas de locomoção pouco estáveis em detrimento de soluções um pouco mais lentas porém muito mais estáveis [135]. Em [136], [137], para além da distância percorrida pelo robô, foram utilizadas como critério de aptidão informações sensoriais, provenientes de um giroscópio simulado a fim de se garantir que os movimentos obtidos fossem não só rápidos mas também estáveis.

Num robô dotado de pernas, a locomoção pode ser gerada de diversas formas. Uma das alternativas possíveis consiste na utilização de uma máquina de estados, onde cada estado determina o ângulo desejado para cada uma das juntas do robô. Nesta abordagem, o controlador realiza continuamente a leitura dos ângulos de cada uma das juntas, para verificar se elas já atingiram os valores desejados.

Em robôs reais, os ângulos das juntas podem ser obtidos através da leitura de sensores (*encoders*) instalados nas mesmas [138]. Desta forma, o controlo do caminhar é realizado da seguinte forma: inicialmente o controlador verifica se as juntas já atingiram os ângulos desejados. As juntas que não os tiverem atingido são movimentadas (os motores são activados) e, quando todas elas tiverem atingido os seus respectivos ângulos, o autómato passa para o estado seguinte. Se alguma das juntas não tiver atingido o ângulo desejado após um limite de tempo, o estado é avançado independente desta.

Para que haja sincronia nos movimentos, é importante que todas as juntas atinjam os ângulos desejados praticamente ao mesmo tempo, o que é possível com a aplicação de uma velocidade angular específica para cada uma das juntas. Para limitar o espaço de busca, o Algoritmo Genético gera apenas valores dentro do intervalo máximo e mínimo de cada junta.

Outra forma de se controlar a locomoção, consiste na utilização de um autómato no qual é determinado, para cada estado, a posição em que as extremidades (pés) do robô se devem encontrar em relação aos eixos x , y e z [139]. De acordo com estes autores, esta abordagem é útil principalmente em robôs bípedes, onde o número de graus de liberdade (número de juntas) normalmente é elevado (quatro ou mais). Já em robôs com menos de quatro graus de liberdade por perna, esta abordagem não é tão eficiente, pois não haverá uma redução do espaço de estados (número de parâmetros).

6.2. PROJECTO DE ROBÔS

Os robôs com pernas apresentam vantagens significativas em relação a mecanismos com rodas ou lagartas, devido à sua capacidade de se moverem em terrenos acidentados e não estruturados e contornar obstáculos. Contudo, sem estratégias de locomoção eficientes essas vantagens não podem ser realizadas. Os robôs que tropeçam e chocam com obstáculos são exemplos clássicos de robôs com estratégias de locomoção ineficientes; em cada contacto com o terreno há uma perda de energia e um desgaste mecânico do veículo, resultando numa redução do tempo de vida esperado.

O projecto de um robô com pernas, capaz de percorrer vários terrenos eficientemente e autonomamente é uma tarefa desafiante e implica o conhecimento de uma larga variedade de disciplinas. No projecto de robôs com pernas é muitas vezes útil considerar sistemas biológicos de locomoção, que tendem a ser muito mais versáteis e parecem ser mais eficazes e elegantes. De facto, os investigadores que trabalham na área da robótica olham na direcção do mundo natural em busca de inspiração e soluções.

As estratégias evolutivas são uma forma alternativa de imitar a natureza. Assim, não se copiam directamente as características dos animais, mas copia-se o processo que a natureza usa para a sua geração e na sua evolução.

Uma possibilidade passa por recorrer a algoritmos genéticos, como ferramenta para a criação da estrutura de robôs [140],[141]. Nestas aplicações efectua-se uma aproximação modular do projecto. Existe uma biblioteca de componentes elementares e vários desses componentes como juntas, elos, engrenagens, fontes de alimentação, etc. são combinados entre si de forma a originarem diferentes estruturas. As estruturas obtidas são avaliadas, recorrendo a funções de aptidão, e são recombinadas entre si recorrendo aos AGs. Este processo dá origem a um sistema robótico que representa o melhor projecto para uma dada aplicação. Estas aplicações informáticas apresentam a possibilidade de uma fácil reconfiguração, sendo passíveis de serem aplicadas na geração de sistemas robóticos para situações muito distintas.

Na literatura encontram-se também descritos trabalhos em que se recorre a estratégias evolutivas tendo em vista a optimização da estrutura de um robô em particular. Como é o caso de Juárez-Guerrero, et al. [142] que desenvolveram um robô bípede recorrendo às estratégias evolutivas. O objectivo final era fazer evoluir a estrutura de um robô bípede, equipado com uma cauda passiva para auxiliar a manter o equilíbrio, que conseguisse implementar um padrão de locomoção simples. O robô obtido foi construído e verificou-se a sua adequação à tarefa proposta. Para além do mecanismo de locomoção, este processo permitiu ainda optimizar a distância do centro de massa do robô à sua cauda, o comprimento da cauda e a área dos pés.

O recurso a AGs para optimizar a estrutura de um robô bípede é também adoptado por Ishiguro, et al. [143]. Pretende-se que o robô seja capaz de se deslocar de forma passiva num plano inclinado e através de juntas actuadas em superfícies planas. Numa primeira fase, são optimizados os parâmetros do corpo do robô recorrendo a um AG. Depois da estrutura do robô estar optimizada, estes autores recorrem a um outro algoritmo genético para optimizar os parâmetros do controlador.

Contrariamente a estes exemplos, em que a estrutura e o sistema de controlo dos robôs são optimizados separadamente, Endo, et al. [144], [145], [146] recorrem a um AG para optimizar simultaneamente a estrutura e o sistema de controlo do robô bípede humanóide PINO (Figura 51). Para iniciarem o processo de optimização (evolução da estrutura do robô) recorrem a um modelo baseado numa estrutura multi-elos. O

resultado da evolução da estrutura do robô é, numa primeira fase, o comprimento óptimo dos elos e, numa segunda fase, o posicionamento e orientação óptima dos servomotores.

O AG adoptado por estes autores é multi-objectivo e implementado em duas fases. Numa primeira fase, a função objectivo é baseada na distância percorrida pelo robô. Os vinte melhores robôs encontradas nesta fase, são a população inicial da segunda fase, em que o AG tem como funções objectivo a eficiência energética e a estabilidade da locomoção do robô. De acordo com estes autores, esta estratégia apresenta vantagens sobre a estratégia tradicional, na qual o sistema de controlo para um dado robô é desenvolvido recorrendo a um algoritmo genético, tendo por base uma estrutura previamente definida.



Figura 51 Robô PINO

6.3. PROJECTO DE CONTROLADORES

O estudo sobre robôs caminchantes identificou certas áreas chave no processo de evolução, o que resultou em mecanismos naturais que são muito versáteis. Por exemplo, o caranguejo que é conhecido pela sua agilidade [147]. Este êxito é basicamente devido à redução, pela evolução natural, do abdómen que fez a locomoção lateral exequível [148] e assim abriu o potencial do movimento extremamente rápido porque os membros adjacentes podem mover-se agora livremente sem existirem colisões entre eles.

Um exemplo é o Robug III (Figura 52), para o qual muitas das características de projecto foram inspiradas por soluções naturais. O Robug III é um robô tele-operado, compacto e poderoso com membros articulados [149]. O Robug III adoptou a locomoção tipo “caranguejo” por causa da velocidade de deslocação ser mais rápida e pela exigência do robô de ser capaz de passar por uma passagem estreita. Contudo o robô é também capaz de usar uma marcha longitudinal (típica dos insectos).



Figura 52 Robô Robug III

A coordenação do movimento dos pés do Robug III envolve uma decisão a respeito de que pé deve ser levantado ou colocado no solo. Os meios pelos quais a decisão é feita são conhecidos como estratégia de marcha [150]. No caso extremo, esta decisão deve ser efectuada tendo em consideração factores como a condição do terreno, as exigências de estabilidade, facilidade de controlo, suavidade do movimento do corpo, exigências de velocidade, exigências de mobilidade e exigências de esforço.

Um algoritmo genético pode ser aplicado no contexto dos modelos de marcha. Pode-se verificar que os padrões de locomoção com margens de estabilidade óptimas, ou perto de óptimas, podem ser obtidos usando um AG para facilitar a obtenção dos parâmetros óptimos do padrão de locomoção e que os padrões resultantes são comparáveis com os encontrados na natureza.

O padrão de locomoção é uma questão significativa do controlo de robôs com pernas uma vez que o movimento coordenado de todas as pernas é necessário para produzir uma boa locomoção. Desenvolver um sistema para aprender padrões de locomoção pode eliminar a necessidade de escrever controladores para cada robô específico, e possibilita ainda ao robô a capacidade de se adaptar a modificações nas suas capacidades e/ou ambiente.

O problema de aprender padrões de locomoção pode dividir-se em duas partes: aprendizagem de ciclo de perna e aprendizagem de ciclo de marcha. Deste modo, podem ser criados controladores para cada perna de forma a produzir ciclos eficazes de movimento. Esses controladores podem, então, ser coordenados para colaborar na produção de um padrão que mantém a estabilidade, movimentando o robô para a frente à velocidade máxima.

Um método para obter o controlador que pode produzir um ciclo de perna, de uma perna individual, pode ser feito usando um AG que aprende as conexões e os seus pesos de uma rede neuronal. Alguns investigadores usaram a programação evolutiva para desenvolver controladores de robôs com pernas [151]-[155].

Beer e Gallagher [151] usaram AGs para desenvolver controladores de rede neuronais para um robô hexápode. O controlador apresenta três neurónios motores (pé, balanço para trás e balanço para frente) e dois neurónios internos não especificados. Os três neurónios motores e os dois neurónios internos foram totalmente unidos e o AG aprendeu os pesos dessas conexões. Depois de ser obtido um ciclo de perna para uma perna genérica, foram feitas seis cópias do controlador de perna. Cada um dos controladores de perna foi unido aos seus controladores de perna vizinhos e foi usado um AG para aprender os pesos de conexão entre as pernas.

Lewis *et al.* [152] também usaram esta evolução organizada. No seu modelo, a posição de cada uma das duas juntas da perna é dirigida pelo estado de um neurónio. Os dois neurónios que controlam a perna são o neurónio de balanço de perna e o neurónio de elevação de perna. Para encontrar os parâmetros desta rede de dois neurónios foi usado um AG.

6.4. PLANEAMENTO DE TRAJECTÓRIAS

Os robôs móveis são cada vez mais comuns na sociedade de hoje. As suas aplicações são tão diversas como o, cientificamente inspirado, *NASA Mars Rover* até robôs aspiradores domésticos. Apesar da aplicação, o planeamento de trajectória é uma parte integrante do projecto de robôs móveis.

O objectivo do planeamento de trajectória é mover o robô móvel de uma posição inicial conhecida a uma posição final desejada evitando obstáculos, como seres humanos, objectos variados ou outros robôs móveis. Algumas aplicações do planeamento de trajectória incluem o uso de robôs móveis para inspecionar um derramamento de material industrial tóxico, posição e extensão do derrame químico. O Centro do Robô Busca Assistida e Resgate da Universidade da Flórida do Sul também está a estudar o uso de robôs móveis (Figura 53) e planeamento de trajectórias de forma a procurar sobreviventes em edifícios sucumbidos [156].

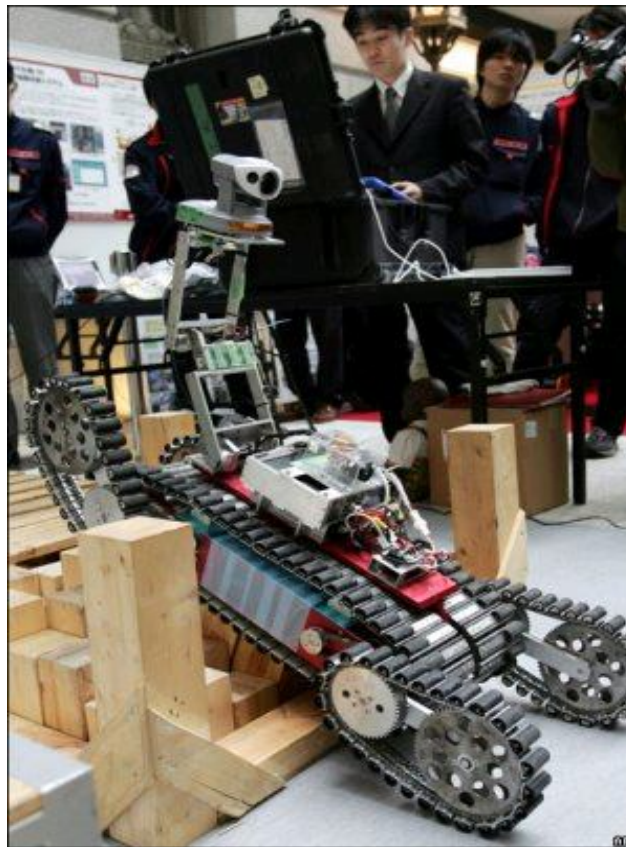


Figura 53 Robô de resgate criado na Universidade da Flórida do Sul

O planeamento de trajectórias é uma questão importante na robótica móvel. Num ambiente com obstáculos, o planeamento de trajectórias deve encontrar um caminho conveniente para que o robô móvel se mova de uma posição de partida para uma posição desejada sem colidir com um obstáculo.

O planeamento de trajectórias de robôs foi uma área de pesquisa activa, e muitos métodos foram desenvolvidos para atacar este problema [157]. Cada método tem as suas vantagens relativamente aos outros em certos aspectos. Geralmente, as dificuldades principais do problema de planeamento de caminho de robô são a complexidade computacional, a condição favorável local e a adaptabilidade. Os investigadores sempre estiveram à procura de modos alternativos e mais eficientes de resolver o problema.

Não há dúvidas de que o planeamento da trajectoria pode ser examinado como um problema de optimização (p.ex., a distância mais curta) tendo em conta certas restrições (p.ex., o ambiente dado e condição de não existirem colisões). Muitos investigadores aplicaram o AG no planeamento de trajectória nos robôs móveis [158]-[160]. Contudo, a maior parte daqueles métodos adoptam o AG clássico que usa cadeias de binário de comprimento fixo e dois operadores genéticos básicos, e poucas modificações foram feitas aos algoritmos.

Além disso, a maior parte dos métodos que utilizam o AG para o planeamento de trajectória de robôs lida com ambientes estruturados ou ambientes com obstáculos de formas simples. Algumas aproximações de AG do planeamento de caminho são capazes de enfrentar obstáculos não só rectangulares ou convexos [161], [162]. Quando se trata de obstáculos complexos, o problema de avaliação transforma-se num desafio, que necessita de uma detecção precisa de colisão entre caminhos e obstáculos e uma avaliação de qualidade eficaz.

Em [163], o AG tem de avaliar os caminhos exequíveis e os caminhos irrealizáveis separadamente, através de duas funções de aptidão em cima da detecção de colisão. O método de avaliação é capaz de descobrir exactamente a colisão entre um trajecto do robô e um obstáculo arbitrariamente formado.

6.5. CONCLUSÃO

Como vimos neste capítulo, os algoritmos genéticos tem uma larga aplicação na robótica, nomeadamente nos robôs autónomos, através da optimização dos controladores de partes do robô ou do controlador principal do robô, através do planeamento da melhor trajectória a seguir, cálculo do trajecto mais curto e evitando a colisão com obstáculos existentes no ambiente onde este se insere.

Os algoritmos genéticos também são utilizados noutros robôs, nomeadamente na optimização dos parâmetros do robô, tamanho das pernas, peso robô, comprimento do passo, etc. Este trabalho irá incidir sobre este aspecto.

Com este capítulo, encerramos a introdução teórica deste trabalho. No próximo capítulo iremos abordar de todos os desenvolvimentos efectuados para a realização deste trabalho.

7. DESENVOLVIMENTO

Neste capítulo serão expostos todos os procedimentos efectuados para a realização deste trabalho. Assim, começa-se por descrever, na secção seguinte, as principais características do sistema de locomoção que se pretende otimizar. De seguida apresenta-se um conjunto de índices dinâmicos, tendo em vista analisar e comparar o desempenho de sistemas robóticos de locomoção multipernas (a sua aptidão).

Para concluir, será descrito o funcionamento do algoritmo genético desenvolvido para obter os parâmetros do sistema de locomoção que minimizam a densidade de energia por unidade de distância percorrida e o erro da trajectória real das ancas relativamente à trajectória planeada.

7.1. MODELO DO SISTEMA DE LOCOMOÇÃO

O modelo do sistema robótico de locomoção aqui descrito tem como base o modelo desenvolvido pelo meu orientador, Prof. Dr. Manuel Santos Silva, no âmbito da sua Tese de Doutoramento [164]. Alguns conceitos que irão ser apresentados a seguir foram retirados dessa mesma tese, de forma a um melhor enquadramento e entendimento do conteúdo deste trabalho.

Esta ferramenta permite o estudo da configuração óptima do sistema e do tipo de movimentos que levam a uma melhor implementação mecânica. Adicionalmente, este modelo de simulação é também utilizado para estudar o controlo das juntas das pernas. O movimento pretendido para o robô é descrito por diversas variáveis de locomoção, e pode ser parametrizado através de diversos parâmetros identificados como relevantes para o efeito (ver Figura 54). Por sua vez, as juntas e as transmissões das pernas podem apresentar diversos fenómenos dinâmicos, nomeadamente saturação não linear, atrito viscoso, flexibilidade e folga.

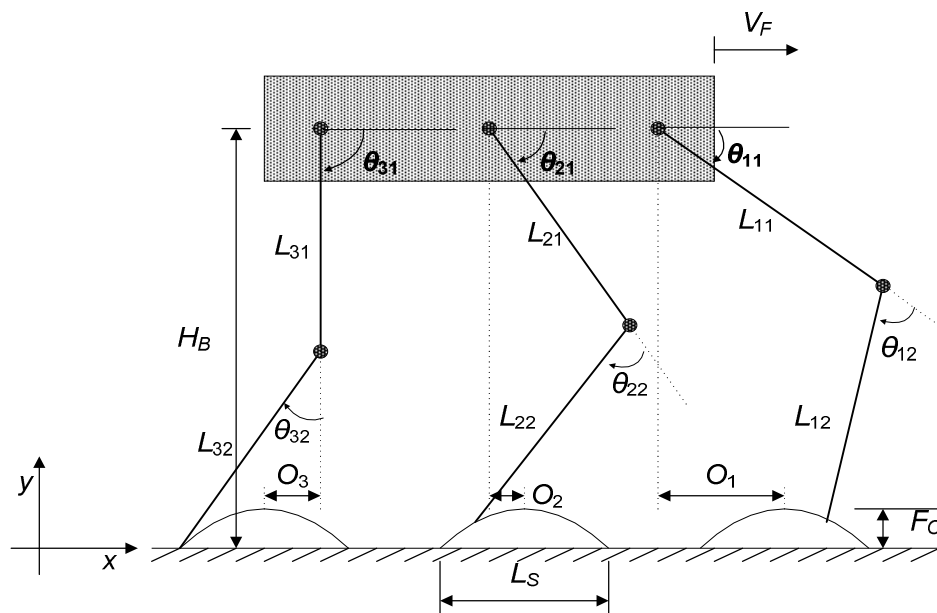


Figura 54 Modelo cinemático do robô utilizado [164]

Neste modelo, o corpo do robô é dividido em vários segmentos de forma a emular o comportamento da coluna vertebral de um animal. Quanto à interacção entre os pés do robô e o solo esta é modelada através de um sistema mola-amortecedor não linear em paralelo, cujos parâmetros são determinados tendo por base os estudos de mecânica dos solos.

7.1.1. MODELO CINEMÁTICO

7.1.1.1. PROPRIEDADES DAS PERNAS DO SISTEMA DE LOCOMOÇÃO

Neste trabalho considera-se um sistema de locomoção multipernas longitudinal, com seis pernas igualmente distribuídas de ambos os lados do corpo do robô. Quanto ao número

de gdl das pernas optou-se por cada perna apresentar dois gdl, correspondendo a duas juntas rotacionais, uma na anca e outra no joelho, como pode ser visto na Figura 54.

As pernas ímpares situam-se do lado esquerdo do corpo do robô e as pernas pares do lado direito (Figura 55).

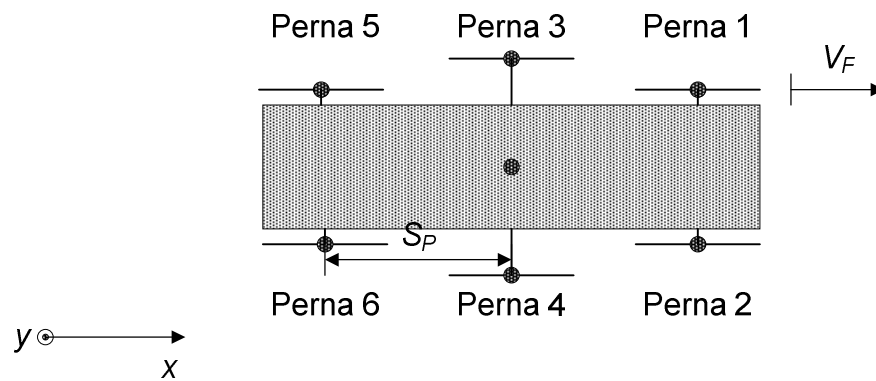


Figura 55 Numeração das pernas do sistema robótico de locomoção [164]

Nas experiências desenvolvidas supõe-se que as pernas se encontram mecanicamente afastadas entre si, no plano transversal ao da locomoção, para que nunca ocorram colisões entre elas, como é possível ver na Figura 55.

Todas as pernas são numeradas consecutivamente da parte da frente do robô para a sua parte de trás. Cada uma das pernas do modelo pode apresentar características diferenciadoras face às outras, implementadas através de diferentes valores para os parâmetros que as caracterizam (ver Figura 54), uma vez que o desempenho do sistema varia com as suas características.

7.1.1.2. PADRÕES DE LOCOMOÇÃO IMPLEMENTADOS PARA A LOCOMOÇÃO DO ROBÔ MULTIPERNAS

Os padrões de locomoção descrevem sequências descontínuas de movimentos das pernas, alternando entre fases de transferência e suporte. No modelo de simulação consideram-se os padrões de locomoção ondulatório (PLO), fase igual e meio ciclo (PLFIMC), fase igual e ciclo completo (PLFICC), ondulatório inverso (PLOI), fase igual e meio ciclo inverso (PLFIMCI) e fase igual e ciclo completo inverso (PLFICCI).

Um padrão de locomoção diz-se ondulatório se o início da fase de transferência progride sequencialmente ao longo do corpo, ou seja, a fase de transferência de uma perna inicia-se imediatamente após uma perna adjacente vizinha, mas antes das outras. Esta onda pode-se propagar de trás para a frente ou da frente para trás do corpo do robô, sendo que neste último caso se costuma designar por padrão de locomoção ondulatório inverso. Estes padrões de locomoção maximizam a margem de estabilidade.

Os padrões de locomoção de fase igual distribuem igualmente os acontecimentos de colocação e levantamento dos pés do veículo ao longo de meio, ou um, ciclo de locomoção. Desta forma, as flutuações no consumo energético são minimizadas.

No padrão de locomoção de fase igual e meio ciclo os acontecimentos de colocação dos pés do sistema de locomoção de um dos lados do robô estão igualmente distribuídos ao longo de meio ciclo de locomoção. Devido à simetria do padrão de locomoção, os acontecimentos de colocação dos pés do veículo do outro lado do robô estão distribuídos ao longo da outra metade do ciclo de locomoção.

Por sua vez, no padrão de locomoção de fase igual e ciclo completo os acontecimentos de colocação dos pés do veículo de um dos lados do robô estão igualmente distribuídos ao longo de um ciclo completo de locomoção.

Nos padrões de locomoção inversos, a sequência dos acontecimentos de colocação dos pés de cada lado do corpo do veículo inicia-se na perna da frente e avança em direcção à perna de trás, ao contrário do que ocorre nos padrões directos.

7.1.2. MODELO DINÂMICO

7.1.2.1. MODELO DINÂMICO DO CORPO DO ROBÔ

A Figura 56 apresenta o modelo dinâmico adoptado para o corpo do sistema artificial de locomoção e para a interacção pé-solo. Este modelo segmentado, considerando complacência entre os diversos segmentos adjacentes, foi adoptado para o corpo do robô, uma vez que todos os animais vertebrados que se deslocam com pernas recorrem a uma coluna vertebral. Este modelo é consistente com a biomecânica dos animais, uma

vez que é reconhecido que estes usam a complacência dos músculos e dos tendões para se deslocarem de uma forma mais eficiente.

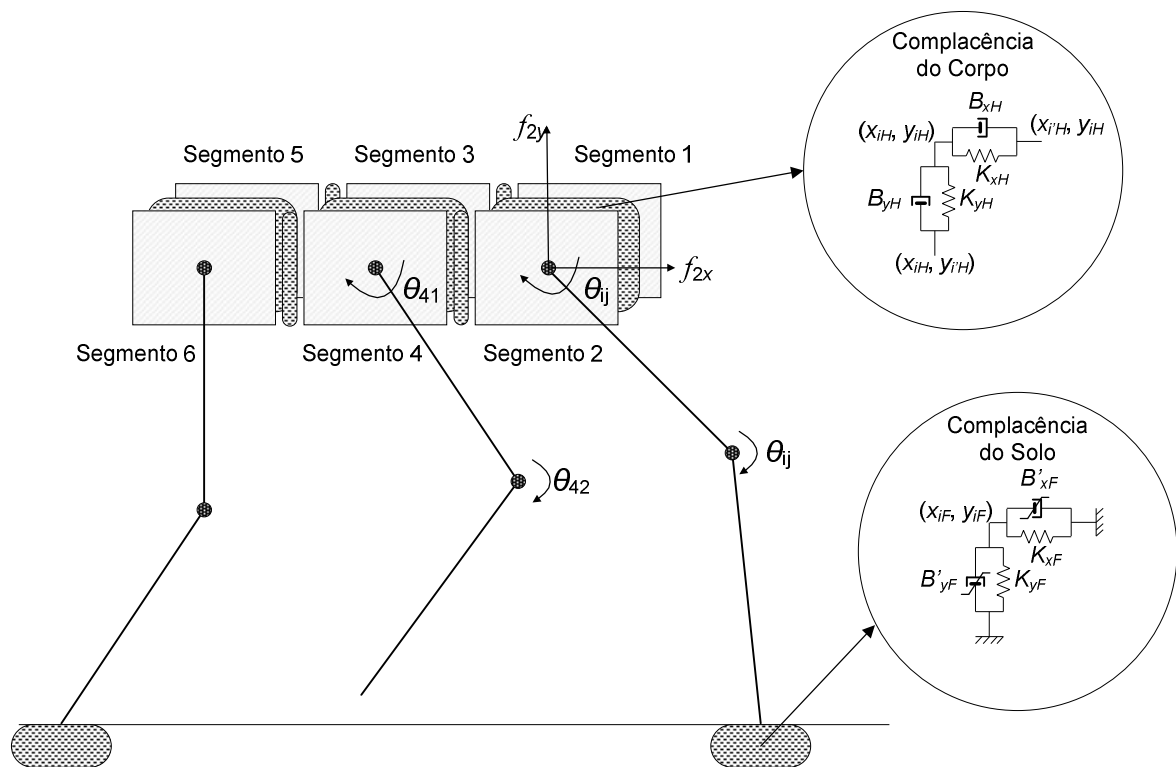


Figura 56 Modelo dinâmico do corpo do robô [164]

Este robô hexápode possui um corpo dividido em seis segmentos, cada um deles possuindo uma perna, encontrando-se os diferentes segmentos do corpo ligados entre si através de juntas lineares passivas, modeladas através de um sistema mola-amortecedor em paralelo.

7.1.2.2. MODELO DE INTERAÇÃO PÉ-SOLO

O contacto dos pés do robô com o solo (Figura 56) pode ser analisado através de diferentes pontos de vista, conduzindo a modelos distintos. Um método considera as relações exactas entre a força e a deformação do solo. Outro método, sob restrições particulares, passa pela utilização de modelos aproximados da deformação do solo baseados nos estudos de mecânica dos solos. Neste trabalho recorre-se à segunda aproximação. Adopta-se uma estratégia mista, ou seja, modeliza-se o contacto do enésimo pé do robô com o solo através de uma mola linear com coeficiente de

elasticidade K_η e um atrito não linear com coeficiente de amortecimento viscoso β_η nas direcções horizontal e vertical, respectivamente (Figura 56).

7.1.3. ARQUITECTURA DE CONTROLO

Apesar da diversidade de aproximações existentes para a geração e controlo do padrão de locomoção, o controlo ao nível da junta dos robôs com pernas é geralmente implementado através de um simples algoritmo do tipo PID aplicado ao controlo das trajectórias de referência das juntas e com realimentação de posição ou de posição e velocidade.

Para a mesma aplicação, têm sido também extensivamente utilizados os algoritmos de controlo PD. No controlo PD existem problemas de sintonia devido ao compromisso entre a sobre-elongação e o erro em regime permanente. Estes problemas são ainda acentuados devido ao facto de não existir um ponto de operação específico em torno do qual efectuar a sintonia do controlador, uma vez que o ponto de operação é determinado, entre outros, pelo padrão de locomoção, velocidade do robô e interacção entre o pé e o solo. Por outro lado, a implementação de controladores PD com ganhos não lineares variáveis tem uma implementação complexa nos microcontroladores geralmente utilizados para efectuar o controlo a baixo nível.

A primeira opção a considerar, em termos de controlo do sistema artificial de locomoção, prende-se com a arquitectura genérica de controlo e com o fecho da malha de realimentação.

Idealmente é necessário efectuar simultaneamente o controlo das trajectórias das ancas e dos pés do sistema artificial de locomoção. Assim, implementou-se a arquitectura de controlo apresentada na Figura 57. Como se pode verificar pela simulação apresentada na Figura 58, esta arquitectura de controlo permite que o robô se movimente de forma estável e com erros de pequena amplitude face às trajectórias planeadas.

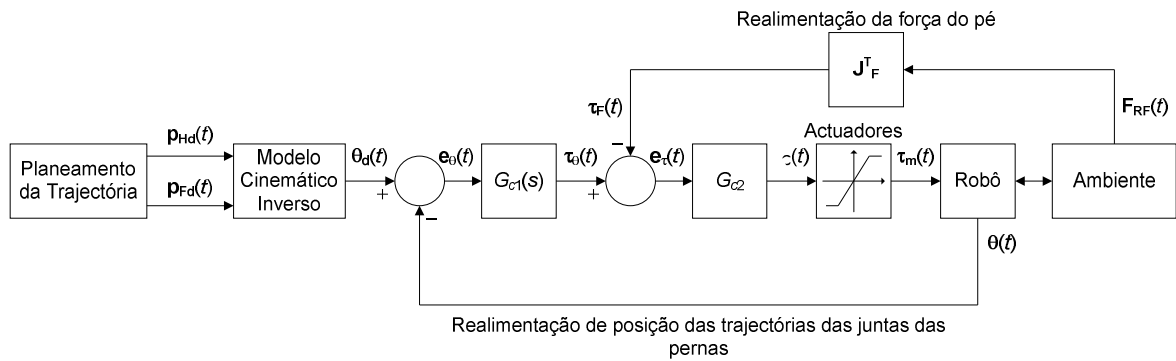


Figura 57 Arquitectura de controlo do sistema de locomoção [164]

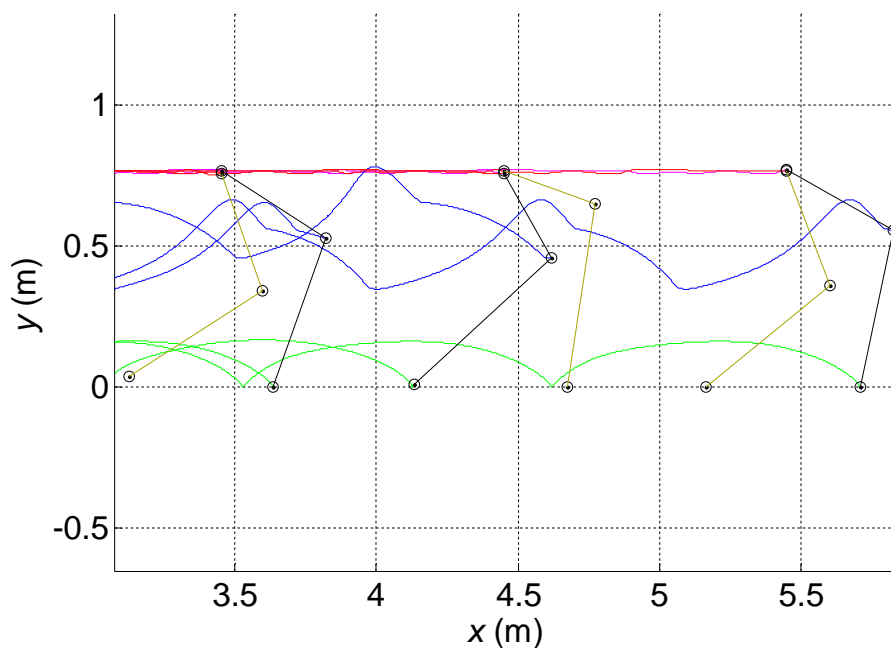


Figura 58 Simulação da locomoção do sistema robótico

7.1.4. MEDIDAS DE DESEMPENHO

Nesta subsecção apresentam-se os índices formulados e utilizados no decurso do trabalho para analisar a locomoção do robô do ponto de vista dinâmico, para o que se recorre ao seu modelo dinâmico. Assim, formulam-se duas medidas globais do desempenho do mecanismo estudado de um ponto de vista médio, nomeadamente o índice $\{E_{av}\}$ baseado na dinâmica do sistema robô-solo e o índice $\{\epsilon_{xyH}\}$ baseado nos erros de seguimento da trajectória da anca do robô.

7.1.4.1. DENSIDADE DE ENERGIA ABSOLUTA MÉDIA

A medida chave nesta análise é a densidade de energia absoluta média por distância percorrida E_{av} . Este índice é calculado considerando que não se verifica recuperação de energia pelos actuadores quando estes se encontram a desenvolver trabalho negativo, isto é, considerando que se toma o valor absoluto da potência mecânica.

Este raciocínio tem por base a constatação de que nos seres biológicos a energia que está envolvida no movimento dos membros, através da actuação muscular, não tem a possibilidade de ser recuperada durante os intervalos de tempo em que se efectua trabalho negativo.

Sabe-se que numa determinada junta j (cada perna tem m juntas) e perna i (o robô tem n pernas), a potência mecânica é o produto do binário desenvolvido pelo actuator pela velocidade angular. Como consequência, o índice global E_{av} é obtido efectuando a média da energia mecânica absoluta consumida ao longo da distância percorrida d :

$$E_{av} = \frac{1}{d} \sum_{i=1}^n \sum_{j=1}^m \int_0^T \left[\tau_{ij}(t) \dot{\theta}_{ij}(t) \right] dt \quad [Jm^{-1}]$$

sendo τ_{ij} o binário motor e $\dot{\theta}_{ij}$ a velocidade da junta.

Portanto, um bom desempenho do sistema requer a minimização do índice E_{av} .

7.1.4.2. ERROS QUADRÁTICOS MÉDIOS ENTRE AS TRAJECTÓRIAS PLANEADAS E REAIS

No que diz respeito aos erros de seguimento das trajectórias das ancas definiu-se o índice:

$$\varepsilon_{xyH} = \sum_{i=1}^n \sqrt{\frac{1}{N_s} \sum_{k=1}^{N_s} (\Delta_{ixH}^2 + \Delta_{iyH}^2)} \quad [m]$$

$$\Delta_{ixH} = x_{iHd}(k) - x_{iH}(k)$$

$$\Delta_{iyH} = y_{iHd}(k) - y_{iH}(k)$$

sendo N_s o número total de amostras, d indica as i -ésimas amostras da posição pretendida e $\{x,y\}$ representam as coordenadas da anca.

A optimização do desempenho requer a minimização deste índice.

7.1.5. APLICAÇÃO PARA SIMULAÇÃO DA LOCOMOÇÃO

O sistema de simulação inclui uma interface gráfica com o utilizador (IGU). Esta IGU, implementada em MATLAB, apresenta uma animação da locomoção do robô e as trajetórias do corpo, dos joelhos e dos pés do robô (Figura 59). No entanto, os algoritmos numéricos do modelo de simulação encontram-se implementados na linguagem de programação Microsoft C/C++. Os resultados da simulação são gravados em ficheiros (em formato texto) que são lidos pela aplicação de IGU de forma a gerar os relatórios gráficos.

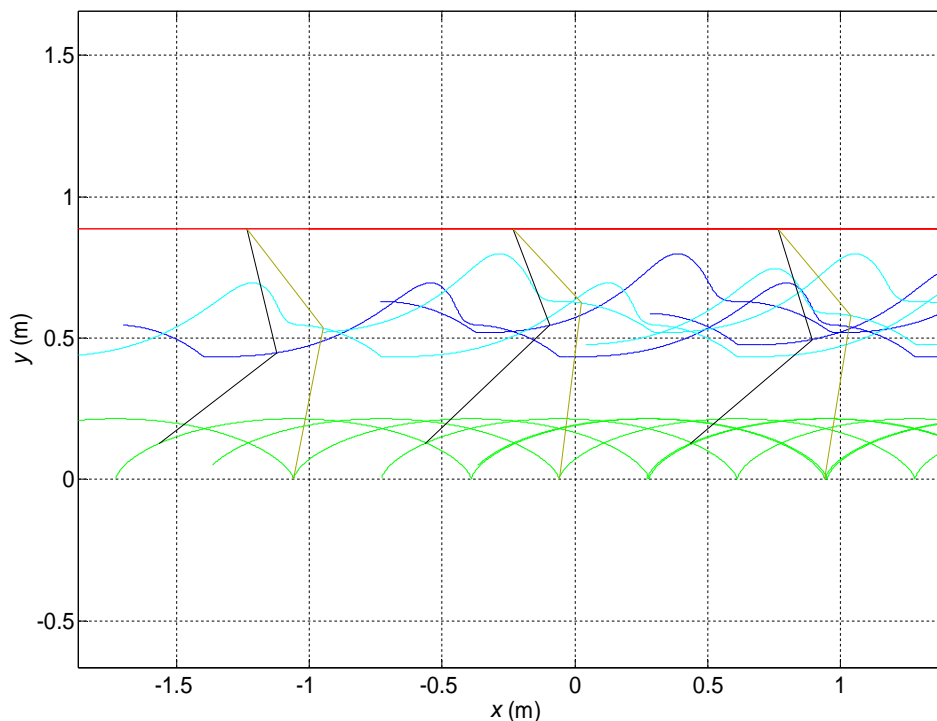


Figura 59 Interface gráfica de simulação do sistema robótico

7.2. ESTRUTURA DO CROMOSSOMA UTILIZADO

O modelo utilizado nas experiências realizadas é o de um robô hexápode, com as características apresentadas na Figura 60, onde se podem ver os parâmetros utilizados no Algoritmo Genético. Dos parâmetros do robô referidos na Figura 60, apenas um não faz parte do cromossoma inicialmente utilizado, sendo esse parâmetro a velocidade de deslocamento (V_F): este valor é fixo e igual a 1 m/s.

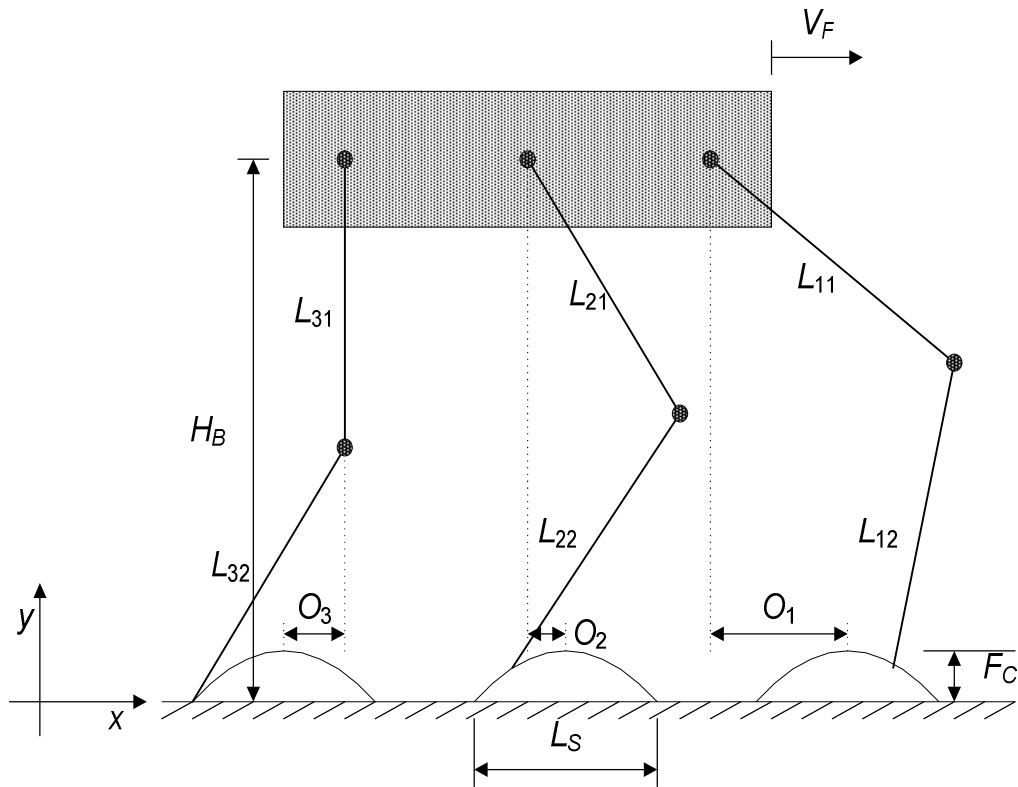


Figura 60 Parâmetros do robô utilizados no AG

Existem outros parâmetros do modelo do sistema de locomoção não apresentados na figura e que não são utilizados no cromossoma criado, ou seja, assumem um valor fixo. Como exemplos temos os parâmetros que definem as características do solo e o centro de massa das pernas do robô, que se assume que se situa exactamente a meio da perna.

Existe também um outro caso particular que é a complacência entre os diversos segmentos que compõem o corpo do robô (ver Figura 56). Assume-se que este valor é igual para as diversas ligações entre segmentos, integrando estes parâmetros (K_{xh} , B_{xh} , K_{yh} e B_{yh}) o cromossoma.

Existem ainda características físicas do robô que não são passíveis de alteração, nomeadamente o peso total do robô que é igual a 100 kg e o comprimento total das pernas que é de 1 m.

Outra particularidade prende-se com a massa das pernas do robô. No AG o cromossoma apresenta seis genes relacionados com esse parâmetro (M_{ij}); o parâmetro M_{12} , por exemplo, representa a soma das massas dos segundos elos das pernas da frente e não

apenas da perna 1 (ou seja, se M_{12} for igual a 4 kg, a massa do segundo elo da perna 1 do robô será de 2 kg e a massa do segundo elo da perna 2 do robô também será 2 kg).

O comprimento das pernas (L_{ij}) apresenta uma configuração diferente da que é referida no ponto anterior. Assim, no caso de L_{12} ser igual a 0,6 m, isto indica que o segundo elo das pernas da frente do robô (pernas 1 e 2) tem, em ambas, 0,6 m de comprimento.

Todos os outros parâmetros que apresentam índices (por exemplo, O_i , $T_{ijmin/max}$, Kp_{ij} e Kd_{ij}) têm uma leitura idêntica à descrita no ponto anterior (comprimento das pernas). Resumindo, o robô apresenta uma simetria em relação ao seu eixo transversal.

Por último, um outro pormenor prende-se com o ganho integral do controlador PID aplicado às pernas do robô. Uma vez que o termo Integral é usado para anular o erro em regime permanente, o valor do ganho integral (Ki_{ij}) é nulo pois o pé ao entrar em contacto com o solo provoca perturbações na trajectória do robô; logo, qualquer valor de Ki não iria surtir grande efeito. Deste modo o controlador usado é do tipo PD.

O cromossoma utilizado no algoritmo genético desenvolvido neste trabalho apresenta 48 genes (i.e., 48 parâmetros do robô). Os genes encontram-se escalonados da forma como estão apresentados na Figura 61: o primeiro gene (L_s) contém a informação do comprimento do passo e o último gene (Kd_{32}) contém o ganho derivativo da junta 2 das pernas traseiras do robô.

Ls	Comprimento do passo	T11min	Binário mínimo do motor da junta 1 das pernas da frente
Hb	Altura do corpo ao solo	T11max	Binário máximo do motor da junta 1 das pernas da frente
Beta	Factor de ocupação	T12min	Binário mínimo do motor da junta 2 das pernas da frente
Fc	Altura do pé ao solo	T12max	Binário máximo do motor da junta 2 das pernas da frente
L11	Comprimento do elo 1 das pernas da frente	T21min	Binário mínimo do motor da junta 1 das pernas do meio
L12	Comprimento do elo 2 das pernas da frente	T21max	Binário máximo do motor da junta 1 das pernas do meio
L21	Comprimento do elo 1 das pernas do meio	T22min	Binário mínimo do motor da junta 2 das pernas do meio
L22	Comprimento do elo 2 das pernas do meio	T22max	Binário máximo do motor da junta 2 das pernas do meio
L31	Comprimento do elo 1 das pernas de trás	T31min	Binário mínimo do motor da junta 1 das pernas de trás
L32	Comprimento do elo 2 das pernas de trás	T31max	Binário máximo do motor da junta 1 das pernas de trás
O1	Offset dos pés da frente em relação às ancas da frente	T32min	Binário mínimo do motor da junta 2 das pernas de trás
O2	Offset dos pés do meio em relação às ancas do meio	T32max	Binário máximo do motor da junta 2 das pernas de trás
O3	Offset dos pés de trás em relação às ancas de trás	Kp11	Ganho proporcional do controlador da junta 1 das pernas da frente
Mb	Massa do corpo do robô	Kd11	Ganho derivativo do controlador da junta 1 das pernas da frente
M11	Massa do elo 1 das pernas da frente	Kp12	Ganho proporcional do controlador da junta 2 das pernas da frente
M12	Massa do elo 2 das pernas da frente	Kd12	Ganho derivativo do controlador da junta 2 das pernas da frente
M21	Massa do elo 1 das pernas do meio	Kp21	Ganho proporcional do controlador da junta 1 das pernas do meio
M22	Massa do elo 2 das pernas do meio	Kd21	Ganho derivativo do controlador da junta 1 das pernas do meio
M31	Massa do elo 1 das pernas de trás	Kp22	Ganho proporcional do controlador da junta 2 das pernas do meio
M32	Massa do elo 2 das pernas de trás	Kd22	Ganho derivativo do controlador da junta 2 das pernas do meio
Kxh	Elasticidade entre segmentos, segundo o eixo X	Kp31	Ganho proporcional do controlador da junta 1 das pernas de trás
Kyh	Elasticidade entre segmentos, segundo o eixo Y	Kd31	Ganho derivativo do controlador da junta 1 das pernas de trás
Bxh	Atrito viscoso entre segmentos, segundo o eixo X	Kp32	Ganho proporcional do controlador da junta 2 das pernas de trás
Byh	Atrito viscoso entre segmentos, segundo o eixo Y	Kd32	Ganho derivativo do controlador da junta 2 das pernas de trás

Figura 61 Estrutura do cromossoma utilizado

7.3. ESTRUTURA BASE DO AG DESENVOLVIDO

A estrutura base do AG desenvolvido para a realização deste trabalho apresenta a configuração representada na Figura 62. Começa-se por gerar uma população inicial de cromossomas e, em seguida, é feita a avaliação da população (*i.e.*, verificar se o algoritmo converge). Caso o algoritmo não tenha convergido, são aplicados os operadores genéticos aos cromossomas da população e volta-se a avaliar a população. Este ciclo repete-se até o AG convergir ou, então, até o número de gerações atingir o limite máximo indicado pelo utilizador.

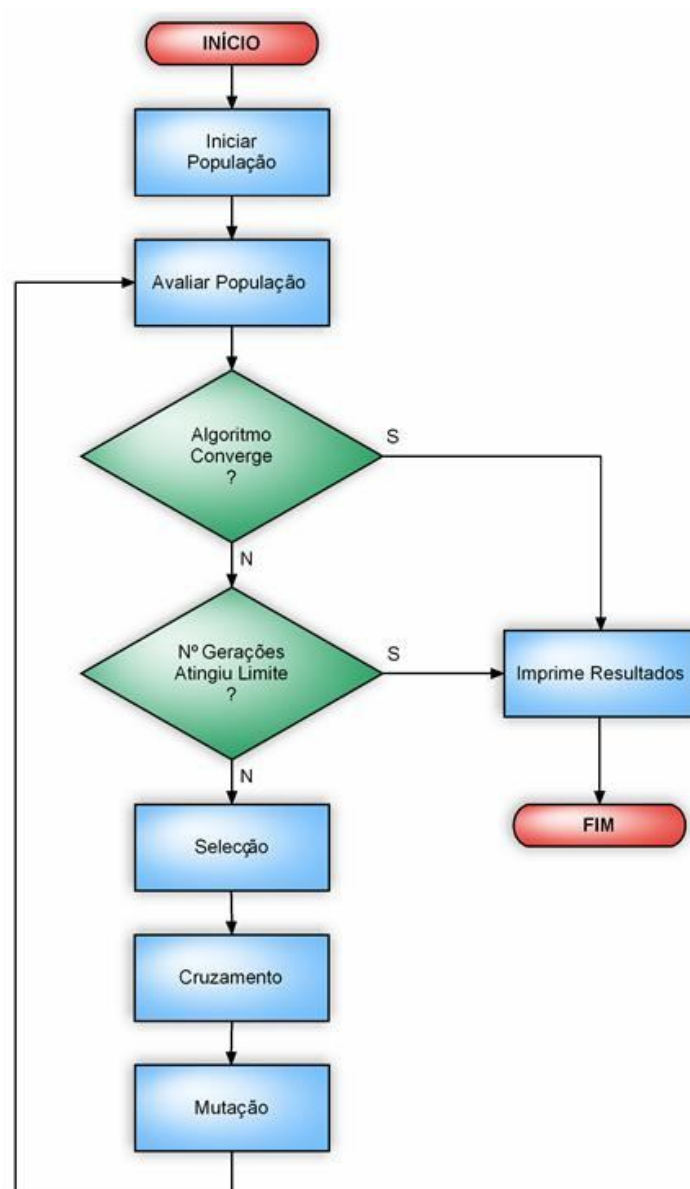


Figura 62 Estrutura base do Algoritmo Genético desenvolvido

7.3.1. FUNÇÃO PARA INICIAR A POPULAÇÃO

Esta função (Figura 63) apresenta quatro métodos diferentes para se criar a população inicial.

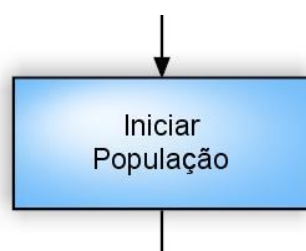


Figura 63 Função para iniciar a população

O primeiro método é o método aleatório, ou seja, são gerados valores aleatórios para cada gene de um cromossoma que pertencem ao intervalo de restrição de cada um deles e, no final, são calculados os valores dos índices E_{av} e ε_{xyH} e também a distância percorrida pelo robô. Caso a solução encontrada não seja possível de implementar (solução impossível), o valor retornado do índice E_{av} é o valor -1111. Neste caso é necessário gerar novos números aleatórios, para cada um dos genes, e voltar a calcular os valores dos índices E_{av} e ε_{xyH} e a distância percorrida pelo robô, até se atingir o tamanho da população indicado pelo utilizador.

No segundo método, é feita a divisão do espaço de busca de cada um dos genes em n intervalos de tamanho igual. Para cada um dos intervalos são gerados x cromossomas ($x = \text{tamanho da população} / n$) de maneira a ser atingido o número de indivíduos pretendido.

O terceiro método é similar ao primeiro método, sendo a única diferença o facto de serem gerados ($n * \text{tamanho população}$) cromossomas e, no final, só sobreviverem os melhores (tamanho população) cromossomas. Este método tem como vantagem, relativamente ao primeiro, a cobertura de um maior espaço de busca.

No quarto e último método, o utilizador insere os parâmetros do robô obtidos previamente, quer através da execução de outros métodos, quer pela execução de outro AG. Após a introdução dos valores dos parâmetros do robô é calculado o valor dos índices E_{av} e ε_{xyH} e a distância percorrida e o resto da população é gerada em torno dos valores dos parâmetros inseridos pelo utilizador.

7.3.2. FUNÇÃO PARA AVALIAR A POPULAÇÃO

A função de avaliação da população (Figura 64) está dividida em duas partes. Na primeira parte é ordenado o vector, consoante o parâmetro que se pretende otimizar: caso o objectivo seja minimizar a energia, o vector é ordenado para que o valor mais próximo do objectivo (menor valor de E_{av}) seja o primeiro do vector e o valor mais afastado do objectivo ocupe a última posição deste. O mesmo acontece quando o objectivo é minimizar o valor do erro de seguimento da trajectória do robô (menor valor de ε_{xyH}).

Quando o objectivo é otimizar ambos os parâmetros, foi necessário recorrer a um conceito ligeiramente diferente, uma vez que a minimização de um dos parâmetros não corresponde necessariamente à minimização do outro. Desta forma recorreu-se ao método da Frente Ótima de Pareto, ou seja, os cromossomas que apresentam simultaneamente os menores valores dos índices E_{av} e ε_{xyH} são colocados nas primeiras posições do vector.

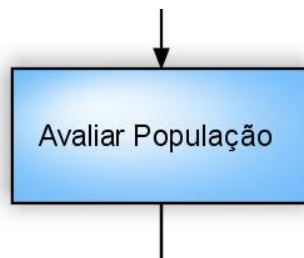


Figura 64 Função para avaliar a população

Na segunda parte da função de avaliação da população são atribuídos os valores de aptidão: as primeiras posições do vector ordenado apresentam maiores valores de aptidão e, por sua vez, as últimas posições do vector têm valores de aptidão inferiores. A probabilidade de um cromossoma ser seleccionado é tanto maior quanto maior for o seu valor de aptidão. Os valores de aptidão estão relacionados com o intervalo de pressão previamente indicado pelo utilizador.

7.3.3. FUNÇÃO PARA ANALISAR A CONVERGÊNCIA DO AG

A função de análise de convergência do AG (Figura 65) efectua a comparação do valor presente na primeira posição do vector de cromossomas, previamente ordenado, com o valor da posição que corresponde a 90% do tamanho da população (por exemplo, no caso do tamanho da população ser de dez indivíduos, é comparado o valor da primeira posição do vector com o valor da posição nove). Caso o valor destas duas posições seja igual, o algoritmo convergiu e irá terminar a execução escrevendo os valores otimizados num ficheiro de texto. Caso o valor seja diferente, o AG continuará o seu ciclo de execução.

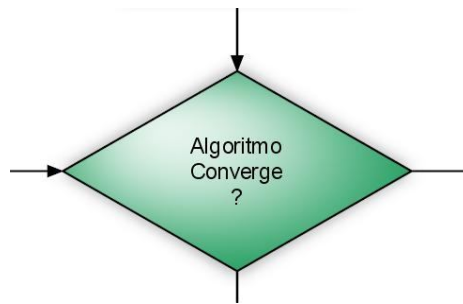


Figura 65 Função para analisar a convergência do AG

7.3.4. FUNÇÃO PARA ANALISAR O NÚMERO DE GERAÇÕES CRIADAS PELO AG

A função de análise do número de gerações criadas pelo AG (Figura 66) efectua a comparação do valor limite de gerações criadas pelo AG (número indicado pelo utilizador) com o número de gerações criadas até ao momento. Caso o valor seja igual, o AG termina a execução escrevendo os valores otimizados num ficheiro de texto. Caso o valor seja diferente, o AG continuará o seu ciclo, aplicando os operadores genéticos aos cromossomas que compõem a população.



Figura 66 Função de análise do número de gerações criadas pelo AG

7.3.5. FUNÇÃO DE SELECÇÃO DOS CROMOSSOMAS

Esta função (Figura 67) apresenta três métodos diferentes para a selecção dos cromossomas que irão gerar descendentes na próxima geração.

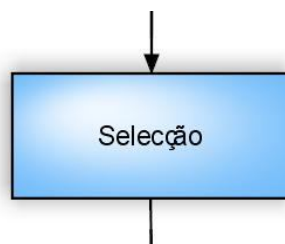


Figura 67 Função de selecção dos cromossomas

O primeiro método é o de selecção aleatória. Neste método, os cromossomas são seleccionados de forma aleatória, logo não é tido em conta se o cromossoma seleccionado é mais ou menos apto, ou seja, se está mais ou menos próximo do valor que se pretende atingir.

O segundo método é o de selecção consoante a aptidão, ou seja, quanto maior for o seu valor de aptidão maior a probabilidade de ser seleccionado para a população intermédia. Desta forma, os cromossomas que estão mais próximos do valor que se pretende atingir têm uma maior probabilidade de serem seleccionados.

O terceiro método é o de selecção por torneio, ou seja, são sempre escolhidos n cromossomas da população e apenas o que apresentar um valor de aptidão mais elevado passa para a população intermédia. Neste caso, o número de cromossomas seleccionados é indicado pelo utilizador.

7.3.6. FUNÇÃO DE CRUZAMENTO DOS CROMOSSOMAS

Esta função (Figura 68) dispõe de três métodos diferentes para se realizar o cruzamento de dois cromossomas da população, de forma a obter-se um (ou dois) novo(s) cromossoma(s).

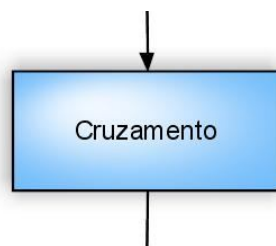


Figura 68 Função de cruzamento dos cromossomas

O primeiro método é o de cruzamento BLX- α ($\alpha = 0,5$). Neste caso é gerado um número aleatório contido no intervalo $[-\alpha, 1+\alpha]$ para cada um dos genes do cromossoma que foi seleccionado para realizar o cruzamento. Cada gene do filho produzido pelo cruzamento é obtido através da seguinte fórmula:

$$filho1_i = pai1_i + \beta(pai2_i - pai1_i) \quad \text{onde } \beta \in [-\alpha, 1 + \alpha]$$

O segundo método é o de cruzamento Aritmético. Nesta situação é gerado um número aleatório contido no intervalo [0, 1] para cada um dos genes do cromossoma que foi seleccionado para realizar o cruzamento e cada gene dos filhos produzidos pelo cruzamento é obtido através da seguinte fórmula:

$$filho1_i = \beta \times pai1_i + (1 - \beta) \times pai2_i$$

$$filho2_i = (1 - \beta) \times pai1_i + \beta \times pai2_i \quad \text{onde } \beta \in [0,1]$$

O terceiro método é o de cruzamento Heurístico. Neste método é gerado um número aleatório contido no intervalo [0, 1] para cada um dos genes do cromossoma que foi seleccionado para realizar o cruzamento, sendo cada gene do filho produzido pelo cruzamento obtido através da seguinte fórmula:

$$filho_i = pai1_i + r \times (pai1_i - pai2_i) \quad \text{onde } r \in [0,1]$$

7.3.7. FUNÇÃO DE MUTAÇÃO DOS CROMOSSOMAS

Esta função (Figura 69) implementa três métodos diferentes para a realização da mutação dos genes dos cromossomas da população.

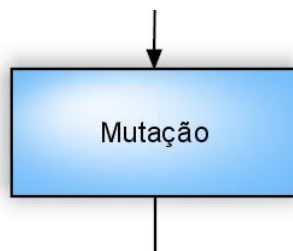


Figura 69 Função de mutação dos cromossomas

O primeiro método é o de mutação uniforme. O que este método faz é a substituição do gene seleccionado para mutação por um valor aleatório compreendido entre os limites do intervalo permitido para o gene em causa.

O segundo método é o de mutação *creep*. Este método efectua a multiplicação do gene seleccionado para mutação por um valor aleatório próximo de um. Este método apresenta como vantagem o facto de não ser muito destrutivo, apenas provocando uma ligeira perturbação do valor do gene.

O terceiro método é o método de mutação limite. A aplicação deste método leva à substituição do gene seleccionado por um dos limites do intervalo de restrição do gene em causa, sendo aleatoriamente atribuído o valor mínimo ou máximo do intervalo.

7.4. ALGORITMO GENÉTICO DESENVOLVIDO

Para a realização deste trabalho foi desenvolvido um algoritmo genético, em linguagem C, que permite obter os parâmetros do robô de maneira a se atingir um sistema de locomoção optimizado, ou seja, atingir um baixo consumo energético e um erro de seguimento da trajectória planeada igualmente reduzido.

O programa pede inicialmente ao utilizador para inserir alguns parâmetros e escolher quais os métodos para a execução do AG. Assim, nesta secção serão explicados todos os menus com os quais o utilizador se irá deparar.

O algoritmo apresenta diversas opções, sendo que a primeira opção (Figura 70) permite ao utilizador escolher entre correr o algoritmo genético para obter os parâmetros optimizados do robô ou, então, introduzir manualmente os parâmetros de um robô e obter o valor da energia consumida, o erro de seguimento da trajectória e a distância percorrida por ele.

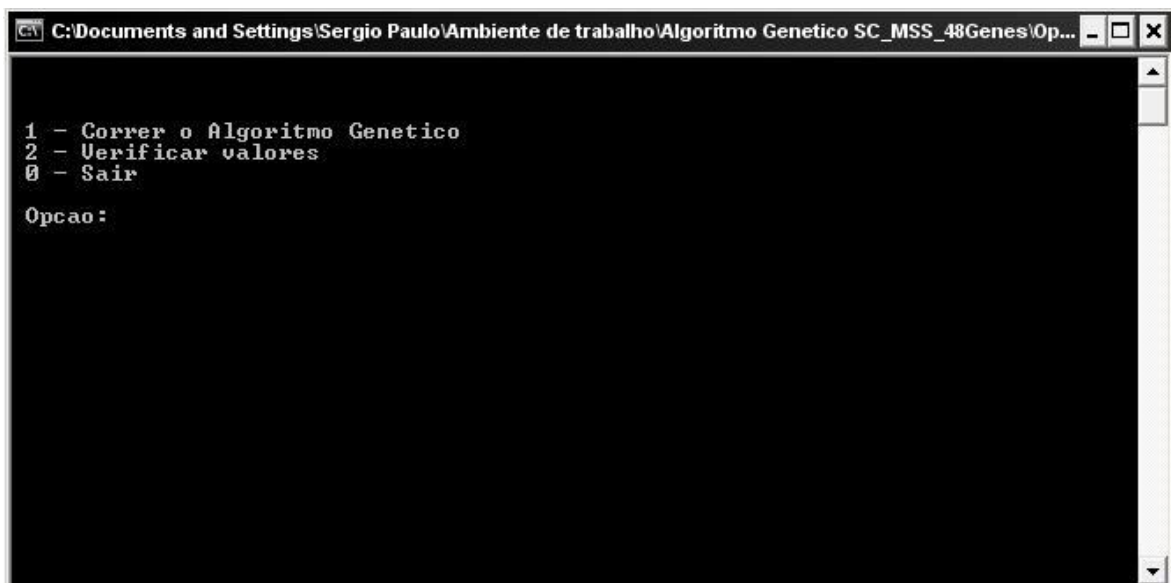
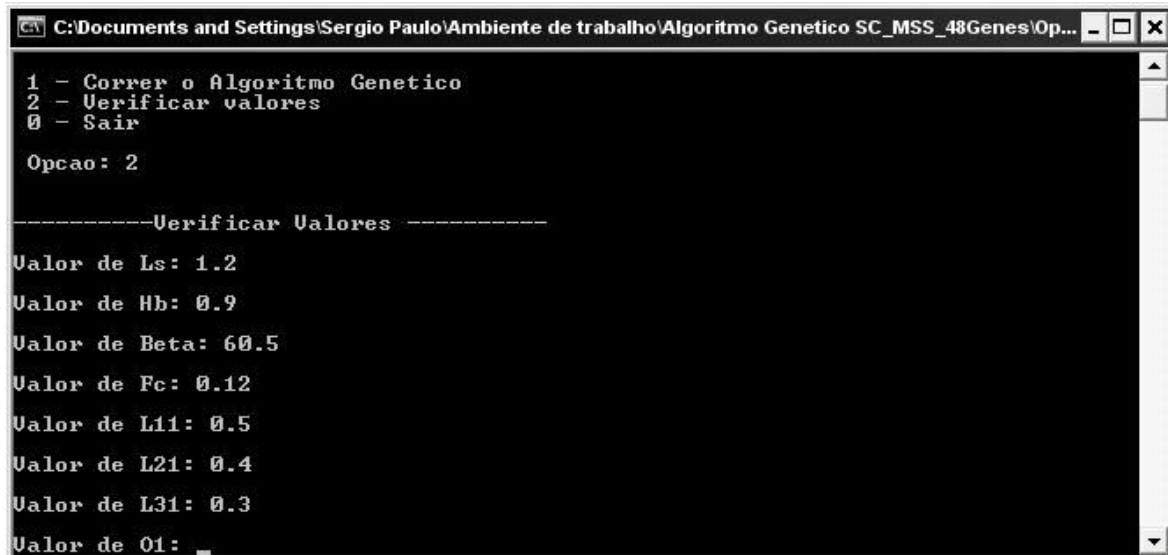


Figura 70 Selecção de execução do AG ou verificar valores

Caso o utilizador opte por introduzir manualmente os parâmetros de um robô, para obter os valores de energia, erro de seguimento da trajectória e distância percorrida, é pedido para introduzir todos os parâmetros do robô (Figura 71).



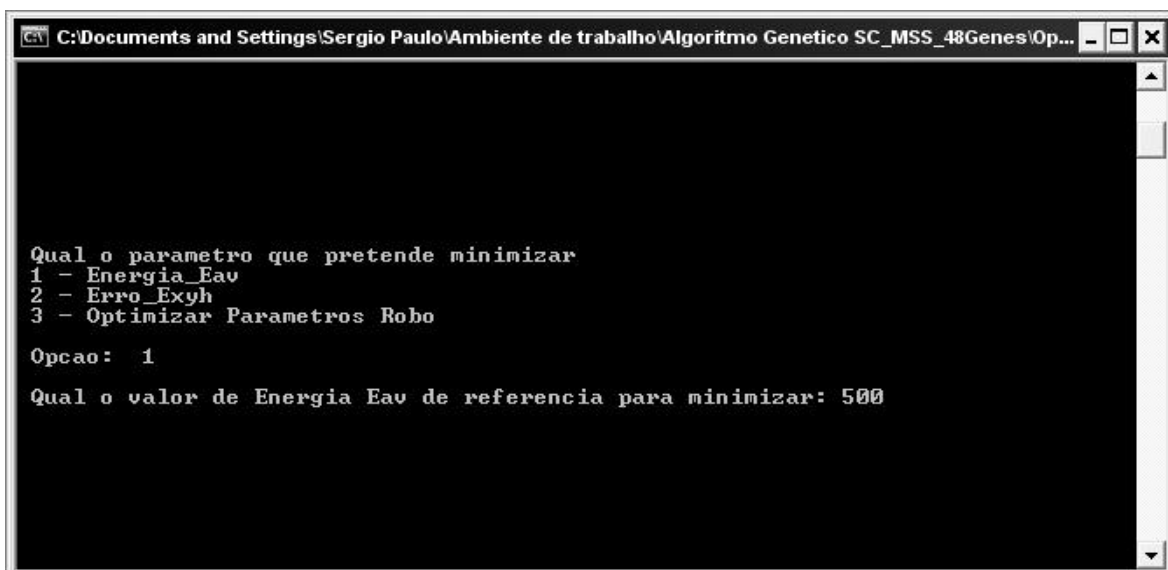
```
C:\Documents and Settings\Sergio Paulo\Ambiente de trabalho\Algoritmo Genetico SC_MSS_48Genes\Op...
1 - Correr o Algoritmo Genetico
2 - Verificar valores
0 - Sair

Opcao: 2

-----Verificar Valores -----
Valor de Ls: 1.2
Valor de Hb: 0.9
Valor de Beta: 60.5
Valor de Fc: 0.12
Valor de L11: 0.5
Valor de L21: 0.4
Valor de L31: 0.3
Valor de O1: _
```

Figura 71 Introdução dos parâmetros do robô

Caso o utilizador opte por correr o AG, como se pode ver na Figura 72, é-lhe pedido que indique qual(ais) o(s) parâmetro(s) que se pretende(m) minimizar (energia, erro ou ambos) e uma outra opção permite ao utilizador introduzir um valor de energia ou de erro de maneira a obter os parâmetros do robô (i.e., o AG tentará atingir o valor introduzido pelo utilizador).



```
C:\Documents and Settings\Sergio Paulo\Ambiente de trabalho\Algoritmo Genetico SC_MSS_48Genes\Op...
Qual o parametro que pretende minimizar
1 - Energia_Eav
2 - Erro_Exyh
3 - Optimizar Parametros Robo

Opcao: 1

Qual o valor de Energia Eav de referencia para minimizar: 500
```

Figura 72 Seleccção do parâmetro(s) que se pretende(m) minimizar

Após a escolha do(s) parâmetro(s) que se pretende(m) minimizar, é solicitado ao utilizador que indique qual o padrão de locomoção que o robô irá apresentar. Para tal, o utilizador tem à sua escolha seis opções (Figura 73).

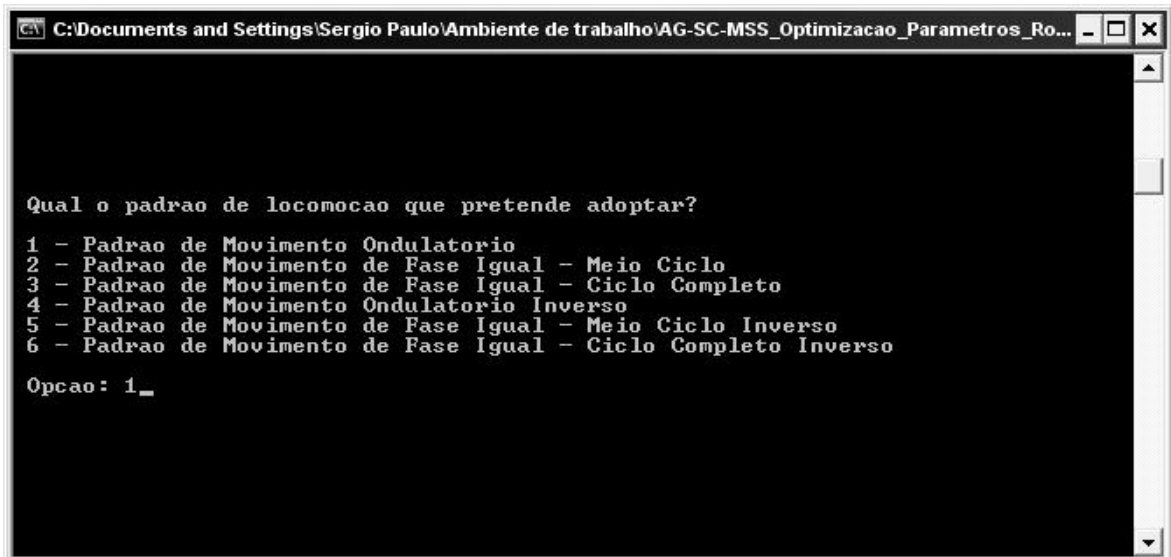
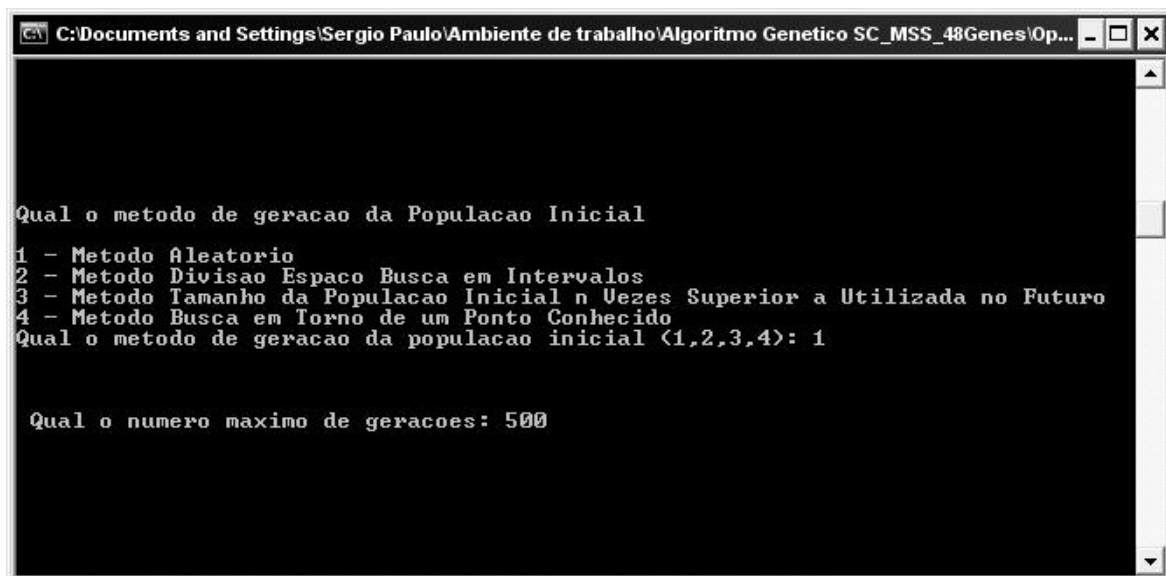


Figura 73 Selecção do padrão de locomoção que o robô irá apresentar

Para a geração da população inicial existem quatro opções (Figura 74). Numa primeira opção são gerados indivíduos de forma aleatória; como segunda opção é dividido o espaço de busca em intervalos de tamanho igual e são gerados n indivíduos por cada intervalo e a terceira opção gera um número de indivíduos n vezes superior ao que irá ser utilizado no decorrer do AG. A quarta opção difere ligeiramente das anteriores, uma vez que, neste caso, se exige um conhecimento à priori dos parâmetros do robô e permite iniciar o algoritmo num ponto previamente conhecido.

Ao utilizador também é pedido para indicar qual o número máximo de gerações que o AG pode criar.

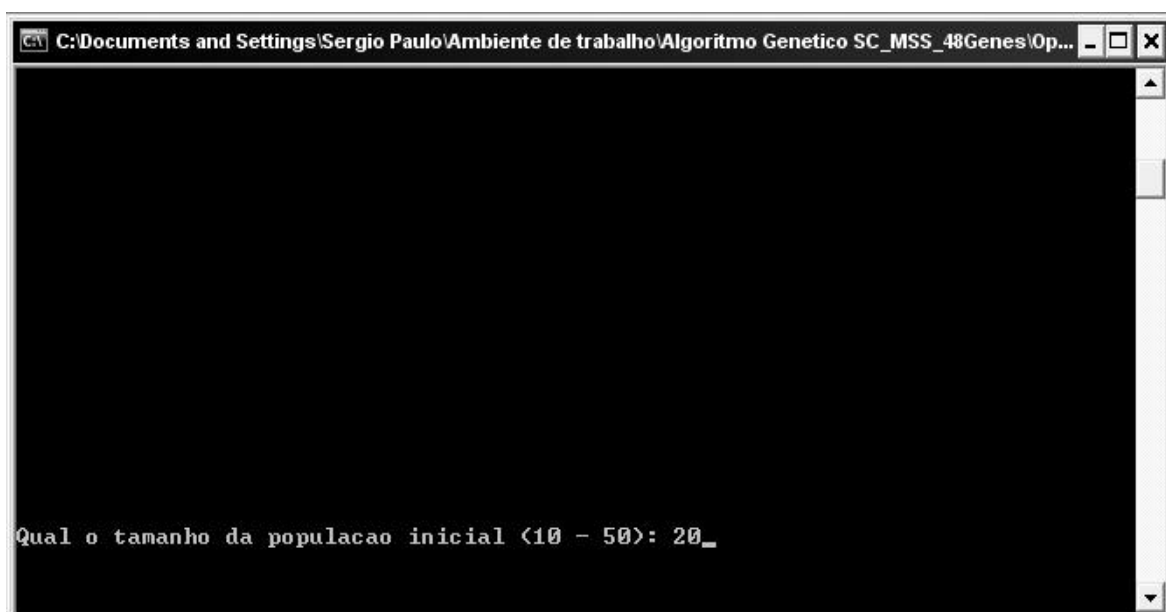


```
C:\Documents and Settings\Sergio Paulo\Ambiente de trabalho\Algoritmo Genetico SC_MSS_48Genes\Op...
Qual o metodo de geracao da Populacao Inicial
1 - Metodo Aleatorio
2 - Metodo Divisao Espaco Busca em Intervalos
3 - Metodo Tamanho da Populacao Inicial n Uezes Superior a Utilizada no Futuro
4 - Metodo Busca em Torno de um Ponto Conhecido
Qual o metodo de geracao da populacao inicial <1,2,3,4>: 1

Qual o numero maximo de geracoes: 500
```

Figura 74 Seleção do método de geração da população inicial

De seguida, ao utilizador é-lhe pedido que indique qual o tamanho da população inicial, o qual se mantém igual ao longo da execução do AG. O tamanho indicado tem de ser um número entre dez e cinquenta, como é visível na Figura 75.



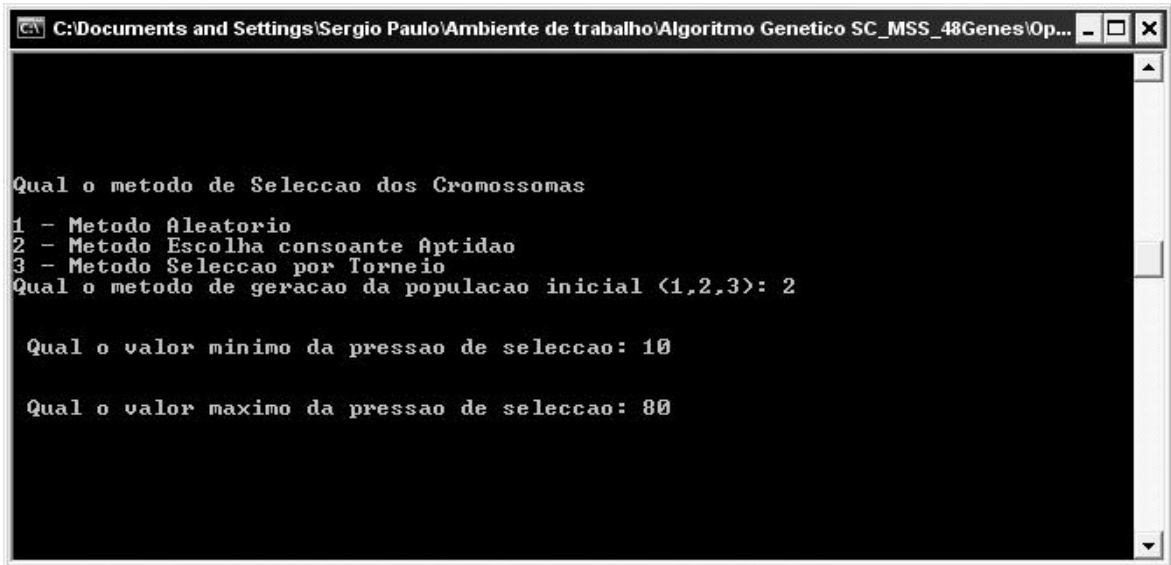
```
C:\Documents and Settings\Sergio Paulo\Ambiente de trabalho\Algoritmo Genetico SC_MSS_48Genes\Op...
Qual o tamanho da populacao inicial <10 - 50>: 20_
```

Figura 75 Introdução do tamanho da população inicial

Após este passo, é solicitado ao utilizador que indique qual o método de selecção dos cromossomas que irão gerar novos cromossomas, quer através do cruzamento, quer através da mutação (Figura 76). Na primeira opção os cromossomas são seleccionados de forma aleatória; na segunda opção os cromossomas mais aptos têm maior probabilidade

de serem seleccionados e na terceira opção são escolhidos n cromossomas de forma aleatória, mas apenas o melhor destes passa para uma geração intermédia.

Também será pedido ao utilizador que indique qual o intervalo da pressão de selecção, isto é, qual a probabilidade de selecção do cromossoma mais e menos apto (Figura 76).



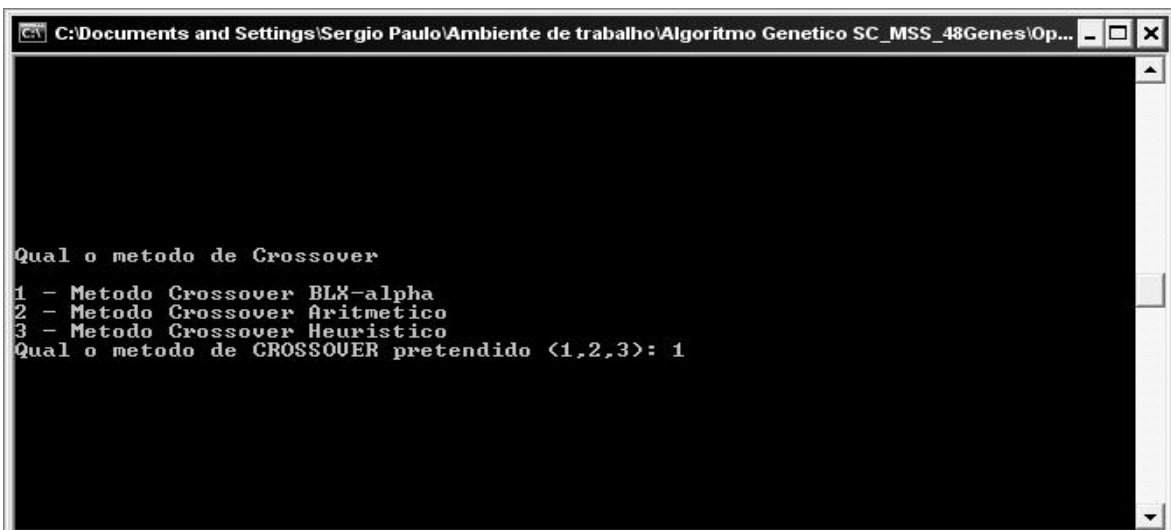
```
C:\Documents and Settings\Sergio Paulo\Ambiente de trabalho\Algoritmo Genetico SC_MSS_48Genes\Op...
Qual o metodo de Seleccao dos Cromossomas
1 - Metodo Aleatorio
2 - Metodo Escolha consoante Aptidao
3 - Metodo Seleccao por Torneio
Qual o metodo de geracao da populacao inicial <1,2,3>: 2

Qual o valor minimo da pressao de seleccao: 10

Qual o valor maximo da pressao de seleccao: 80
```

Figura 76 Escolha do método de selecção dos cromossomas

De seguida, o utilizador terá de indicar qual o método de cruzamento que pretende que o AG use. Foram implementados três métodos distintos (Figura 77), métodos estes que já foram descritos na secção 5.3.1. O primeiro método é o cruzamento BLX- α , o segundo é o cruzamento aritmético e, por fim, o terceiro é o cruzamento heurístico.



```
C:\Documents and Settings\Sergio Paulo\Ambiente de trabalho\Algoritmo Genetico SC_MSS_48Genes\Op...
Qual o metodo de Crossover
1 - Metodo Crossover BLX-alpha
2 - Metodo Crossover Aritmetico
3 - Metodo Crossover Heuristico
Qual o metodo de CROSSOVER pretendido <1,2,3>: 1
```

Figura 77 Selecção do método de cruzamento

Após este passo, o utilizador deve escolher qual o método de substituição da geração dos pais pela geração dos filhos. Para o efeito existem duas opções (Figura 78): na primeira o(s) filho(s) substituem os pais que os geraram (substituição geracional) e na segunda opção (substituição *steady-state*) o(s) filho(s) gerados substituem os piores pais da população (estes dois métodos já foram abordados na secção 5.7.4).



```
C:\Documents and Settings\Sergio Paulo\Ambiente de trabalho\Algoritmo Genetico SC_MSS_48Genes\Op... _ _ X
1 - Metodo de substituicao geracional
2 - Metodo de substituicao steady-state
Qual o metodo de substituicao <1,2>: 1
Qual a taxa de Crossover 60 e 90: 80_
```

Figura 78 Selecção do método de substituição das gerações

Após a opção por um ou outro método é pedido ao utilizador para indicar qual a taxa de cruzamento (Figura 78) e, de seguida, que escolha qual o método de mutação (Figura 79). Aqui existem três opções: na primeira opção o método de mutação é a mutação uniforme; o segundo método é a mutação *creep* e na terceira opção adopta-se a mutação limite, tendo todos estes métodos sido abordados na secção 5.4.3. Após seleccionar a opção correspondente ao método de mutação pretendido é pedido ao utilizador que indique qual a taxa de mutação.

```
C:\Documents and Settings\Sergio Paulo\Ambiente de trabalho\Algoritmo Genetico SC_MSS_48Genes\Op...
Qual o metodo de Mutacao
1 - Metodo Mutacao Uniforme
2 - Metodo Mutacao Creep
3 - Metodo Mutacao Limite
Qual o metodo de MUTACAO pretendido <1,2,3>: 2
Qual a taxa de Mutacao 0.1 e 5: 2
```

Figura 79 Seleção do método de mutação

Por fim, o utilizador tem de indicar quantos indivíduos passam de uma geração para a seguinte sem sofrer qualquer tipo de alteração (taxa de elitismo) (Figura 80). Estes indivíduos são apelidados de “cromossomas elite” ou “super-indivíduos”.

```
C:\Documents and Settings\Sergio Paulo\Ambiente de trabalho\Algoritmo Genetico SC_MSS_48Genes\Op...
Qual o numero de cromossomas 'Elite'
Numero de individuos que passam para geracao seguinte <Elitismo> 1-4: 1
```

Figura 80 Seleção do número de cromossomas elite

7.5. CONCLUSÃO

Neste capítulo foram descritas as características mais relevantes do sistema de locomoção multipernas usado, dando maior relevância aos aspectos que interessa otimizar, tais como as características das pernas do robô, características do corpo, etc. Foram, de igual modo, abordados os parâmetros que se pretendem minimizar, nomeadamente o índice E_{av} baseado na dinâmica do sistema robô-solo e o índice ε_{xyH} baseado nos erros de seguimento das trajectórias das ancas do robô. Foi igualmente abordado neste capítulo o AG desenvolvido para a realização deste trabalho e foram apresentadas todas as opções que este dispõe para a sua execução.

No próximo capítulo serão apresentados os resultados que se pensa serem mais significativos obtidos através dos diversos ensaios realizados ao longo do trabalho.

8. ENSAIOS E RESULTADOS DAS SIMULAÇÕES

Com este capítulo encerra-se o trabalho e ao longo deste são apresentados todos os ensaios realizados com o AG desenvolvido, os parâmetros óptimos do sistema de locomoção e a influência da velocidade (V_f) nos mesmos.

Num primeiro ponto são apresentados todos os ensaios realizados com o AG desenvolvido de forma a verificar o seu correcto funcionamento e também quais os métodos que apresentam melhores resultados (métodos em que o AG converge mais rapidamente). Estes ensaios são importantes, porque a escolha do melhor método permite a obtenção de resultados de forma mais rápida.

Num segundo ponto serão apresentados os parâmetros óptimos do sistema robótico e o respectivo resultado da simulação para cada um dos padrões de locomoção (PLO, PLFIMC, PLFICC, PLOI, PLFIMCI e PLFICCI).

No final são apresentados diversos gráficos onde se mostra a influência da velocidade de deslocamento em diversos parâmetros do sistema de locomoção. Serão usadas velocidades desde 0,1 m/s até 10 m/s.

8.1. ENSAIOS DO AG

Para realizar os primeiros testes do funcionamento do AG, foram executados diversos ensaios utilizando todos os métodos disponíveis para a sua execução. Estes ensaios foram realizados não só para testar o correcto funcionamento do AG, como também para se ficar a conhecer quais os melhores métodos disponíveis no algoritmo.

Na Tabela 1 são apresentadas as opções seleccionadas nos diversos ensaios do AG, nomeadamente quanto aos métodos de geração da população inicial, de selecção, cruzamento, substituição das gerações, mutação e elitismo.

Tabela 1 Configurações dos diversos ensaios do Algoritmo Genético

	Geração População	Seleccção	Cruzamento	Substituição Gerações	Mutação	Elitismo
Ensaio 1	Aleatório	Aleatório	BLX- α	Geracional	Uniforme	1
Ensaio 2	Intervalos	Aptidão	Aritmético	Geracional	Limite	1
Ensaio 3	N*Tamanho	Torneio	Aritmético	Geracional	<i>Creep</i>	1
Ensaio 4	Aleatório	Aleatório	Aritmético	<i>Steady-State</i>	Uniforme	1
Ensaio 5	Intervalos	Torneio	Heurístico	<i>Steady-State</i>	<i>Creep</i>	1
Ensaio 6	Aleatório	Torneio	Aritmético	Geracional	Limite	1
Ensaio 7	N*Tamanho	Torneio	Heurístico	<i>Steady-State</i>	Uniforme	1
Ensaio 8	Intervalos	Aptidão	Heurístico	Geracional	Uniforme	1
Ensaio 9	N*Tamanho	Aleatório	BLX- α	<i>Steady-State</i>	<i>Creep</i>	1
Ensaio 10	Intervalos	Torneio	Heurístico	<i>Steady-State</i>	Uniforme	1
Ensaio 11	Aleatório	Aleatório	Aritmético	<i>Steady-State</i>	Limite	1
Ensaio 12	N*Tamanho	Aptidão	Heurístico	Geracional	<i>Creep</i>	1

Em todos estes ensaios, o padrão de locomoção adoptado foi o padrão ondulatório e o parâmetro que se pretendia minimizar era o índice E_{av} . O comportamento de cada um dos métodos descritos na tabela anterior está representado na Figura 81.

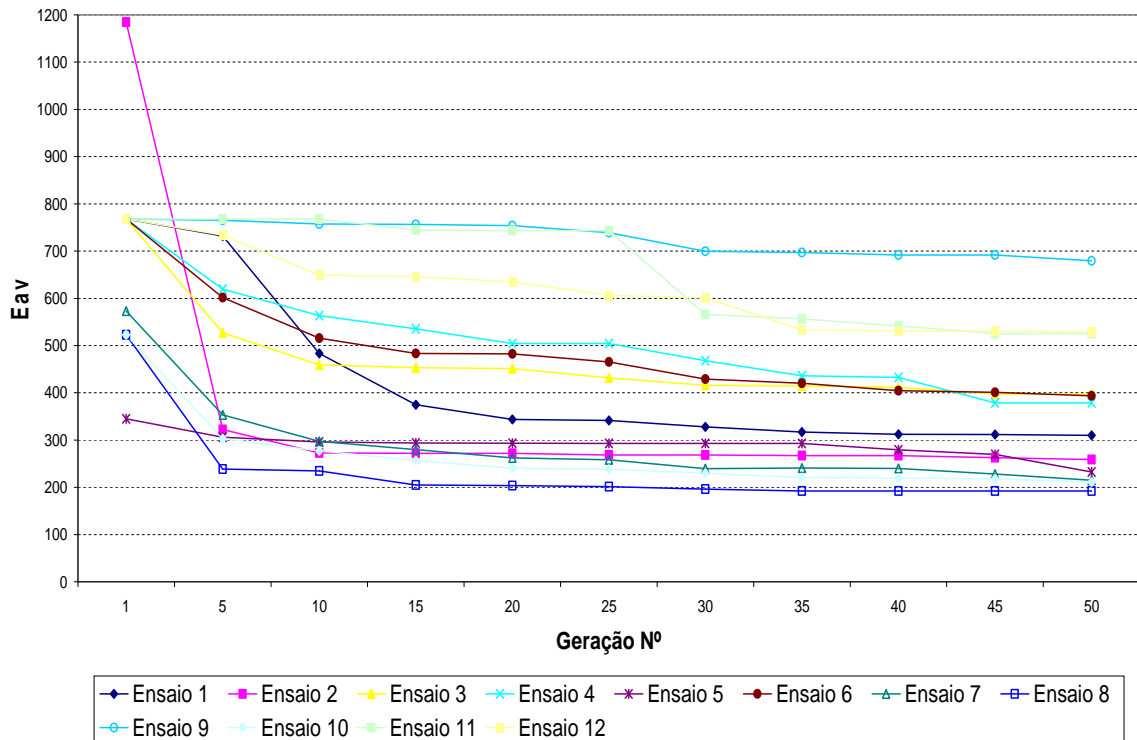


Figura 81 Comportamento dos diversos métodos do AG

Como se pode verificar através deste gráfico, os métodos que apresentam uma convergência mais rápida são o método 8, o método 10 e o método 7, respectivamente. Apesar destes resultados, convém ter a noção que na execução de um método do AG repetidas vezes, este pode apresentar outros tantos valores diferentes, ou seja, não apresenta necessariamente os mesmos valores nas mesmas gerações, de cada vez que é executado. No entanto, para a obtenção dos parâmetros otimizados do robô, deveremos optar por um destes três métodos, uma vez que se pretende atingir esses valores da forma mais rápida possível.

Um outro ensaio que foi realizado está relacionado com o intervalo de pressão de selecção. Este intervalo define qual a probabilidade que cada um dos cromossomas presentes na população tem de ser seleccionado para a próxima geração. Na Tabela 2 são apresentados os valores inferiores e superiores do intervalo de pressão de selecção.

Tabela 2 Ensaios do AG para diversos intervalos de pressão

	Intervalo Pressão 1	Intervalo Pressão 2	Intervalo Pressão 3	Intervalo Pressão 4	Intervalo Pressão 5
Mín. Intervalo	0	15	20	25	25
Máx. Intervalo	100	85	70	50	40

O comportamento do AG, com estes diversos intervalos de pressão de selecção, está representado na Figura 82. Mais uma vez, o objectivo era minimizar o valor do índice E_{av} . As restantes opções do AG foram as mesmas para todos estes ensaios (padrão de locomoção, tamanho da população, método de selecção, método de cruzamento, método de mutação, etc.).

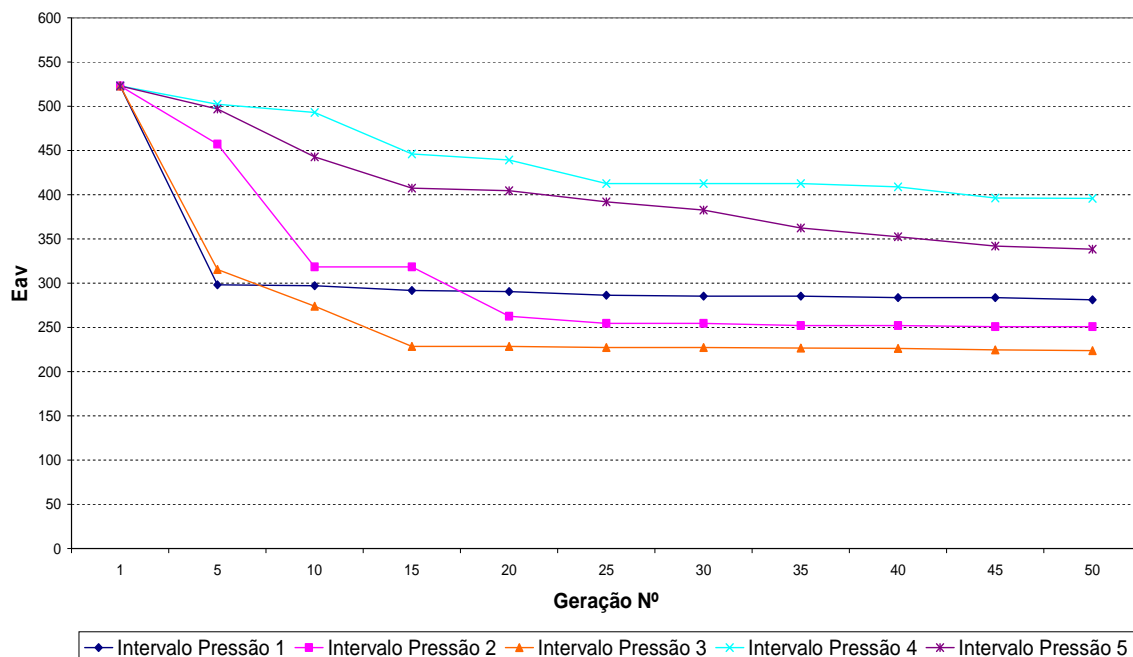


Figura 82 Influência dos intervalos de pressão no comportamento do AG

Como se pode observar através deste gráfico, as melhores opções correspondem ao intervalo de pressão 3 e ao intervalo de pressão 2. Estes dois intervalos apresentam uma boa amplitude, mas também atribuem alguma probabilidade aos piores cromossomas da população de poderem gerar descendência. Assim, para a obtenção dos parâmetros otimizados do robô deveremos adoptar um intervalo de pressão de selecção próximo do intervalo de pressão 3, ou seja, valor mínimo do intervalo igual a 20 e valor máximo do intervalo igual a 70.

O comportamento do AG para os dois diferentes métodos de substituição das gerações está representado na Figura 83, sendo o objectivo minimizar o valor do índice E_{av} . Na aplicação dos métodos Subs. Geracional 1 e *Steady-State* 1, as restantes opções do AG foram iguais, para os métodos Subs. Geracional 2 e *Steady-State* 2, as restantes opções do AG também foram iguais, mas diferentes dos dois métodos anteriores, e o mesmo aconteceu para os métodos Subs. Geracional 3 e *Steady-State* 3.

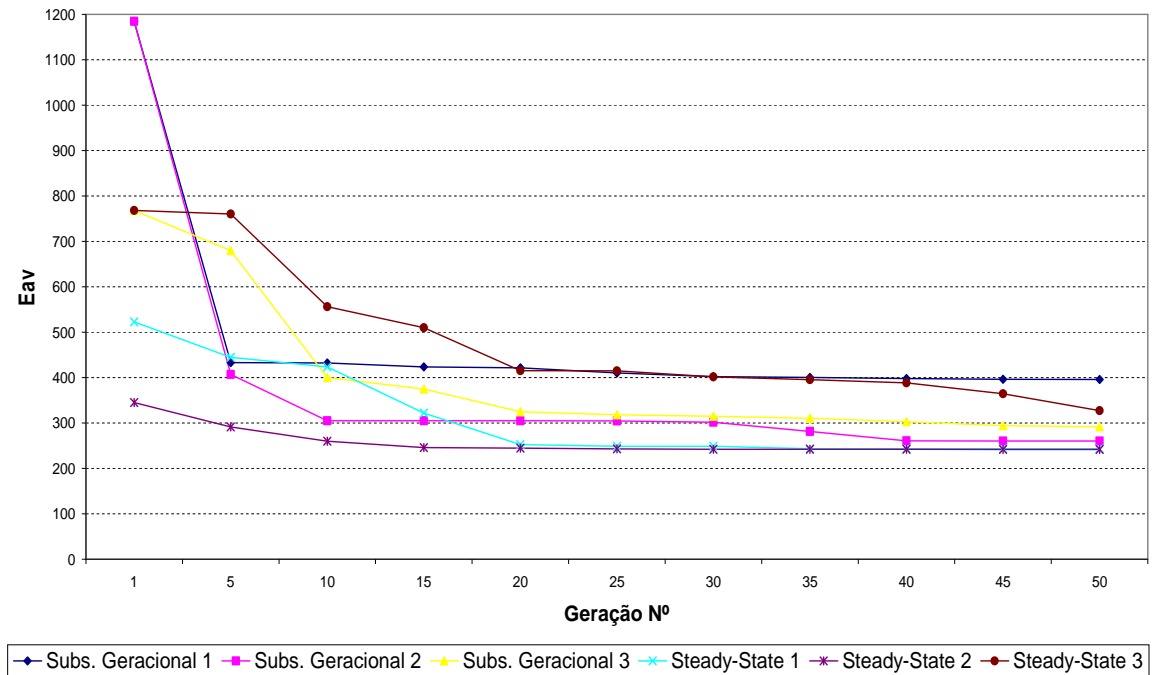


Figura 83 Influência do método de substituição no AG

Através da análise do gráfico verifica-se que não existem diferenças significativas, mas a substituição *steady-state* parece apresentar um maior índice de convergência. Em teoria, este método terá alguma vantagem uma vez que os piores cromossomas são sempre substituídos; já no caso da substituição geracional, um “mau filho” pode substituir um “bom pai”.

Um outro ensaio realizado consistiu em verificar a influência do elitismo na execução do AG. Os resultados correspondentes a estes ensaios estão representados na Figura 84. Mais uma vez, o objectivo do AG era minimizar o valor do índice E_{av} . Tal como no caso anterior as restantes opções do AG foram iguais para cada par de métodos (Com Elitismo 1 e Sem Elitismo 1, Com Elitismo 2 e Sem Elitismo 2, Com Elitismo 3 e Sem Elitismo 3).

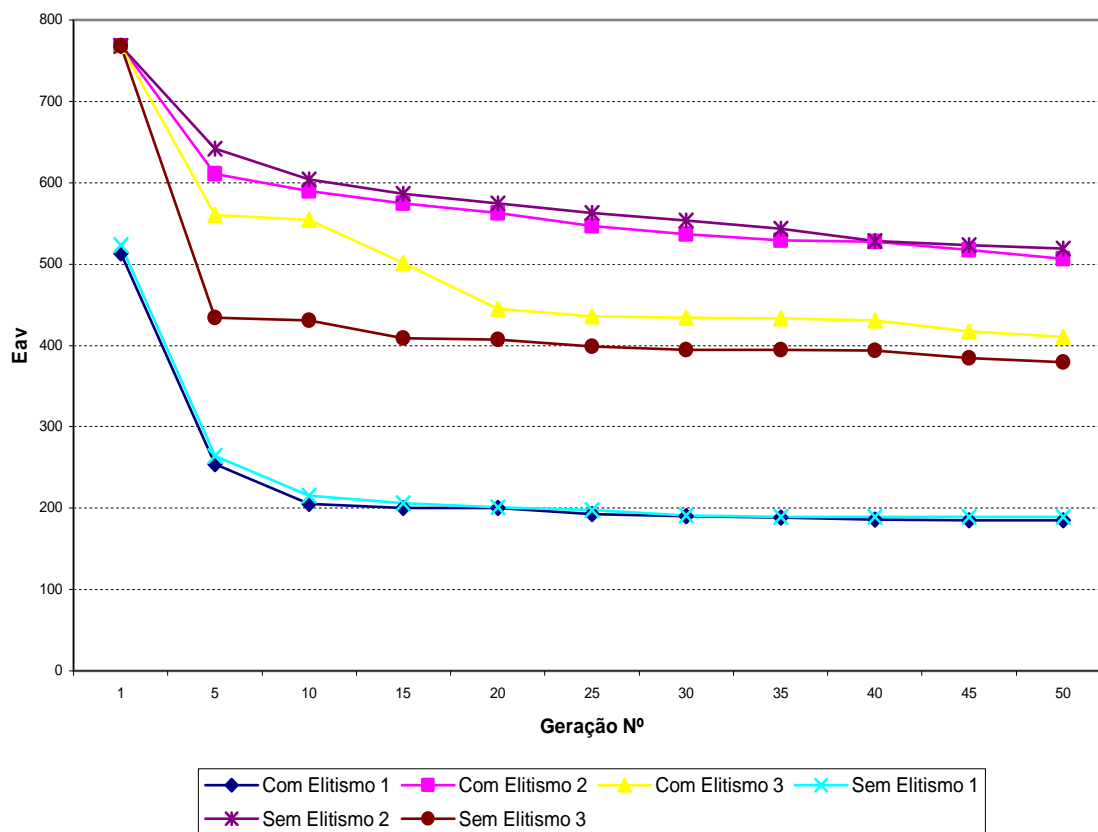


Figura 84 Influência do elitismo no AG

Os resultados obtidos neste teste foram uma surpresa, uma vez que se esperavam melhores resultados aquando da utilização do elitismo em relação à sua não utilização, uma vez que o uso do elitismo é prevenir a eventual perda da melhor solução obtida até ao momento. Aquilo que poderá estar a ocorrer é que uma ligeira alteração dos valores dos genes do cromossoma elite pode fazer com este se torne numa solução ainda melhor.

Após estes ensaios passou-se à execução do algoritmo genético desenvolvido de maneira a se obterem os parâmetros óptimos do robô, para os diversos padrões de locomoção implementados. As opções adoptadas na execução do algoritmo genético têm em conta os resultados obtidos nestes ensaios prévios, de forma a se atingirem os valores desejados da maneira mais rápida possível, poupando tempo e evitando o esforço computacional desnecessário.

8.2. OBTENÇÃO DOS PARÂMETROS ÓPTIMOS DO SISTEMA DE LOCOMOÇÃO

Neste primeiro ensaio foi assumido o valor de 1 m/s para a velocidade de deslocamento do robô, sendo o cromossoma utilizado para a execução do AG composto por 48 genes (48 parâmetros do robô). O objectivo deste primeiro ensaio era a obtenção dos parâmetros óptimos do robô para cada um dos padrões de locomoção.

Serão apresentadas as diversas soluções obtidas pelo AG ao longo da sua execução, bem como o valor final, ou seja, os valores otimizados dos diversos parâmetros, e será apresentado também o resultado da simulação do sistema de locomoção.

No final desta secção será apresentado um resumo com os diversos parâmetros otimizados obtidos pelo AG para cada um dos padrões de locomoção.

8.2.1. PADRÃO DE LOCOMOÇÃO ONDULATÓRIO

Para a obtenção dos parâmetros otimizados do sistema de locomoção, o AG desenvolvido foi executado três vezes: numa primeira execução o objectivo foi minimizar o valor da densidade de energia absoluta média (E_{av}), na segunda execução o objectivo passou por minimizar o valor do erro do seguimento das trajectórias das ancas (ε_{xyH}) e na terceira execução o objectivo consistiu em minimizar ambos os índices.

Na Figura 85 apresentam-se várias soluções possíveis para a implementação do sistema de locomoção que se pretende otimizar. O gráfico apresenta três zonas distintas; se o objectivo for a minimização da energia consumida, pode-se verificar que o valor de E_{av} apresenta um valor próximo de 200 Jm^{-1} a que corresponde um valor de ε_{xyH} próximo de 0,6 m. Por outro lado, se o objectivo for minimizar o erro de seguimento da trajectória das ancas (ε_{xyH}), obtém-se um valor próximo de 0,15 m a que corresponde um valor de E_{av} de aproximadamente 1200 Jm^{-1} . Por fim, se o objectivo for minimizar ambas as medidas de desempenho simultaneamente, é possível constatar que o índice E_{av} apresenta um valor na ordem dos 325 Jm^{-1} e o valor de ε_{xyH} é aproximadamente de 0,35 m.

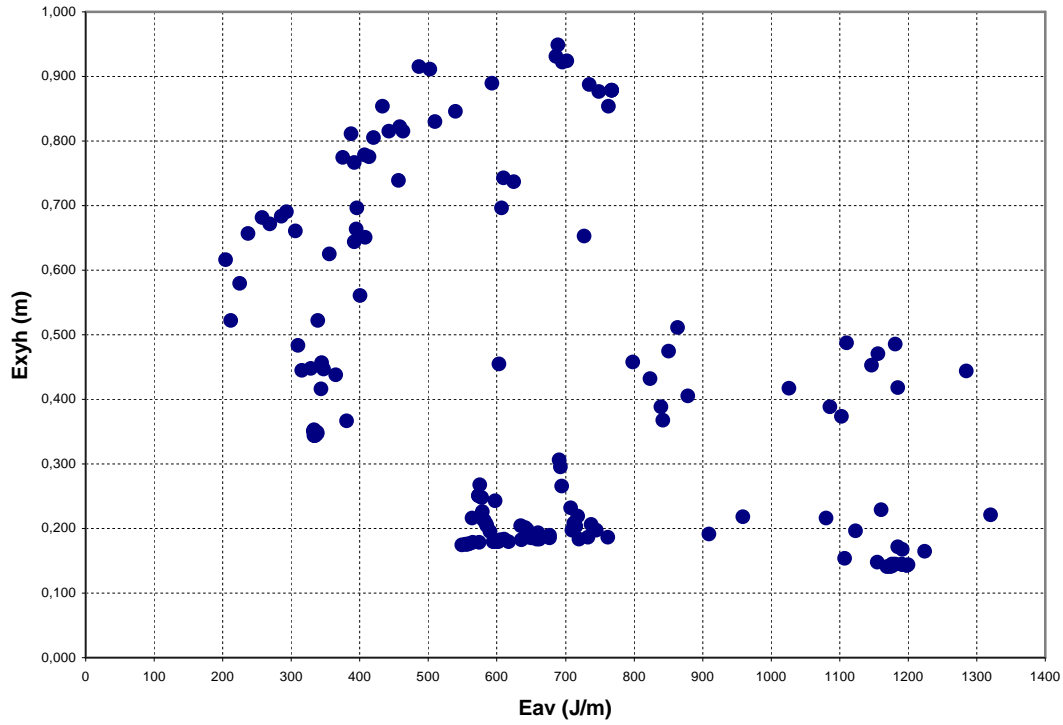


Figura 85 Soluções obtidas pelo AG – PLO

Como o objectivo deste trabalho é a optimização dos parâmetros do robô, ou seja, minimizar simultaneamente ambos os índices (E_{av} e ε_{xyH}), é apresentado na Figura 86 o resultado da simulação do robô, quando este apresenta os valores óptimos determinados pelo AG. Os correspondentes parâmetros do robô são apresentados na Tabela 3.

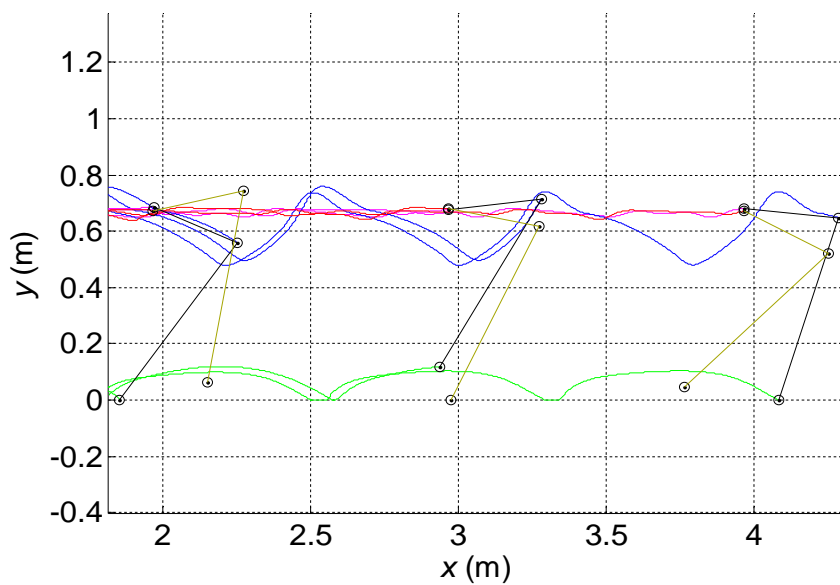


Figura 86 Simulação do sistema de locomoção – PLO

Tabela 3 Parâmetros otimizados do robô (padrões de locomoção directos)

PLO	PLFIMC	PLFICC
Ls= 0,798 m Hb= 0,685 m Beta= 34,112 % Fc= 0,125 m L11= 0,321 m L12= 0,679 m L21= 0,314 m L22= 0,686 m L31= 0,311 m L32= 0,689 m O1= -0,606 m O2= -0,546 m O3= -0,657 m Mb= 84,138 kg M11= 3,634 kg M12= 1,723 kg M21= 3,574 kg M22= 1,449 kg M31= 2,959 kg M32= 2,523 kg Kxh= 89106,766 N/m Kyh= 9990,477 N/m Bxh= 776,511 N/m Byh= 90,151 N/m T11min= -358,508 Nm T11max= 176,209 Nm T12min= -288,704 Nm T12max= 53,051 Nm T21min= -264,891 Nm T21max= 75,424 Nm T22min= -229,980 Nm T22max= 156,389 Nm T31min= -386,089 Nm T31max= 123,213 Nm T32min= -378,953 Nm T32max= 80,422 Nm Kp11= 943,627 Kd11= 336,111 Kp12= 3582,081 Kd12= 14,327 Kp21= 831,258 Kd21= 100,013 Kp22= 3948,079 Kd22= 30,294 Kp31= 3934,615 Kd31= 183,397 Kp32= 1275,400 Kd32= 109,285 Velocidade= 1,000 m/s	Ls= 0,996 m Hb= 0,733 m Beta= 42,972 % Fc= 0,173 m L11= 0,521 m L12= 0,479 m L21= 0,449 m L22= 0,551 m L31= 0,426 m L32= 0,574 m O1= -0,335 m O2= -0,325 m O3= -0,413 m Mb= 71,124 kg M11= 4,498 kg M12= 5,561 kg M21= 3,937 kg M22= 6,261 kg M31= 3,921 kg M32= 4,698 kg Kxh= 95452,297 N/m Kyh= 11012,926 N/m Bxh= 883,493 N/m Byh= 97,097 N/m T11min= -212,238 Nm T11max= 241,178 Nm T12min= -200,241 Nm T12max= 240,882 Nm T21min= -216,674 Nm T21max= 202,856 Nm T22min= -219,617 Nm T22max= 230,988 Nm T31min= -232,821 Nm T31max= 233,590 Nm T32min= -226,783 Nm T32max= 247,631 Nm Kp11= 3213,644 Kd11= 331,793 Kp12= 3928,315 Kd12= 393,434 Kp21= 3384,143 Kd21= 348,813 Kp22= 3954,050 Kd22= 382,481 Kp31= 4167,394 Kd31= 356,461 Kp32= 4456,879 Kd32= 371,735 Velocidade= 1,000 m/s	Ls= 0,996 m Hb= 0,733 m Beta= 42,972 Fc= 0,173 m L11= 0,502 m L12= 0,498 m L21= 0,449 m L22= 0,551 m L31= 0,411 m L32= 0,589 m O1= -0,328 m O2= -0,301 m O3= -0,413 m Mb= 71,124 kg M11= 4,498 kg M12= 5,561 kg M21= 3,937 kg M22= 6,261 kg M31= 3,921 kg M32= 4,698 kg Kxh= 95452,289 N/m Kyh= 11012,926 N/m Bxh= 883,491 N/m Byh= 97,097 N/m T11min= -212,238 Nm T11max= 240,257 Nm T12min= -193,803 Nm T12max= 240,883 Nm T21min= -216,674 Nm T21max= 202,678 Nm T22min= -219,617 Nm T22max= 230,988 Nm T31min= -232,821 Nm T31max= 233,590 Nm T32min= -226,783 Nm T32max= 247,631 Nm Kp11= 3213,645 Kd11= 331,881 Kp12= 3928,315 Kd12= 393,434 Kp21= 3384,143 Kd21= 373,052 Kp22= 3954,048 Kd22= 382,481 Kp31= 4167,401 Kd31= 356,461 Kp32= 4456,879 Kd32= 398,928 Velocidade= 1,000 m/s
E_{av} = 334,135 J/m	E_{av} = 566,696 J/m	E_{av} = 572,301 J/m
ε_{xyh} = 0,344 m	ε_{xyh} = 0,419 m	ε_{xyh} = 0,429 m
Distância= 0,789 m	Distância= 0,990 m	Distância= 0,990 m

8.2.2. PADRÃO DE LOCOMOÇÃO DE FASE IGUAL E MEIO CICLO

Para a obtenção dos parâmetros óptimos do sistema robótico de locomoção, o AG foi executado por três vezes. Tal como no caso anterior, numa primeira execução o objectivo foi minimizar o valor da densidade de energia absoluta média (E_{av}), na segunda execução o objectivo passou por minimizar o valor do erro de seguimento das trajectórias das ancas (ε_{xyH}) e na terceira execução o objectivo consistia em minimizar simultaneamente ambos os índices.

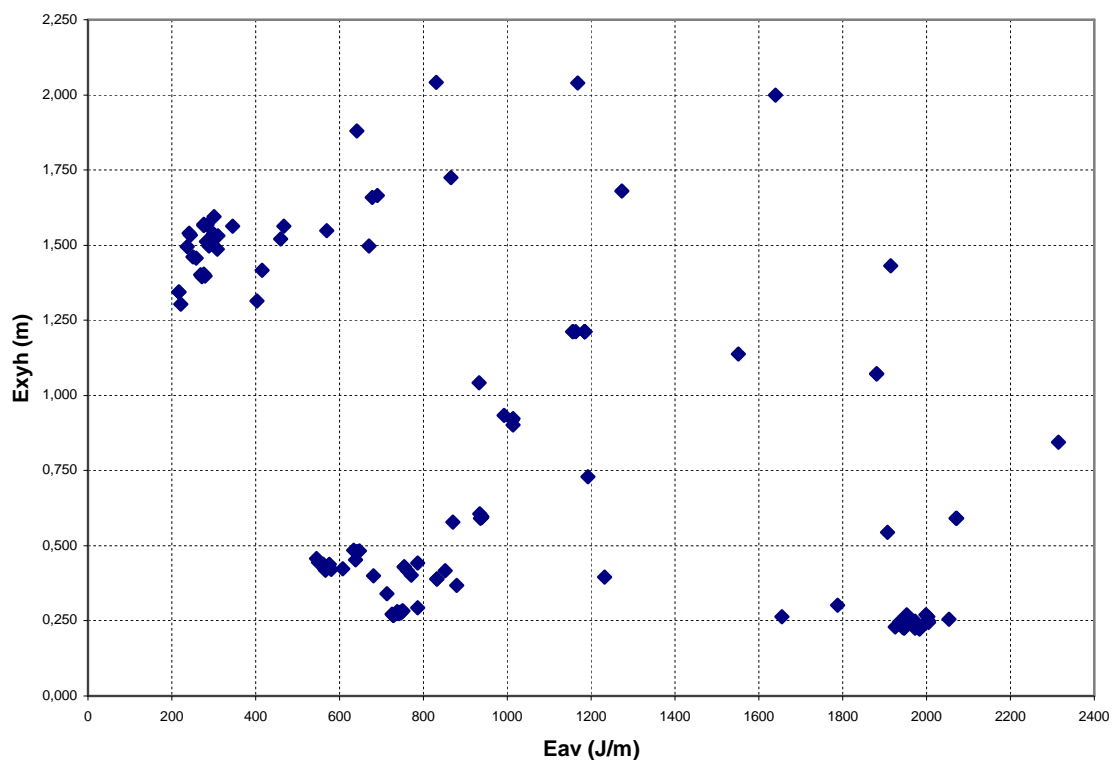


Figura 87 Soluções obtidas pelo AG – PLFIMC

Nesta figura podem-se ver várias soluções possíveis para a implementação do sistema de locomoção que se pretende otimizar. O gráfico apresenta três zonas distintas; se o objectivo for a minimização da energia consumida, pode-se verificar que o valor do índice E_{av} apresenta um valor próximo de 200 Jm^{-1} o que corresponde a um erro no seguimento da trajectória das ancas (ε_{xyH}) de aproximadamente 1,3 m. Por outro lado, se o objectivo for minimizar o erro do seguimento da trajectória das ancas (ε_{xyH}), obtém-se um valor de 0,25 m a que corresponde um valor de E_{av} de aproximadamente 2000 Jm^{-1} . Por fim, se o

objectivo for minimizar ambos os índices, podemos constatar que o valor de E_{av} apresenta um valor na ordem dos 575 Jm^{-1} e o valor de ε_{xyH} é de aproximadamente 0,4 m.

Na Figura 88 é apresentado o resultado da simulação do robô, quando este apresenta os valores otimizados pelo AG. Os parâmetros óptimos do robô para esta situação são apresentados na Tabela 3.

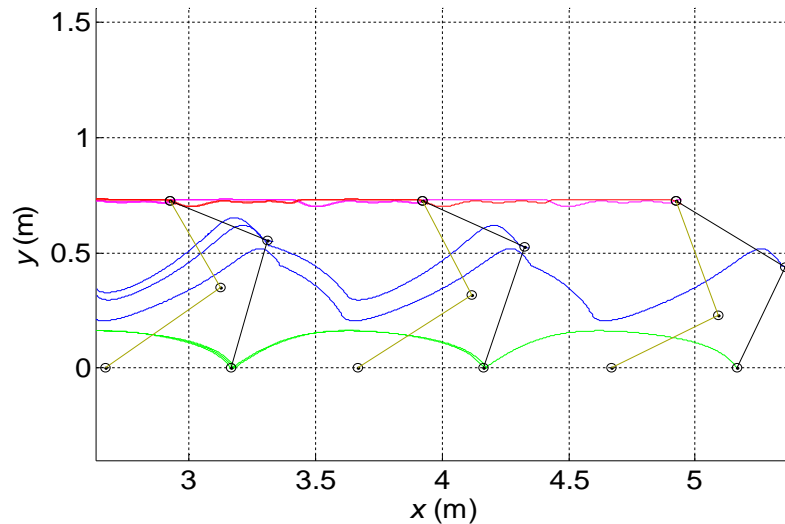


Figura 88 Simulação do sistema de locomoção – PLFIMC

8.2.3. PADRÃO DE LOCOMOÇÃO DE FASE IGUAL E CICLO COMPLETO

Para este padrão de locomoção, o AG foi executado, novamente, por três vezes. Numa primeira execução o objectivo foi minimizar o valor da densidade de energia absoluta média (E_{av}), na segunda execução o objectivo passou por minimizar o valor do erro de seguimento das trajectórias das ancas (ε_{xyH}) e na terceira execução o objectivo era minimizar ambos.

A Figura 89 apresenta três zonas distintas; se o objectivo for a minimização da energia consumida, pode verificar-se que o valor de E_{av} apresenta um valor próximo de 300 Jm^{-1} , a que corresponde um valor de ε_{xyH} de aproximadamente 0,475 m. Por outro lado, se o objectivo for minimizar o valor de ε_{xyH} , têm-se um valor aproximado de 0,20 m para este índice a que corresponde um valor de E_{av} de aproximadamente 1450 Jm^{-1} . Por fim, se o objectivo for minimizar ambos os índices verifica-se que E_{av} apresenta um valor na ordem dos 550 Jm^{-1} e o valor de ε_{xyH} é de aproximadamente 0,40 m.

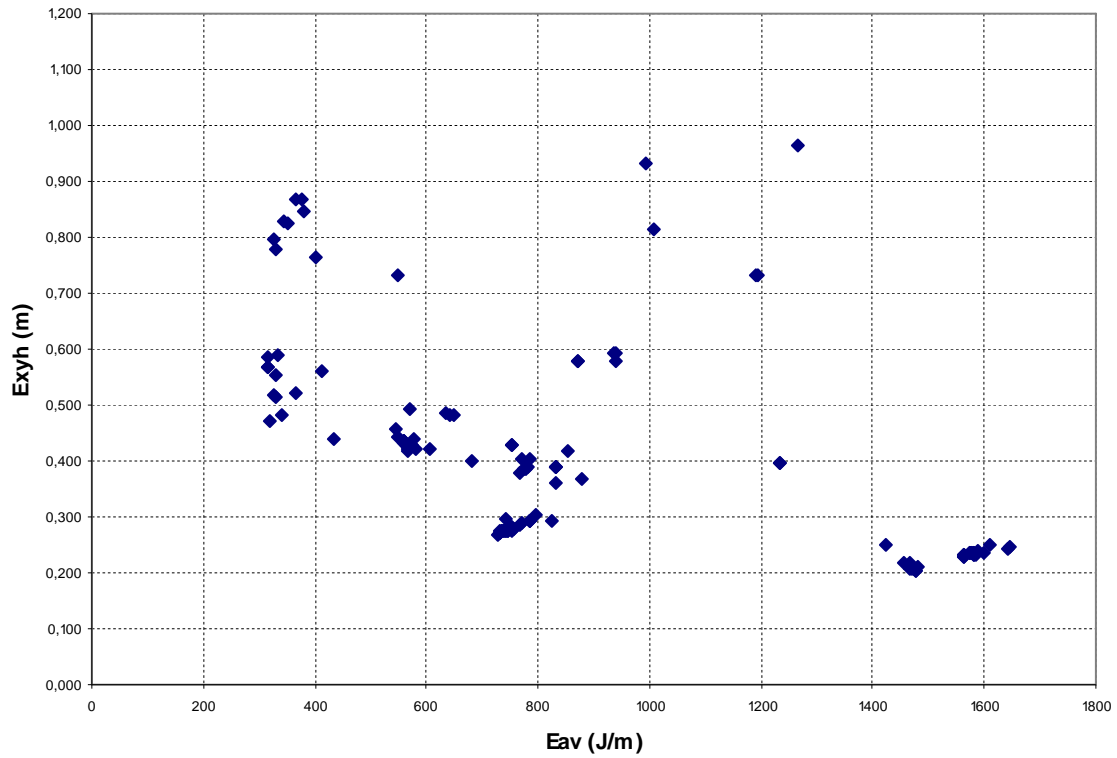


Figura 89 Soluções obtidas pelo AG – PLFICC

Na Figura 90 é apresentado o resultado da simulação do robô, quando este apresenta os valores óptimos obtidos pelo AG. Os correspondentes parâmetros do robô para este padrão de locomoção, são os apresentados na Tabela 3.

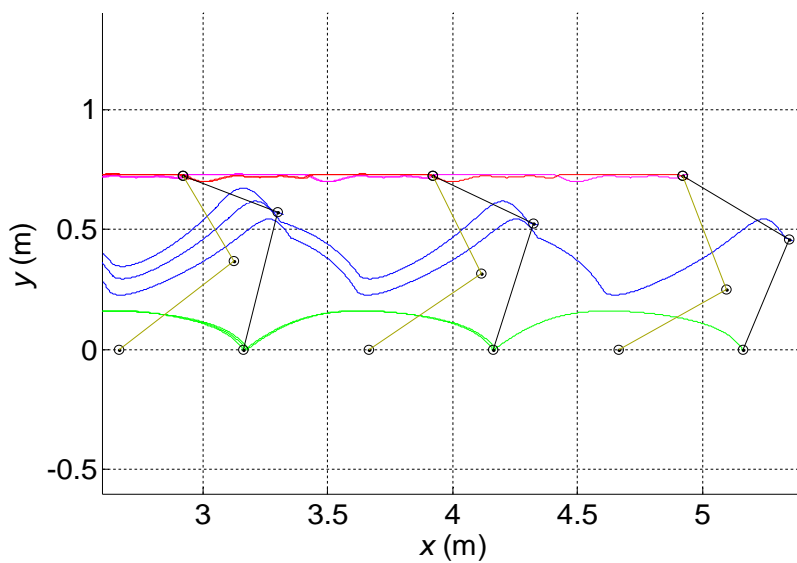


Figura 90 Simulação do sistema de locomoção – PLFICC

8.2.4. PADRÃO DE LOCOMOÇÃO ONDULATÓRIO INVERSO

Para a obtenção dos parâmetros otimizados do sistema de locomoção para o PLOI, executou-se o AG por três vezes: numa primeira execução o objectivo foi minimizar o valor de E_{av} , na segunda minimizar o valor de ε_{xyH} e na terceira minimizar ambos.

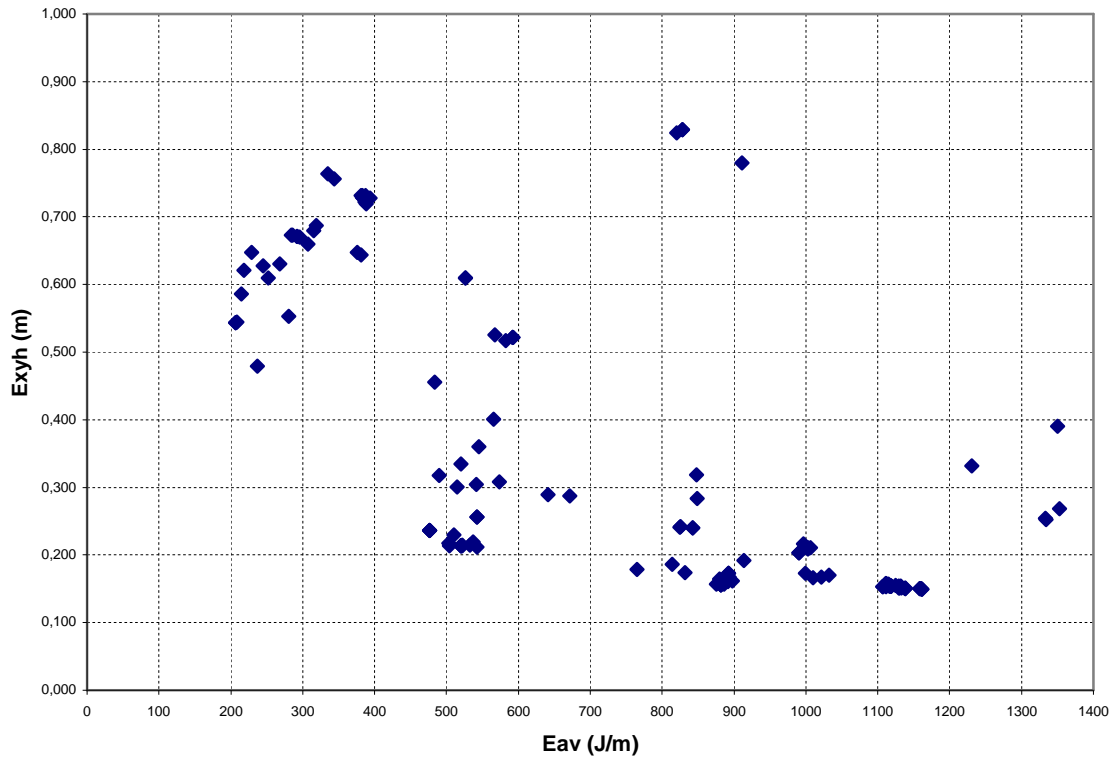


Figura 91 Soluções obtidas pelo AG – PLOI

Na Figura 91 é possível ver várias soluções possíveis para a implementação do sistema de locomoção que se pretende otimizar. Se o objectivo for a minimização da energia consumida, pode-se verificar que o índice E_{av} apresenta um valor próximo de 200 Jm^{-1} , a que corresponde um valor de ε_{xyH} próximo de 0,6 m. Se o objectivo for minimizar o índice ε_{xyH} este tem um valor aproximado de 0,15 m, a que corresponde um valor de E_{av} de aproximadamente 1100 Jm^{-1} . Por último, se o objectivo for a minimização de ambos os índices, é possível constatar que o valor de E_{av} é de 500 Jm^{-1} , ao qual corresponde um valor de ε_{xyH} de aproximadamente de 0,2 m.

Mostra-se na Figura 92 o resultado da simulação do robô, quando este apresenta os valores óptimos determinados pelo AG para o PLOI. Os parâmetros do robô para este padrão de locomoção encontram-se na Tabela 4.

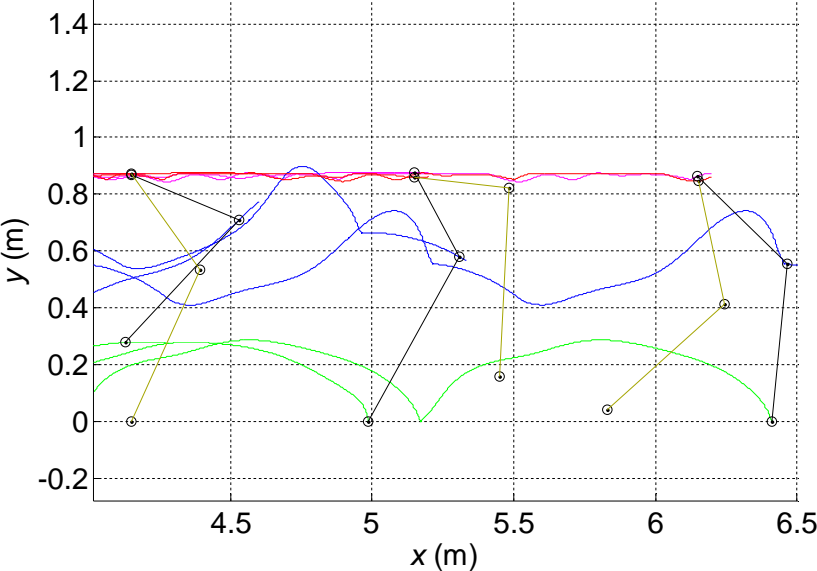


Figura 92 Simulação do sistema de locomoção – PLOI

Tabela 4 Parâmetros otimizados do robô (padrões de locomoção inversos)

PLOI	PLFIMCI	PLFICCI
Ls= 1,234 m Hb= 0,874 m Beta= 35,471 % Fc= 0,286 m L11= 0,443 m L12= 0,557 m L21= 0,335 m L22= 0,665 m L31= 0,412 m L32= 0,588 m O1= 0,131 m O2= 0,212 m O3= -0,837 m Mb= 60,840 kg M11= 2,261 kg M12= 6,025 kg M21= 5,152 kg M22= 6,088 kg M31= 4,233 kg M32= 15,400 kg Kxh= 84325,609 N/m Kyh= 10652,140 N/m Bxh= 1023,130 N/m Byh= 111,217 N/m T11min= -417,609 Nm T11max= 242,097 Nm T12min= -121,740 Nm T12max= 299,230 Nm T21min= -227,569 Nm T21max= 172,693 Nm T22min= -251,289 Nm T22max= 116,233 Nm T31min= -332,831 Nm T31max= 277,395 Nm T32min= -112,641 Nm T32max= 104,715 Nm Kp11= 1642,866 Kd11= 732,050 Kp12= 1463,171 Kd12= 53,706 Kp21= 8459,506 Kd21= 399,984 Kp22= 41,854 Kd22= 155,758 Kp31= 4303,043 Kd31= 447,706 Kp32= 2771,249 Kd32= 395,324 Velocidade= 1,000 m/s	Ls= 1,347 m Hb= 0,747 m Beta= 45,160 % Fc= 0,261 m L11= 0,417 m L12= 0,583 m L21= 0,306 m L22= 0,694 m L31= 0,312 m L32= 0,688 m O1= -0,890 m O2= 0,315 m O3= -0,803 m Mb= 64,121 kg M11= 3,456 kg M12= 5,561 kg M21= 7,502 kg M22= 6,809 kg M31= 1,333 kg M32= 11,218 kg Kxh= 84620,438 N/m Kyh= 11263,390 N/m Bxh= 1154,109 N/m Byh= 98,594 N/m T11min= -239,559 Nm T11max= 164,180 Nm T12min= -404,738 Nm T12max= 346,426 Nm T21min= -210,971 Nm T21max= 207,571 Nm T22min= -242,419 Nm T22max= 69,649 Nm T31min= -361,092 Nm T31max= 412,507 Nm T32min= -118,282 Nm T32max= 379,033 Nm Kp11= 4344,789 Kd11= 446,979 Kp12= 2175,358 Kd12= 303,462 Kp21= 5601,465 Kd21= 529,078 Kp22= 3254,481 Kd22= 242,931 Kp31= 6898,499 Kd31= 437,147 Kp32= 7846,035 Kd32= 527,509 Velocidade= 1,000 m/s	Ls= 1,406 m Hb= 0,747 m Beta= 45,160 % Fc= 0,261 m L11= 0,417 m L12= 0,583 m L21= 0,306 m L22= 0,694 m L31= 0,312 m L32= 0,688 m O1= -0,890 m O2= 0,315 m O3= -0,803 m Mb= 64,121 kg M11= 3,456 kg M12= 5,561 kg M21= 7,502 kg M22= 6,809 kg M31= 1,333 kg M32= 11,218 kg Kxh= 84620,438 N/m Kyh= 11263,390 N/m Bxh= 1154,109 N/m Byh= 98,594 N/m T11min= -239,559 Nm T11max= 164,180 Nm T12min= -404,738 Nm T12max= 346,426 Nm T21min= -210,971 Nm T21max= 207,571 Nm T22min= -242,419 Nm T22max= 69,649 Nm T31min= -361,092 Nm T31max= 412,507 Nm T32min= -118,282 Nm T32max= 379,033 Nm Kp11= 4344,789 Kd11= 446,979 Kp12= 2175,358 Kd12= 303,462 Kp21= 5601,465 Kd21= 529,078 Kp22= 3254,481 Kd22= 242,931 Kp31= 6898,499 Kd31= 437,147 Kp32= 7846,035 Kd32= 527,509 Velocidade= 1,000 m/s
$E_{av} = 476,209 \text{ Jm}^{-1}$	$E_{av} = 750,408 \text{ J/m}$	$E_{av} = 752,677 \text{ J/m}$
$\varepsilon_{xyh} = 0,236 \text{ m}$	$\varepsilon_{xyh} = 0,249 \text{ m}$	$\varepsilon_{xyh} = 0,250 \text{ m}$
Distância= 1.241 m	Distância= 1,346 m	Distância= 1,406 m

8.2.5. PADRÃO DE LOCOMOÇÃO DE FASE IGUAL E MEIO CICLO INVERSO

Para a determinação dos parâmetros óptimos para a locomoção do robô hexápode, executou-se o AG três vezes, sendo que na primeira execução o objectivo foi minimizar o valor do índice E_{av} , na segunda passou por minimizar o valor de ε_{xyH} e na terceira foi minimizar ambos.

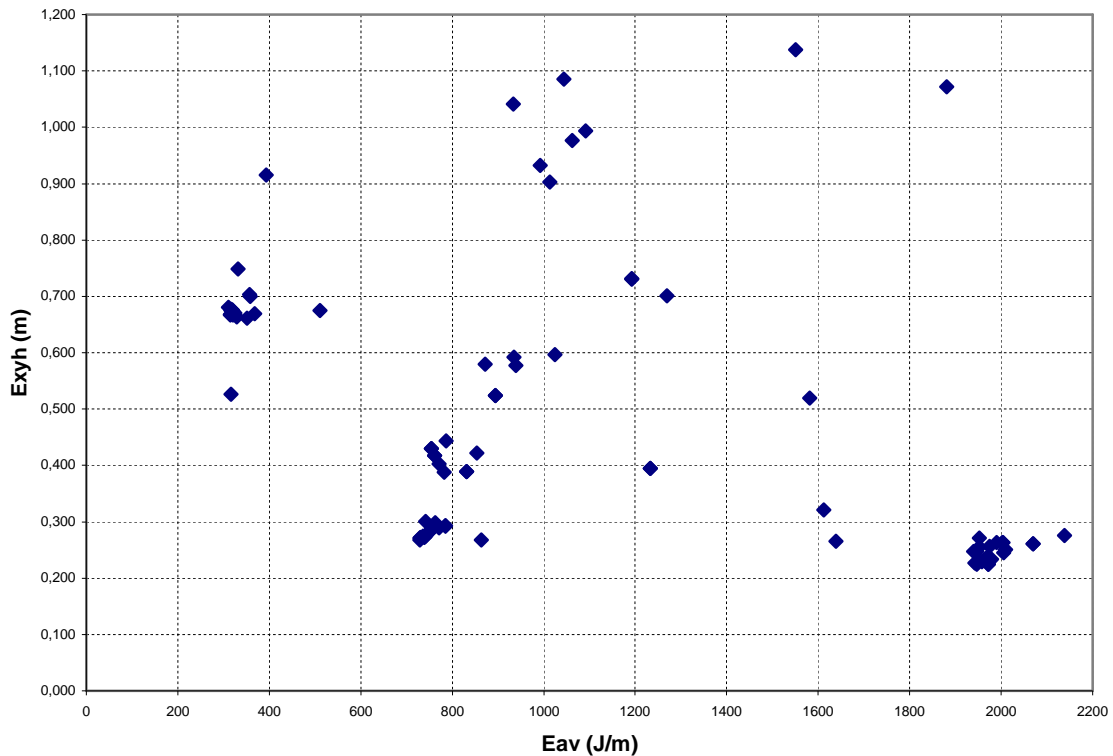


Figura 93 Soluções obtidas pelo AG – PLFIMCI

Analisando a esta figura, podem-se ver várias soluções possíveis para a implementação do sistema de locomoção que se pretende otimizar. O gráfico apresenta três zonas distintas: se o objectivo for a minimização da energia consumida, verifica-se que E_{av} apresenta um valor próximo de 300 Jm^{-1} , a que corresponde um valor de ε_{xyH} próximo de $0,7 \text{ m}$. Por outro lado, se o objectivo for minimizar o índice ε_{xyH} , este apresenta um valor aproximado de $0,2 \text{ m}$, a que corresponde um valor de E_{av} de aproximadamente 2000 Jm^{-1} . Por fim, se o objectivo for minimizar ambos, constata-se que E_{av} apresenta um valor na ordem dos 750 Jm^{-1} e ε_{xyH} é aproximadamente $0,3 \text{ m}$.

Na Figura 94 é apresentado o resultado da simulação do robô, quando este apresenta os valores óptimos. Os parâmetros do robô para esta situação são apresentados na Tabela 4.

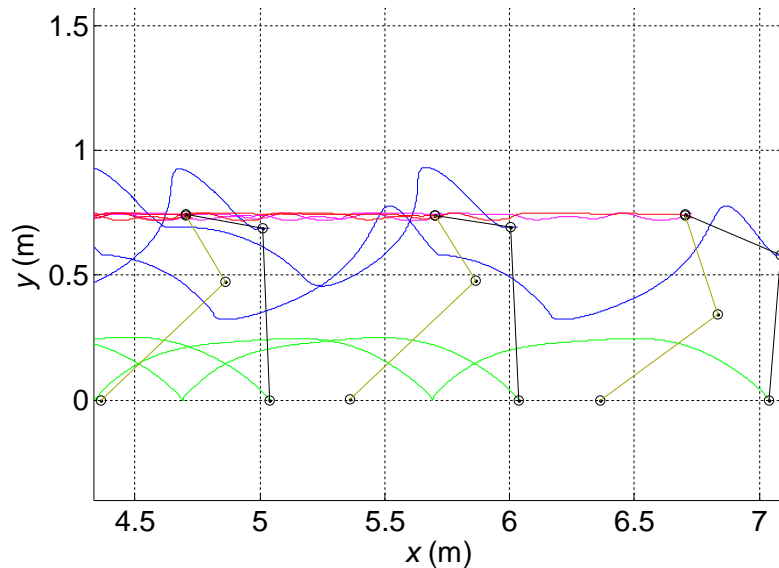


Figura 94 Simulação do sistema de locomoção - PLFIMCI

8.2.6. PADRÃO DE LOCOMOÇÃO DE FASE IGUAL E CICLO COMPLETO INVERSO

Para a obtenção dos parâmetros otimizados do sistema de locomoção hexápode, executou-se novamente o AG três vezes, sendo o objectivo na primeira execução minimizar o valor de E_{av} , na segunda execução minimizar o valor do índice ε_{xyH} e na terceira execução minimizar ambos os índices simultaneamente. Os valores dos índices correspondentes aos diversos cromossomas obtidos pelo AG encontram-se representados na Figura 95.

Da observação da figura, constata-se que este apresenta três zonas distintas. Assim, quando o objectivo é a minimização da energia, o valor de E_{av} situa-se próximo de 250 Jm^{-1} e o de seguimento da trajectória das ancas ε_{xyH} é aproximadamente $0,75 \text{ m}$. Se, por outro lado, o objectivo for minimizar ε_{xyH} , este índice apresenta um valor aproximado de $0,20 \text{ m}$, a que corresponde um valor de E_{av} de 1600 Jm^{-1} . Por fim, se o objectivo é a minimização de ambos, o valor obtido para E_{av} é de 700 Jm^{-1} e o de ε_{xyH} é aproximadamente $0,25 \text{ m}$.

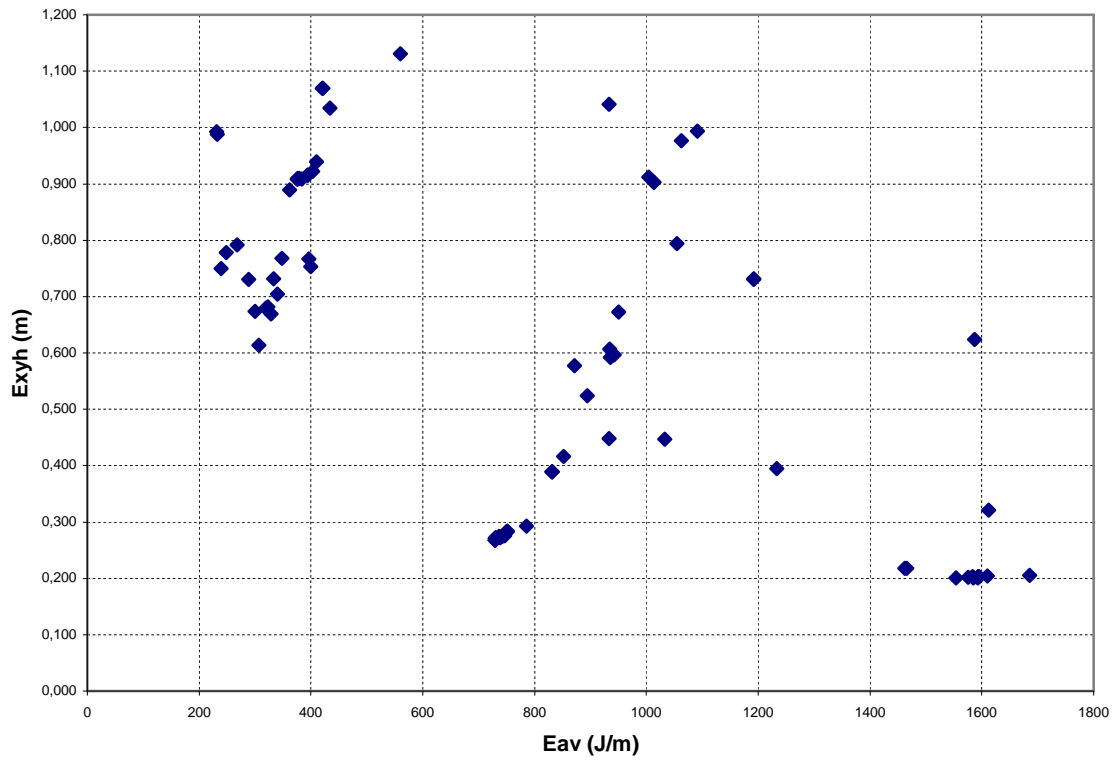


Figura 95 Soluções obtidas pelo AG – PLFICCI

É apresentado, na Figura 96, o resultado da simulação do robô, quando este apresenta os valores óptimos obtidos pelo AG (apresentados na Tabela 4).

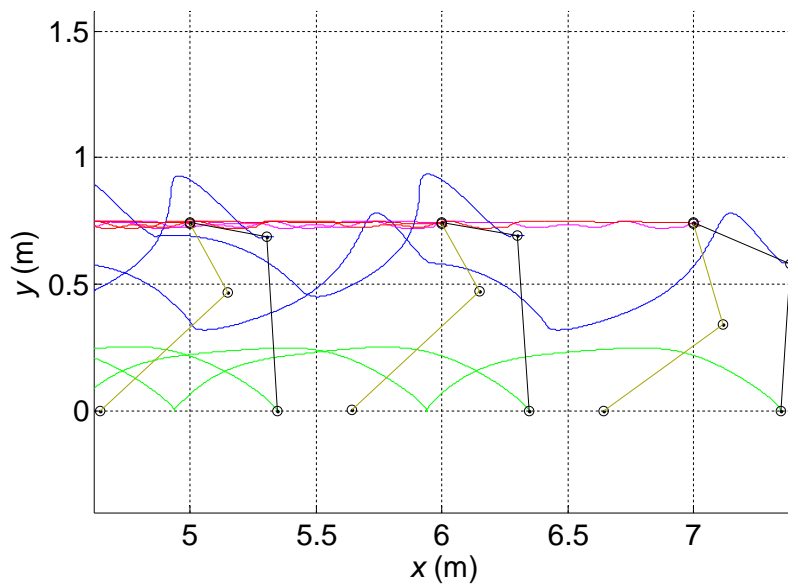


Figura 96 Simulação do sistema de locomoção – PLFICCI

8.2.7. RESUMO DOS VALORES OBTIDOS PELO AG

Nesta subsecção apresentam-se os valores obtidos pelo AG para cada um dos padrões de locomoção testados. Na Tabela 5 apresenta-se o valor da densidade de energia absoluta média por distância percorrida (E_{av}) obtida pelo AG e os correspondentes valores do erro da trajectória das ancas (ε_{xyH}) e da distância percorrida pelo robô, quando o objectivo era a minimização do índice E_{av} .

De referir que estes valores são próximos dos valores óptimos, uma vez que o critério de paragem do AG foi o número de gerações criadas. Neste caso o número de gerações criadas foi de 200 e cada geração apresentava 20 indivíduos. Tal opção foi tomada porque o esforço computacional era considerável e verificou-se que seriam necessários vários dias para ser atingido o valor óptimo.

Tabela 5 Valores mínimos do índice E_{av} , para cada padrão de locomoção, e correspondentes valores de ε_{xyH} e da distância percorrida pelo robô

	E_{av}	ε_{xyH}	Distância
PLO	211,987 J/m	0,521 m	1,043 m
PLFIMC	215,022 J/m	1,410 m	0,760 m
PLFICC	286,258 J/m	0,365 m	0,966 m
PLOI	238,540 J/m	1,144 m	1,138 m
PLFIMCI	310,166 J/m	0,680 m	0,797 m
PLFICCI	245,018 J/m	0,325 m	0,951 m

Na Tabela 6 apresenta-se o valor do erro mínimo da trajectória das ancas (ε_{xyH}) obtido pelo AG e os correspondentes valores da densidade de energia absoluta média por distância percorrida (E_{av}) e da distância percorrida pelo robô, para o caso em que a função objectivo do AG era a minimização do índice ε_{xyH} .

De referir, mais uma vez, que os valores obtidos, são aproximados dos valores óptimos, uma vez que também neste caso, o critério de paragem do AG foi o número de gerações

criadas (também aqui o número de gerações criadas foi de 200 e cada geração apresentava 20 indivíduos).

Tabela 6 Valores mínimos do índice ε_{xyH} para cada padrão de locomoção, e correspondentes valores de E_{av} e da distância percorrida pelo robô

	E_{av}	ε_{xyH}	Distância
PLO	4787,057 J/m	0,191 m	0,935 m
PLFIMC	1465,565 J/m	0,217 m	1,256 m
PLFICC	1476,885 J/m	0,204 m	1,255 m
PLOI	1018,311 J/m	0,165 m	1,162 m
PLFIMCI	1562,567 J/m	0,200 m	1,214 m
PLFICCI	1593,135 J/m	0,200 m	1,214 m

Para concluir esta análise, indicam-se na Tabela 7 os valores obtidos para os índices E_{av} e ε_{xyH} e para a distância percorrida pelo robô quando este apresenta os parâmetros otimizados para cada um dos padrões de locomoção.

Mais uma vez, os valores obtidos são aproximados dos valores ótimos uma vez que, tal como nas experiências anteriores, o critério de paragem do AG foi o número de gerações criadas. Também aqui o número de gerações criadas foi de 200, apresentando cada geração 20 indivíduos.

Tabela 7 Resumo dos valores ótimos para cada padrão de locomoção

	E_{av}	ε_{xyH}	Distância
PLO	334,135 Jm ⁻¹	0,344 m	0,789 m
PLFIMC	566,696 Jm ⁻¹	0,419 m	0,990 m
PLFICC	572,301 Jm ⁻¹	0,429 m	0,990m
PLOI	476,209 Jm ⁻¹	0,236 m	1,241 m
PLFIMCI	750,408 Jm ⁻¹	0,249 m	1,346 m
PLFICCI	752,677 Jm ⁻¹	0,250 m	1,406 m

Da análise dos dados indicados nesta tabela, podem-se extrair as seguintes conclusões:

- Os padrões de locomoção directos, de fase igual e meio ciclo e de fase igual e ciclo completo apresentam valores muito aproximados: quer os valores da energia absoluta média por distância percorrida (E_{av}), quer o erro de seguimento da trajectória das ancas (ε_{xyH}) são muito próximos e o valor da distância percorrida pelo robô é igual.
- Os padrões de locomoção de fase igual e meio ciclo inverso e o de fase igual e ciclo completo inverso também apresentam valores igualmente aproximados.

8.3. INFLUÊNCIA DA VELOCIDADE NOS PARÂMETROS DO ROBÔ

No conjunto de experiências realizadas anteriormente, assumiu-se que a velocidade de locomoção do robô era fixa e igual a 1 m/s. Nesta secção analisa-se a influência deste parâmetro (V_F) nos resultados obtidos (ou seja, nos parâmetros do robô que se pretende otimizar), pelo que se executa o AG para os valores de V_F no conjunto {0,1 m/s; 0,5 m/s; 1 m/s; 2 m/s; 3 m/s; 4 m/s; 5 m/s; 7 m/s; 10 m/s}.

8.3.1. INFLUÊNCIA DA VELOCIDADE NO COMPRIMENTO DO PASSO DO ROBÔ

Como se pode ver na Figura 97, o valor do comprimento do passo (L_s) aumenta com a velocidade de deslocamento do robô. Para baixas velocidades o valor do comprimento do passo é igualmente baixo e à medida que a velocidade vai aumentando, o valor do comprimento do passo aumenta também e, neste caso, chega a atingir o valor de 1,5 m.

De referir também que para velocidades superiores a 5 m/s a variação de L_s começa a apresentar um comportamento estranho. A informação que consta no gráfico não parece fiável, portanto não deve ser tirada nenhuma conclusão para esta gama de valores da velocidade.

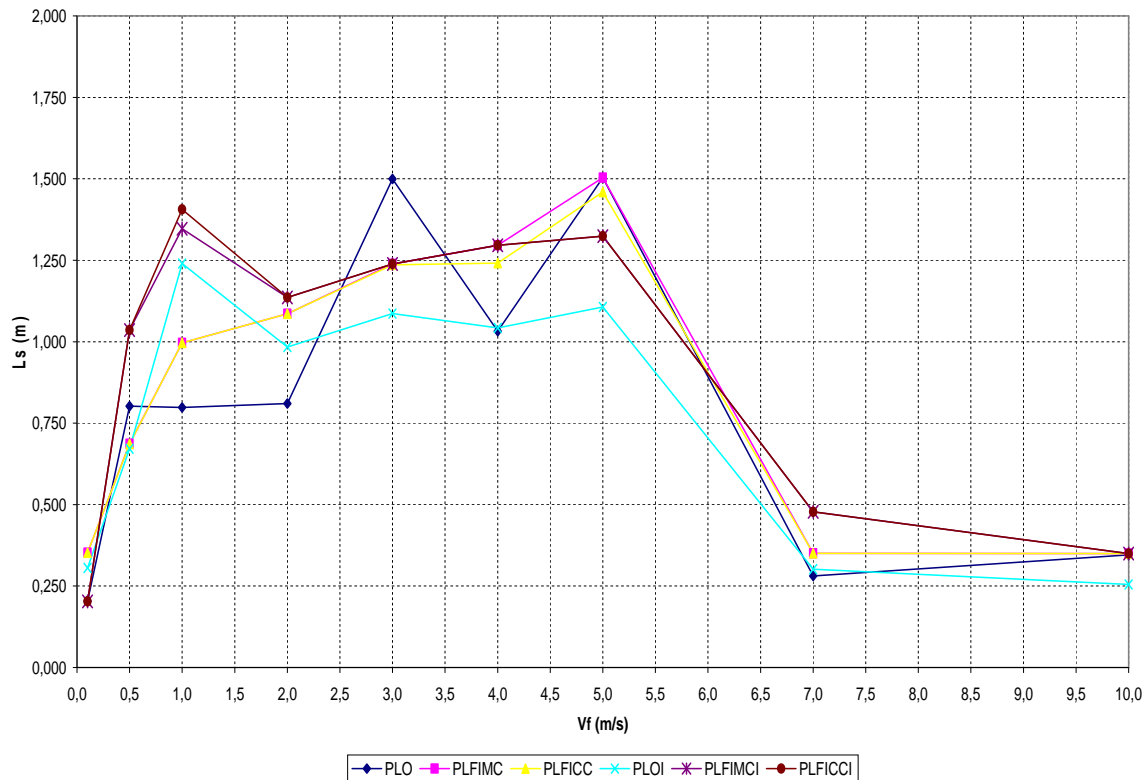


Figura 97 Comprimento do passo (L_s) em função da velocidade

8.3.2. INFLUÊNCIA DA VELOCIDADE DO ROBÔ NO FACTOR DE OCUPAÇÃO

Como se pode ver na Figura 98, o valor do factor de ocupação (β) diminui à medida que a velocidade de deslocamento do robô aumenta. Para baixas velocidades o valor é superior a 50%, para a maior parte dos padrões de locomoção, mas vai baixando à medida que a velocidade aumenta e para velocidades mais elevadas chega-se a atingir um valor de 20%.

De referir novamente que, para velocidades superiores a 5 m/s, os resultados obtidos começam a ter um comportamento pouco confiável, como se pode ver no gráfico.

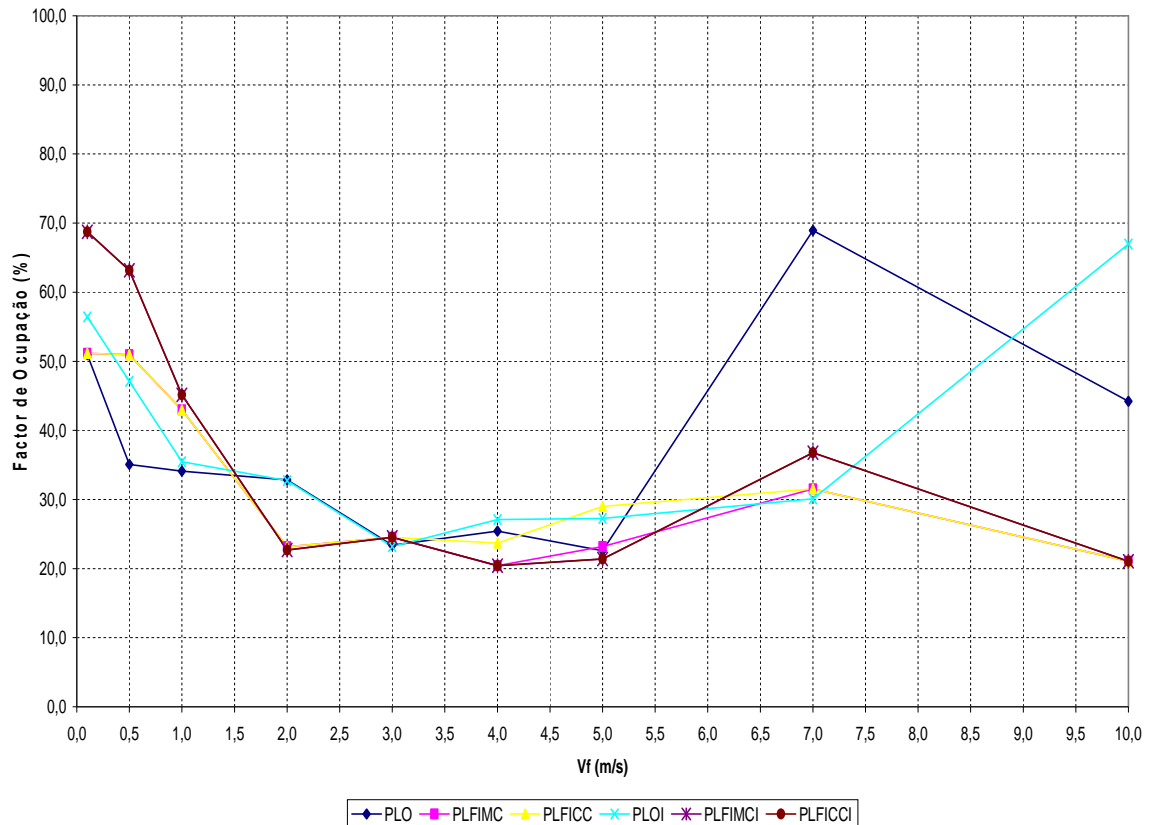


Figura 98 Factor de ocupação (β) em função da velocidade

8.3.3. INFLUÊNCIA DA VELOCIDADE NA ALTURA DO CORPO DO ROBÔ AO SOLO

Quanto ao valor da altura do corpo ao solo (Hb), este aumenta ligeiramente à medida que a velocidade de deslocamento do robô também aumenta, como é possível ver na Figura 99. Até à velocidade de 2,0 m/s o valor ronda os 0,70 m e para velocidades mais elevadas o valor de Hb chega a atingir 0,85 m.

Ao contrário dos dois parâmetros analisados anteriormente que apresentaram uma evolução de acordo com o esperado, o comportamento deste parâmetro foi uma surpresa, uma vez que era esperado que à medida que a velocidade aumentasse a altura do corpo ao solo diminuísse.

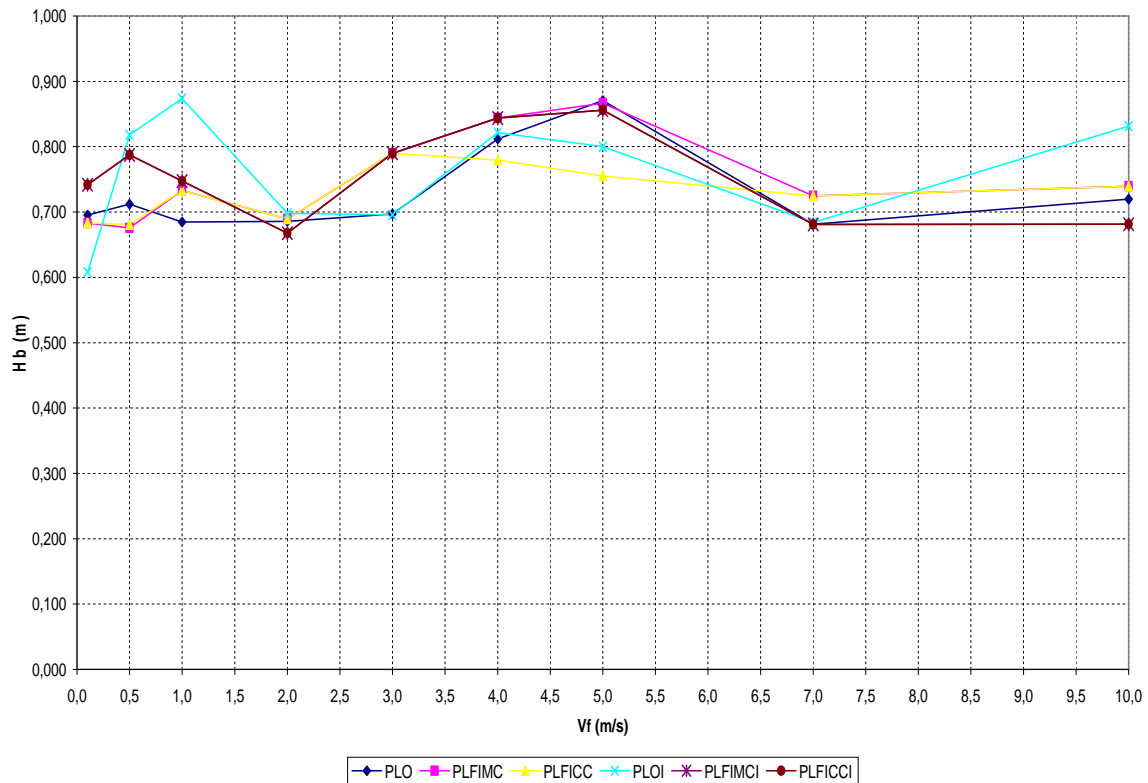


Figura 99 Altura do corpo ao solo (H_b) em função da velocidade

8.3.4. INFLUÊNCIA DA VELOCIDADE NA ALTURA DO PÉ DO ROBÔ AO SOLO

Na Figura 100 apresenta-se a variação do valor da altura do pé ao solo com a velocidade de locomoção do robô. Da análise deste gráfico, pode-se verificar que o valor de F_c não apresenta uma relação directa com V_F , uma vez que os gráficos dos diversos padrões de locomoção são bastante “oscilantes”. Por exemplo, para a velocidade de 1 m/s a altura do pé ronda os 0,2 m, para a velocidade de 2 m/s esse valor baixa para menos de 0,1 m mas para a velocidade de 3 m/s o valor volta a subir para uma altura de 0,15 m.

A única conclusão que é possível extrair da análise deste gráfico é que F_c deve apresentar um valor relativamente reduzido.

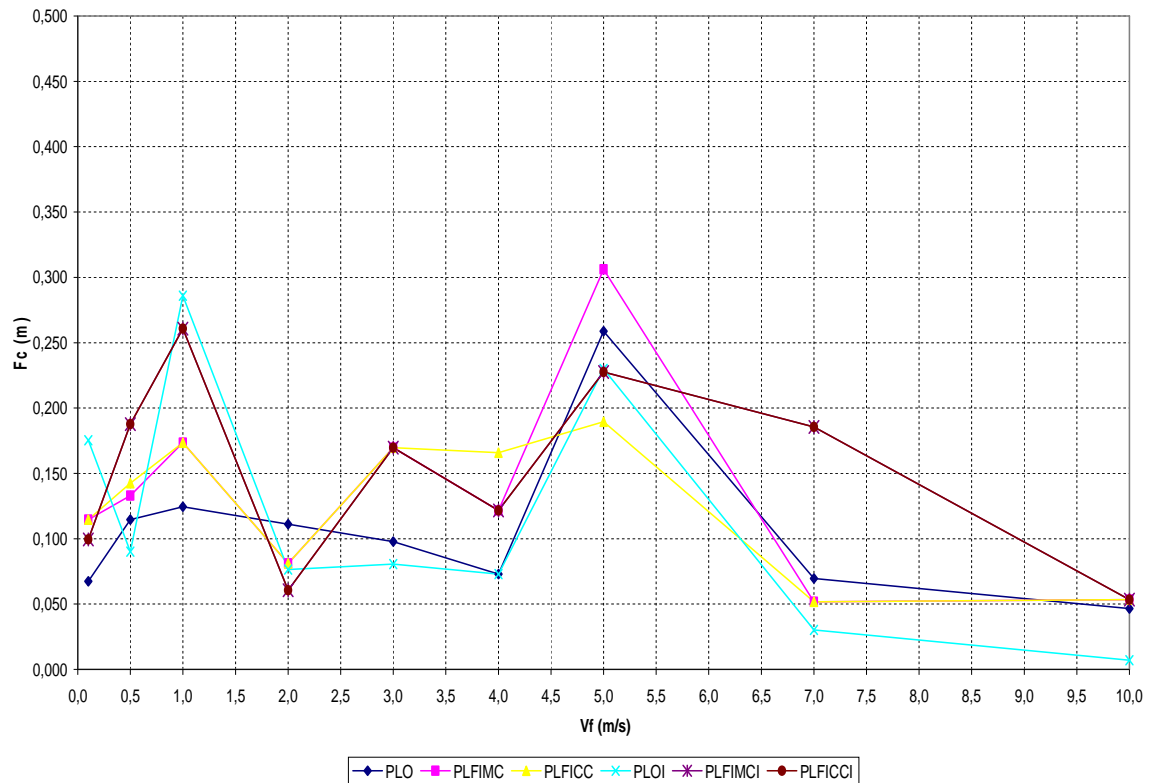


Figura 100 Altura do pé ao solo (F_c) em função da velocidade

8.3.5. INFLUÊNCIA DA VELOCIDADE NA MASSA DO CORPO

Como se pode ver da análise da Figura 101, o valor da massa do corpo não apresenta um comportamento linear com o aumento da velocidade. Assim, para velocidades até 0,5 m/s o valor ronda os 50 kg, para a velocidade de 1 m/s esse valor aumenta para os 70 kg e, de seguida, diminui até serem atingidos os 40 kg a uma velocidade de 4 m/s. Para velocidades superiores, o sistema robótico começa a ter um comportamento estranho, pelo que a informação que consta no gráfico não parece ser fiável.

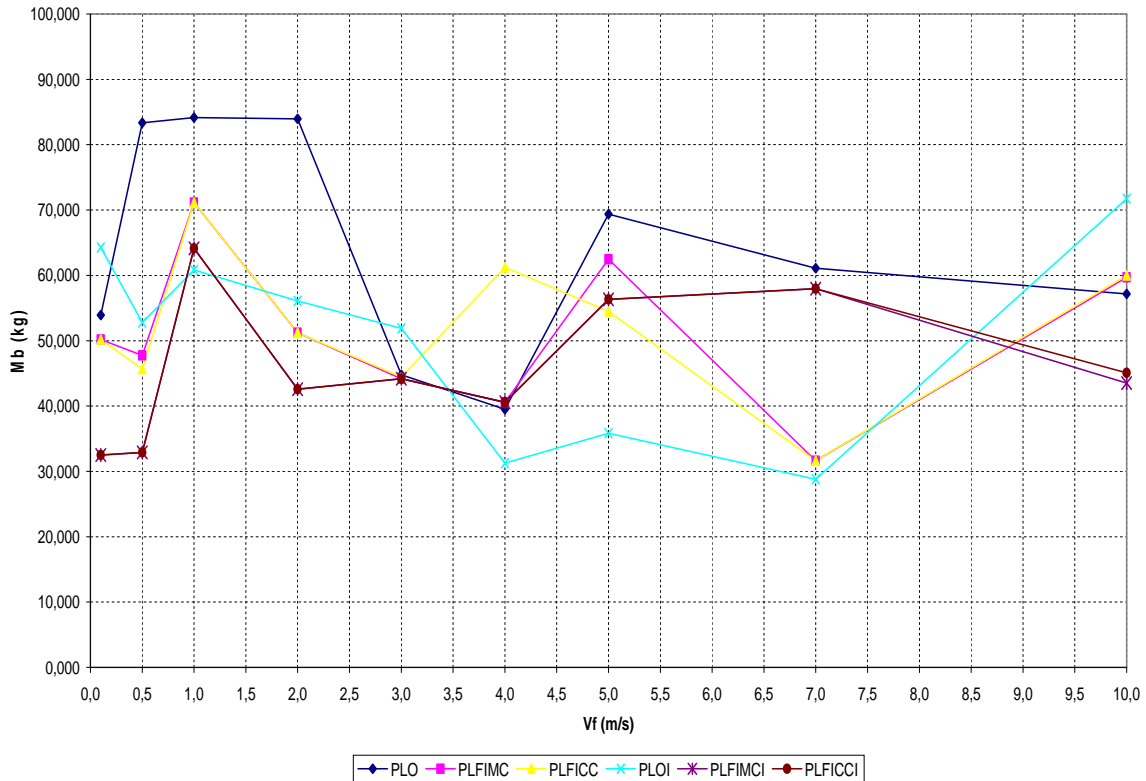


Figura 101 Massa do corpo (M_b) em função da velocidade

8.3.6. INFLUÊNCIA DA VELOCIDADE NA MASSA DOS ELOS DAS PERNAS DIANTEIRAS

Como é possível ver analisando as Figura 102 e Figura 103, o valor da massa dos elos das pernas dianteiras (M_{1j}) não tem um comportamento uniforme ao longo da gama de velocidades estudada. Desta forma não é possível tirar nenhuma conclusão relativamente à influência da velocidade nestes dois parâmetros do robô.

De referir também que o valor da massa referido nos gráficos das Figura 102 e Figura 103 é a soma das massas do primeiro elo das duas pernas dianteiras do robô, ou seja, a massa do primeiro elo de cada perna é metade desse valor.

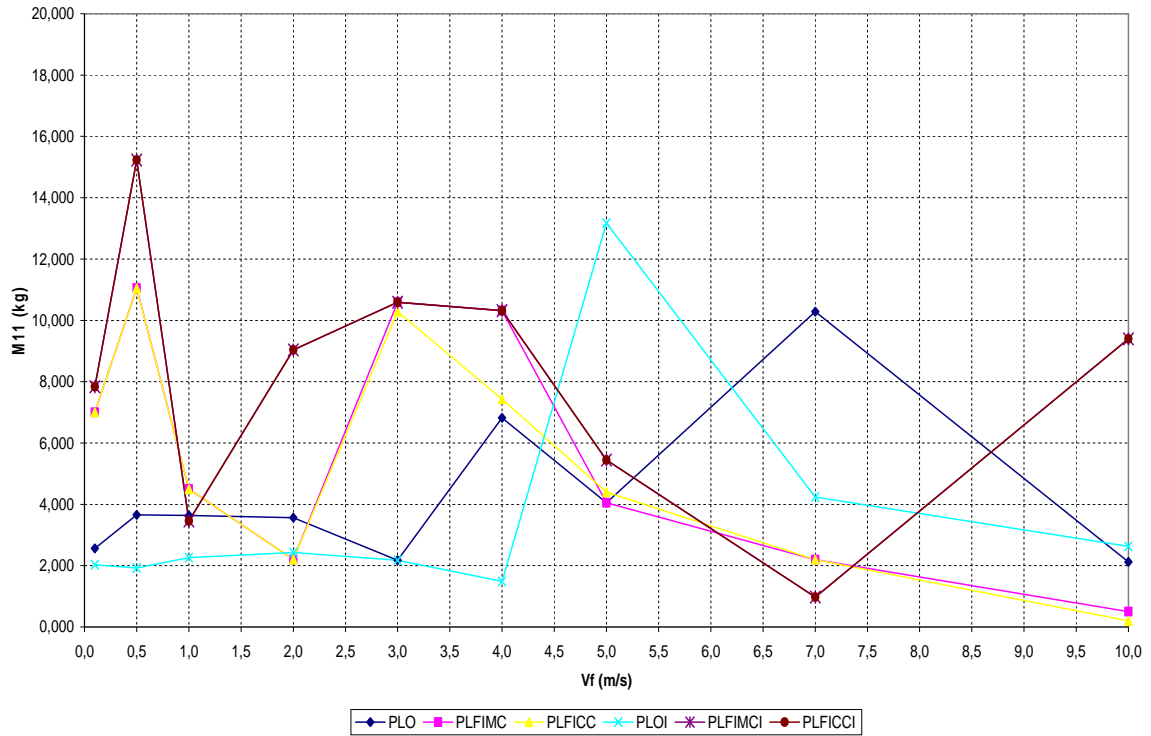


Figura 102 Massa do primeiro elo das pernas dianteiras (M_{11}) em função da velocidade

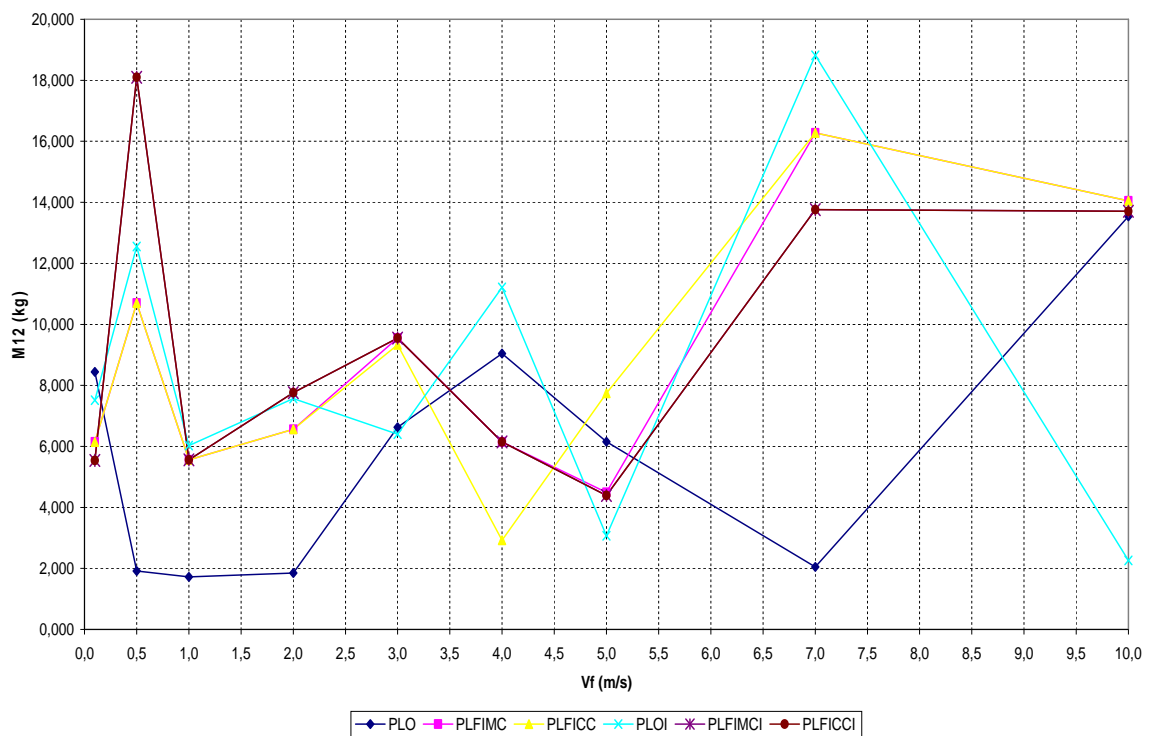


Figura 103 Massa do segundo elo das pernas dianteiras (M_{12}) em função da velocidade

8.3.7. INFLUÊNCIA DA VELOCIDADE NO COMPRIMENTO DOS ELOS DAS PERNAS DIANTEIRAS

Na Figura 104 apresenta-se a evolução óptima do comprimento do primeiro elo das pernas dianteiras do robô (L_{1j}) à medida que a velocidade varia. Para velocidades reduzidas o comprimento do primeiro elo ronda os 0,5 m e, à medida que o valor da velocidade aumenta, esse valor vai diminuindo ligeiramente até atingir os 0,3 m. Comportamento contrário apresenta o comprimento do segundo elo das pernas dianteiras do robô (ver Figura 105), uma vez que o comprimento total das pernas do robô é um valor fixo (1 m), logo um primeiro elo mais longo implica um segundo elo mais curto e vice-versa.

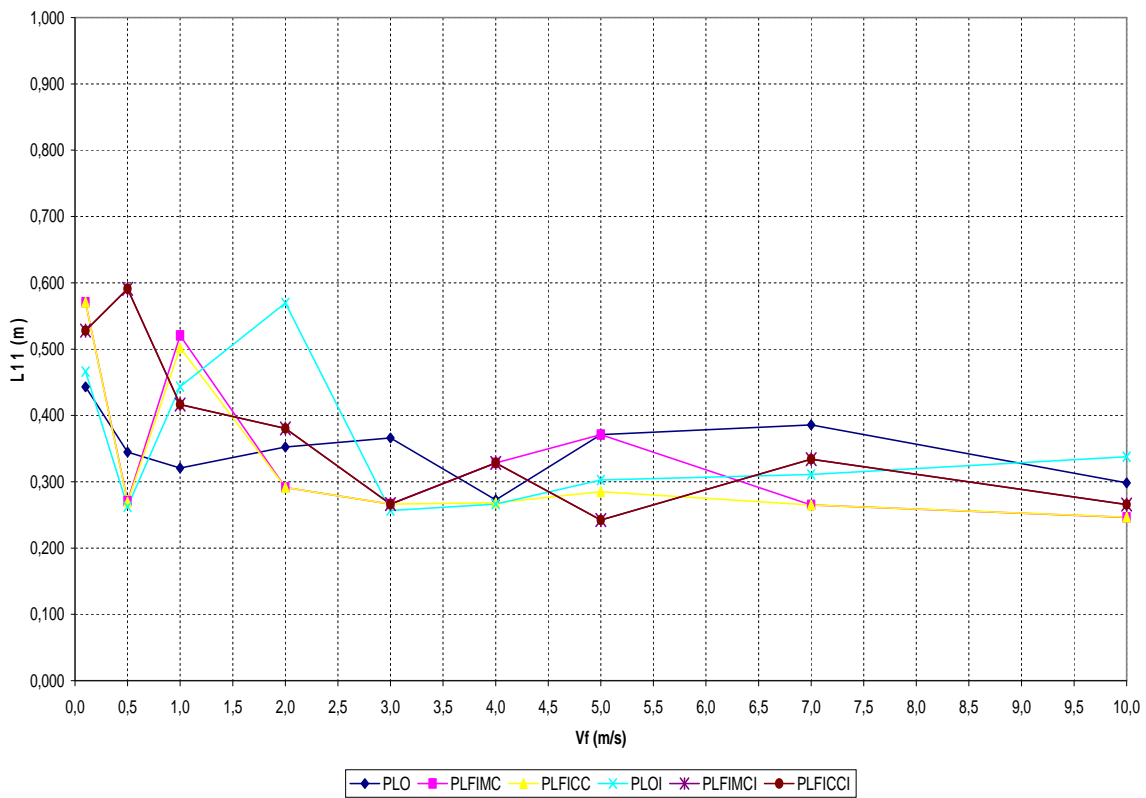


Figura 104 Comprimento do primeiro elo das pernas dianteiras (L_{11}) em função da velocidade

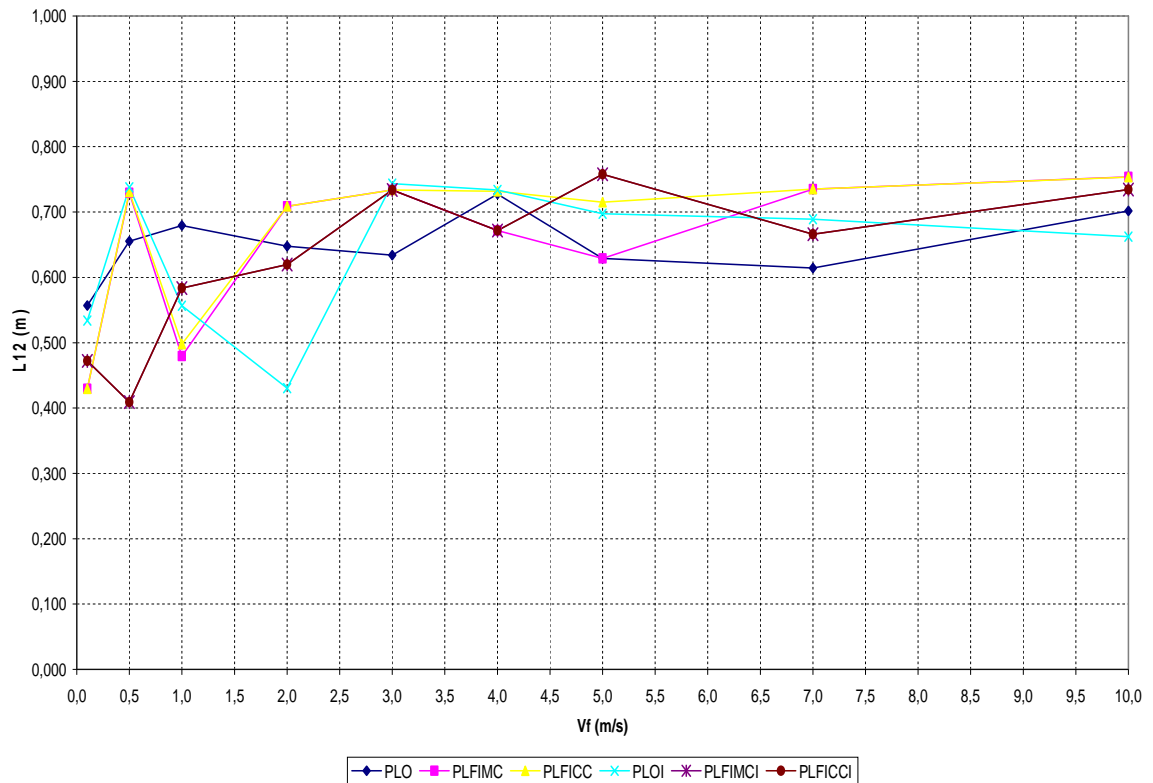


Figura 105 Comprimento do segundo elo das pernas dianteiras (L_{12}) em função da velocidade

8.3.8. INFLUÊNCIA DA VELOCIDADE NA MASSA DOS ELOS DAS PERNAS DO MEIO

Através da análise das Figura 106 e Figura 107, constata-se que não parece haver nenhuma influência directa da velocidade no valor da massa dos elos das pernas do meio do robô (M_{2j}), ou seja, o valor da massa dos elos destas pernas apresenta um comportamento “oscilante” ao longo da gama de valores da velocidade testada, não apresentando nenhuma tendência visível.

Novamente, para velocidades superiores a 5 m/s, os resultados obtidos começam a ter um comportamento pouco confiável, como se pode ver em ambos os gráficos.

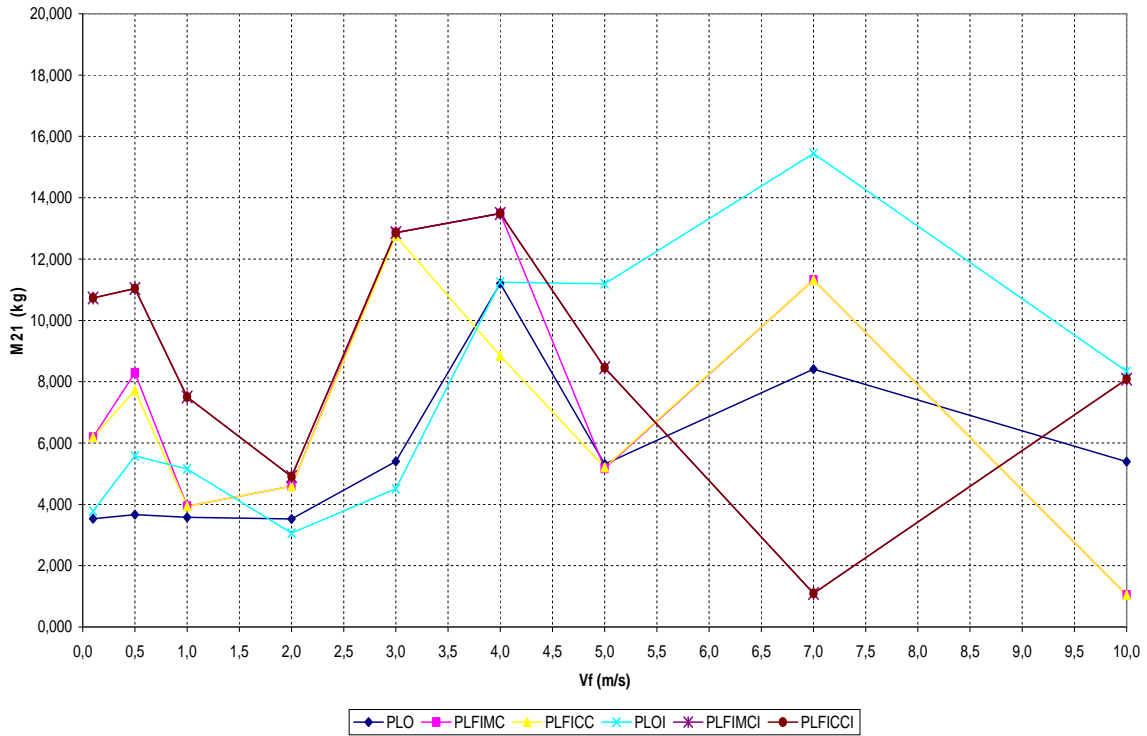


Figura 106 Massa do primeiro elo das pernas do meio (M_{21}) em função da velocidade

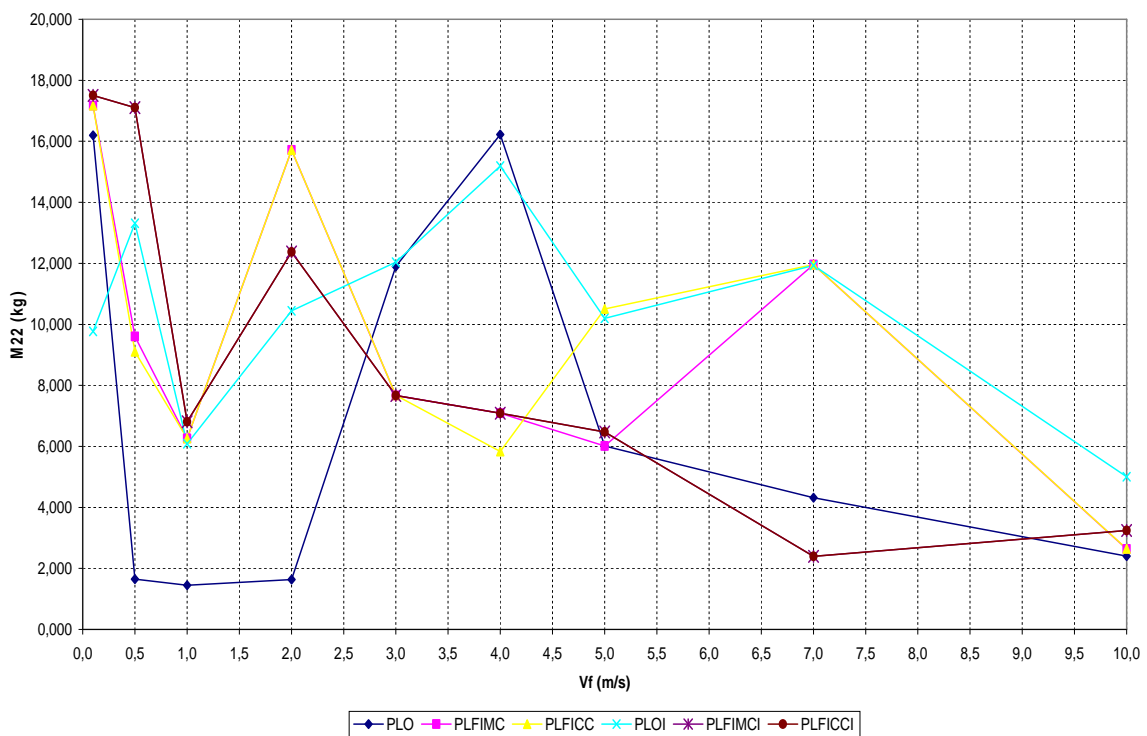


Figura 107 Massa do segundo elo das pernas do meio (M_{22}) em função da velocidade

8.3.9. INFLUÊNCIA DA VELOCIDADE NO COMPRIMENTO DOS ELOS DAS PERNAS DO MEIO

Como se pode ver na Figura 108 o comprimento do primeiro elo das pernas do meio apresenta uma clara variação para velocidades inferiores a 1 m/s. No entanto, para valores de velocidade superiores o comprimento do primeiro elo estabiliza nos 0,35 m. O segundo elo das pernas do meio do robô (Figura 109) também apresenta, igualmente, uma variação do seu comprimento, complementar da descrita, para valores inferiores a 1 m/s e para valores superiores de velocidade o seu comprimento estabiliza nos 0,65 m.

De referir novamente que, para valores da velocidade superiores a 5 m/s os resultados não apresentam um comportamento estável, logo os valores obtidos pelo AG não parecem fiáveis.

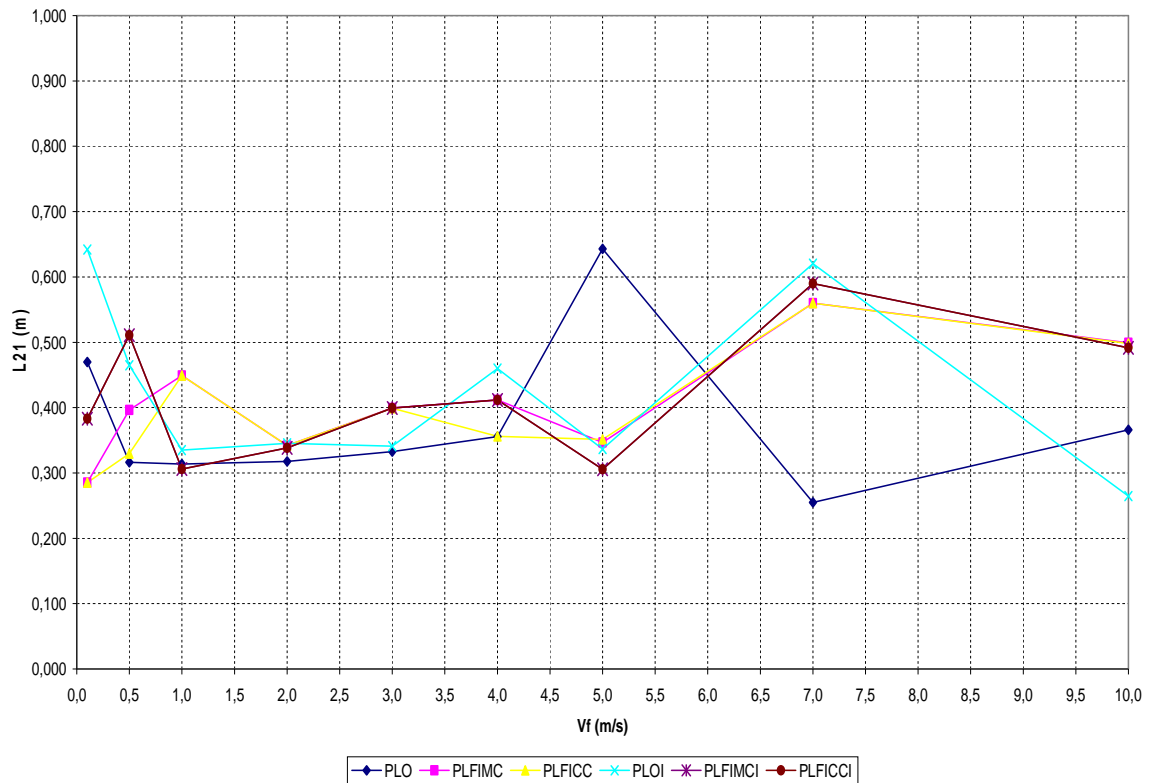


Figura 108 Comprimento do primeiro elo das pernas do meio (L_{21}) em função da velocidade

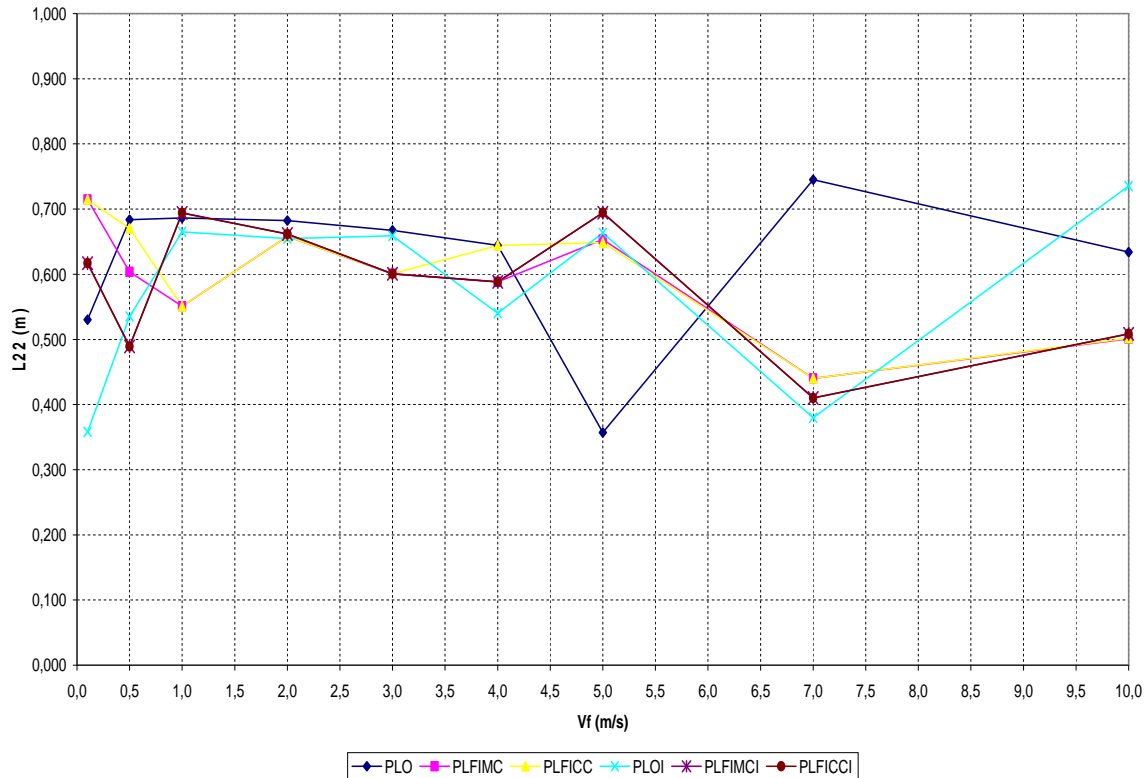


Figura 109 Comprimento do segundo elo das pernas do meio (L_{22}) em função da velocidade

8.3.10. INFLUÊNCIA DA VELOCIDADE NA MASSA DOS ELOS DAS PERNAS TRASEIRAS

Através da observação do gráfico da Figura 110 não é possível tirar conclusões sobre a influência da velocidade na massa do primeiro elo das pernas traseiras, uma vez que o gráfico apresenta um comportamento "oscilatório". Parece haver um aumento deste valor com o aumento do valor da velocidade, mas tal não é muito nítido.

De igual modo, não é possível tirar uma conclusão definitiva sobre a influência da velocidade na massa do segundo elo das pernas traseiras do robô (Figura 111). Parece existir um decréscimo do valor da massa do segundo elo com o aumento do valor da velocidade, mas não é possível afirmá-lo categoricamente.

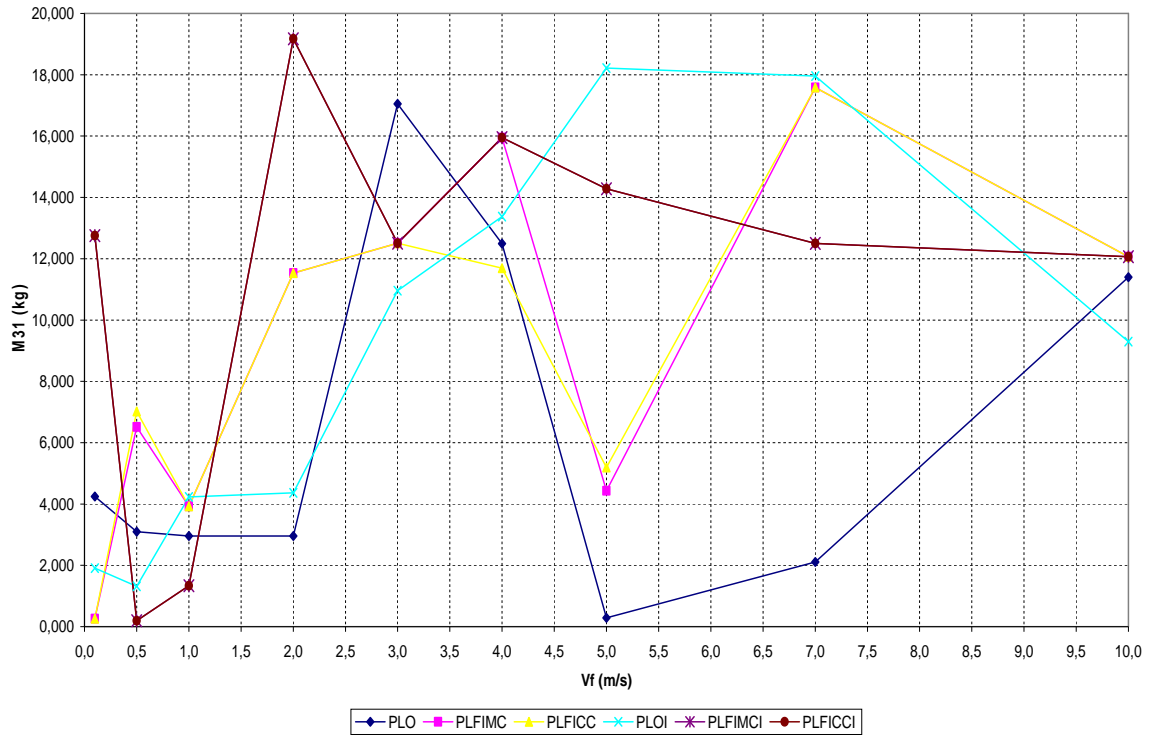


Figura 110 Massa do primeiro elo das pernas traseiras (M_{31}) em função da velocidade

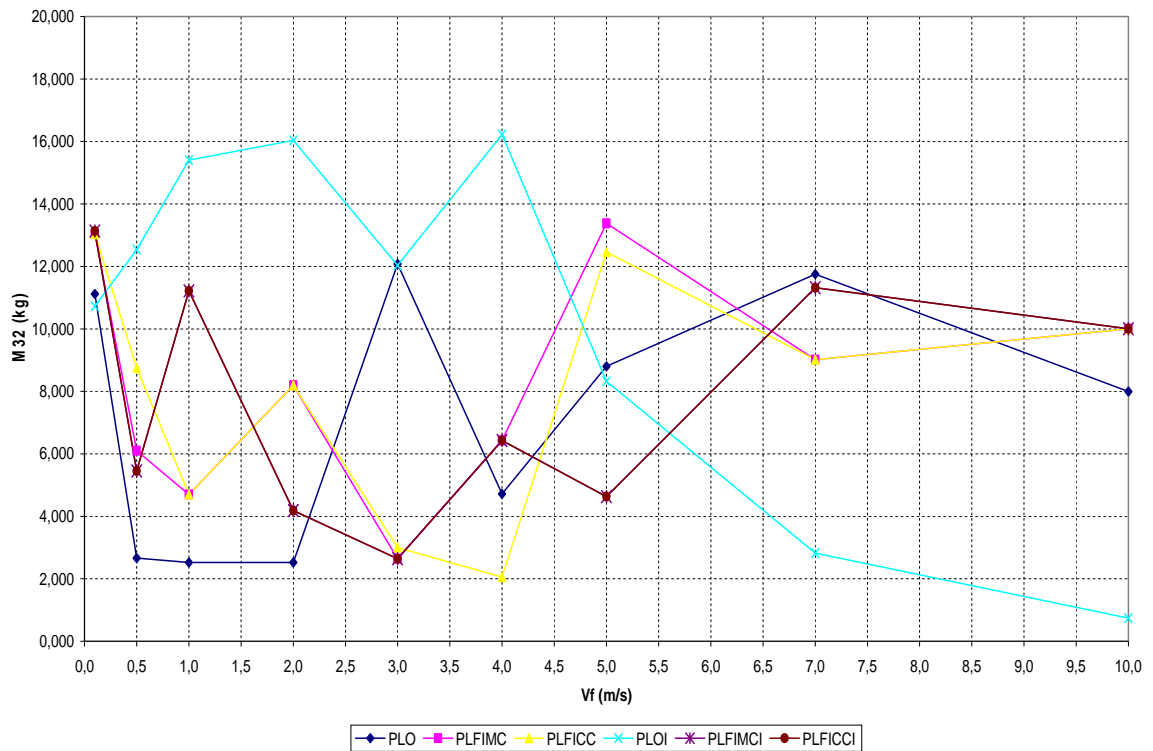


Figura 111 Massa do segundo elo das pernas traseiras (M_{32}) em função da velocidade

8.3.11. INFLUÊNCIA DA VELOCIDADE NO COMPRIMENTO DOS ELOS DAS PERNAS TRASEIRAS

Como se pode concluir analisando a Figura 112, o comprimento do primeiro elo das pernas traseiras apresenta um valor próximo de 0,4 m, para velocidades reduzidas, diminuindo ligeiramente até atingir o valor de 0,2 m para uma velocidade de 4 m/s. Comportamento contrário apresenta o comprimento do segundo elo das pernas traseiras (Figura 113). Este começa com um valor próximo de 0,6 m e depois vai aumentando ligeiramente até atingir o valor de 0,8 m para a velocidade de 4 m/s.

Mais uma vez, convém referir que para valores da velocidade superiores a 5 m/s os valores obtidos deixam de ser fiáveis.

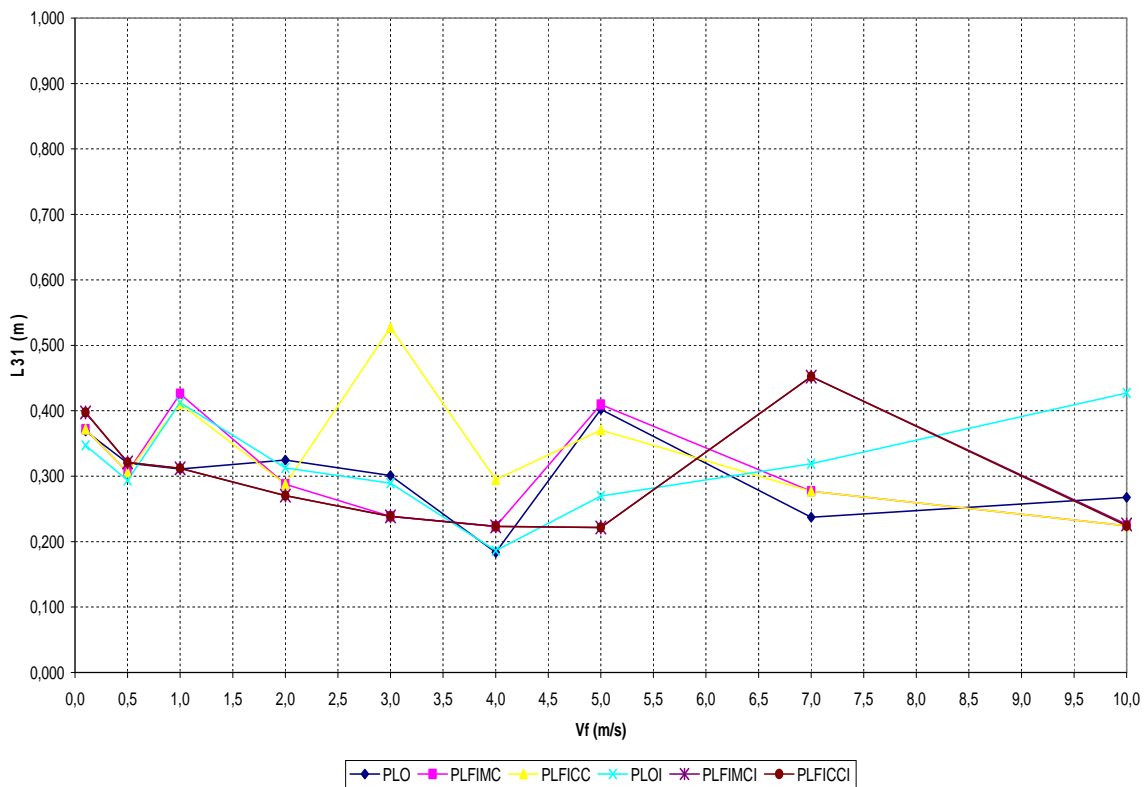


Figura 112 Comprimento do primeiro elo das pernas traseiras (L_{31}) em função da velocidade

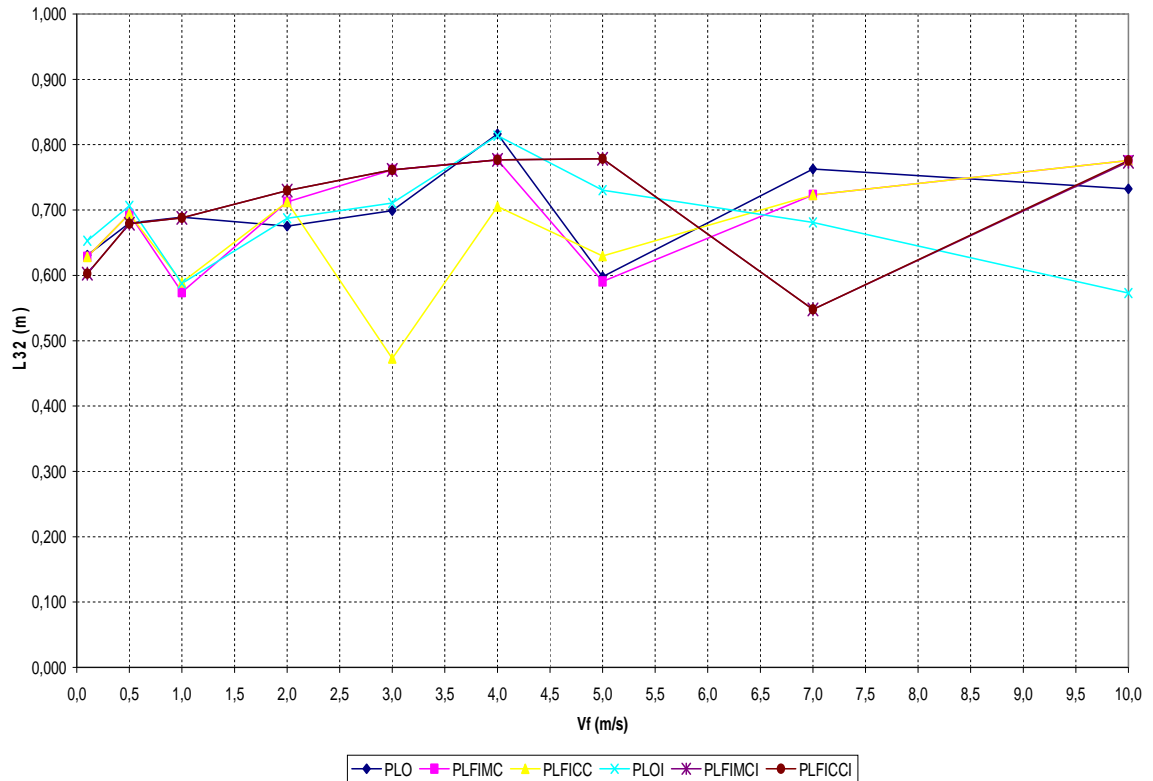


Figura 113 Comprimento do segundo elo das pernas traseiras (L_{32}) em função da velocidade

8.3.12. INFLUÊNCIA DA VELOCIDADE NO *OFFSET* DOS PÉS EM RELAÇÃO ÀS ANCAS

Como se pode observar nas Figura 114-Figura 116, também não é possível tirar conclusões relativamente à influência da velocidade no *offset* dos diversos pés do robô em relação às respectivas ancas, uma vez que os gráficos apresentam um comportamento muito "instável".

A única conclusão que é possível tirar é que o *offset* das pernas do meio do robô (Figura 115) apresenta um valor negativo para a maior parte dos padrões de locomoção e para a grande maioria dos valores de velocidade.

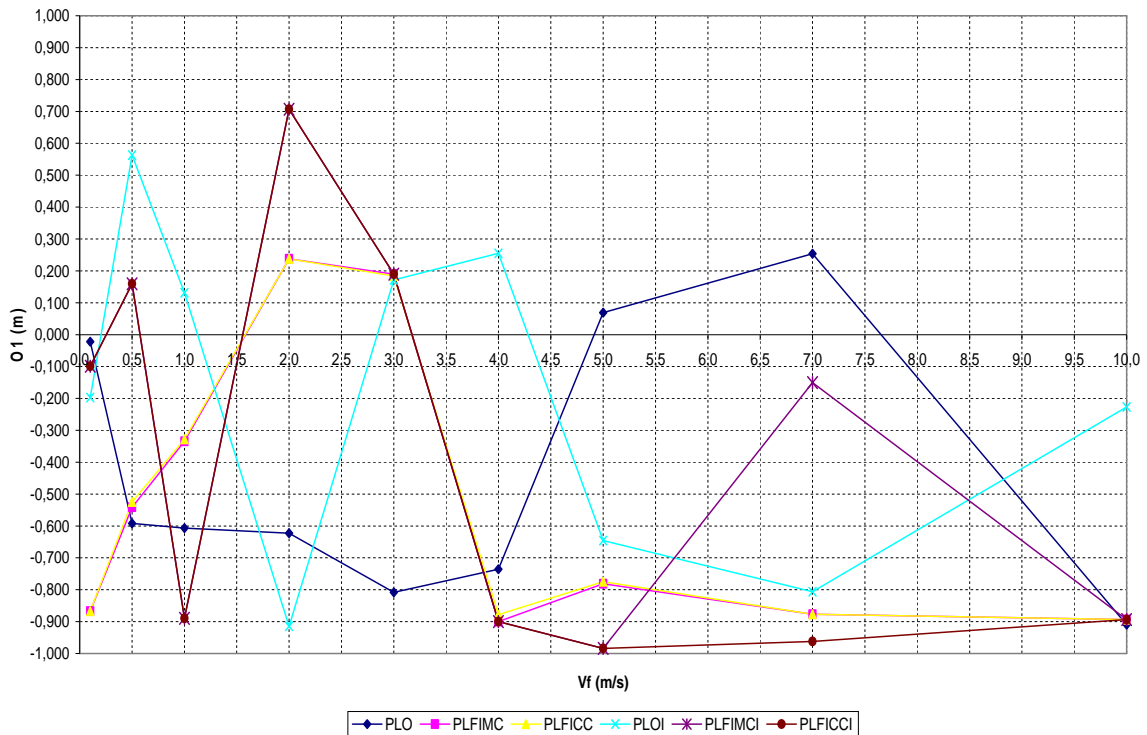


Figura 114 Offset dos pés dianteiros em relação às ancas (O_1) em função da velocidade

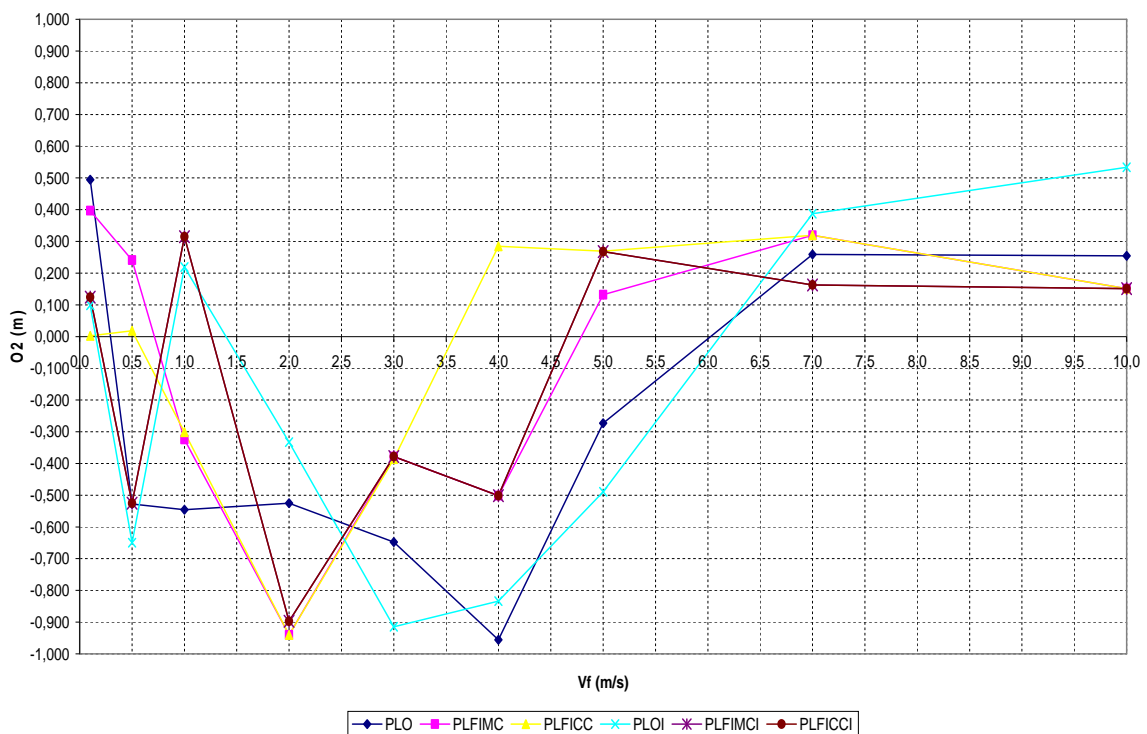


Figura 115 Offset dos pés do meio em relação às ancas (O_2) em função da velocidade

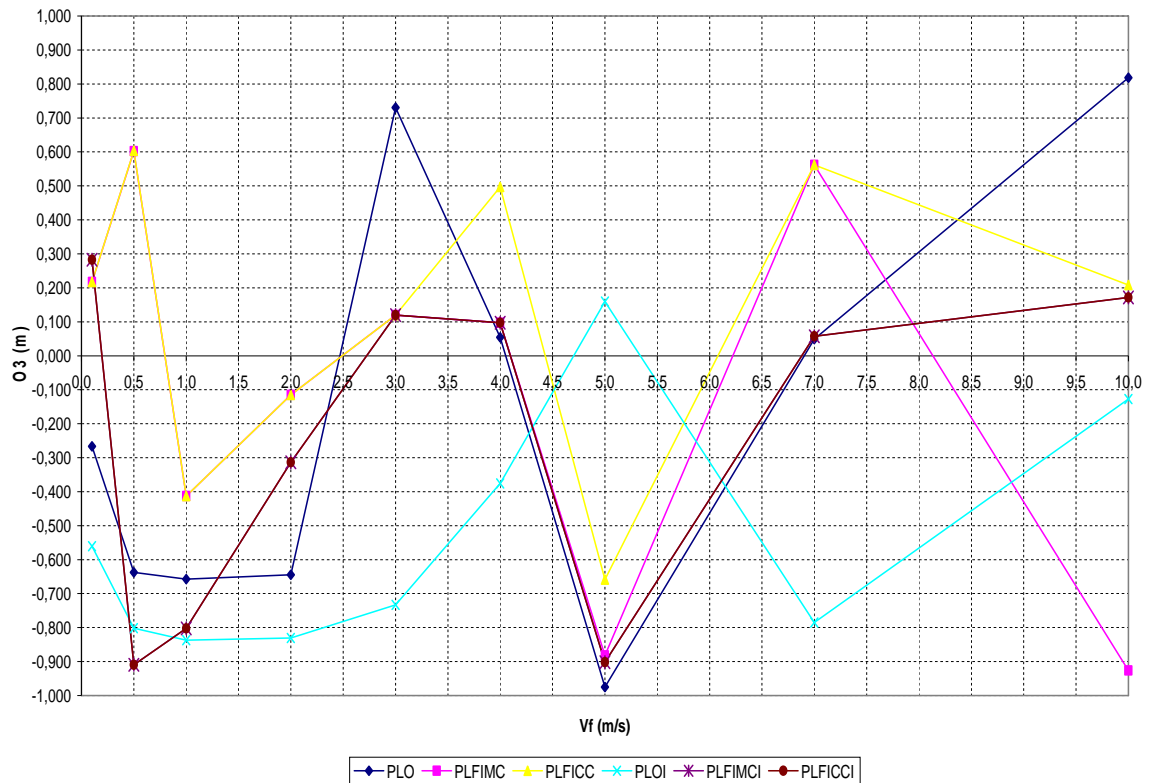


Figura 116 Offset dos pés traseiros em relação às ancas (O_3) em função da velocidade

8.3.13. INFLUÊNCIA DA VELOCIDADE NO VALOR DO ÍNDICE E_{av}

Como é possível observar na Figura 117, E_{av} apresenta valores menores para velocidades reduzidas (aproximadamente 500 J/m) e aumenta de valor com o aumento da velocidade de deslocamento do robô, atingindo valores superiores a 2000 J/m.

De salientar o facto de E_{av} diminuir para valores de velocidade superiores a 5 m/s (na maior parte dos padrões de locomoção), o que faz com esses valores não pareçam ser fiáveis.

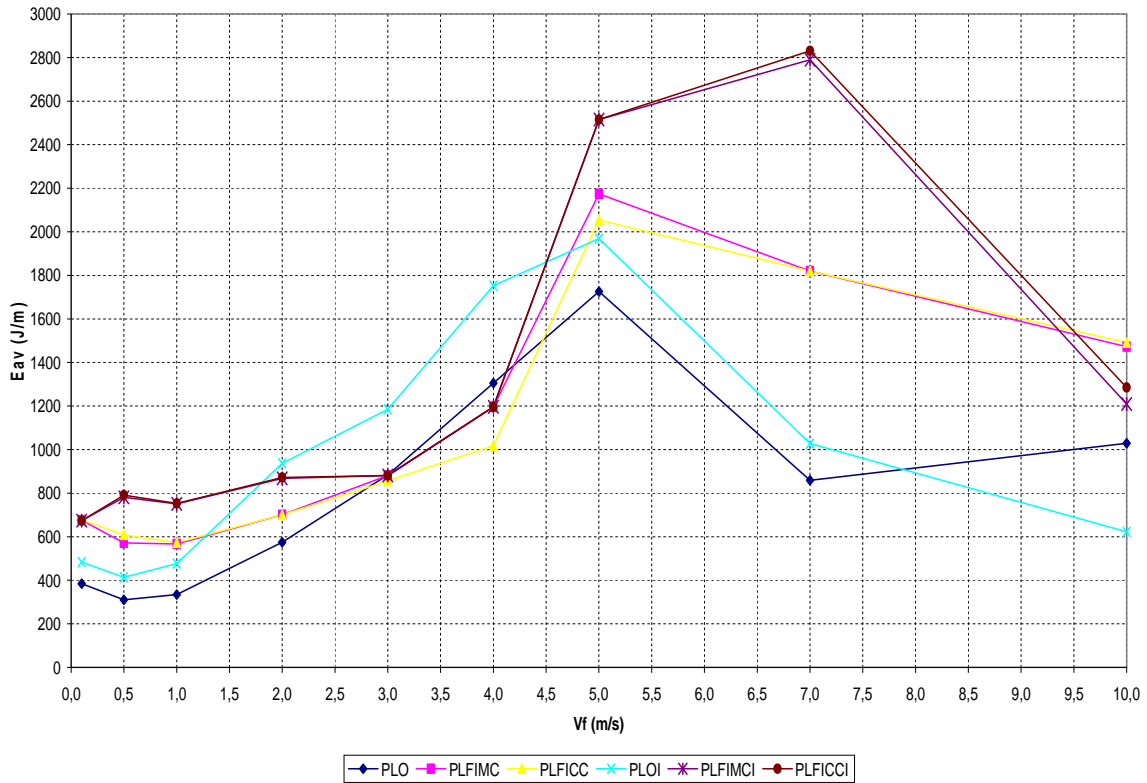


Figura 117 Valor do índice E_{av} em função da velocidade

8.3.14. INFLUÊNCIA DA VELOCIDADE NO VALOR DO ÍNDICE ϵ_{xyH}

Através da observação do gráfico da Figura 118, pode-se concluir que para baixas velocidades o valor do índice ϵ_{xyH} é menor (para uma velocidade de 0,1 m/s o valor de ϵ_{xyH} é aproximadamente 0,2 m). Tal como o índice analisado no ponto anterior, o valor de ϵ_{xyH} aumenta com o aumento do valor da velocidade de deslocamento do robô (para um valor da velocidade de 4 m/s, ϵ_{xyH} apresenta um valor de 0,4 m).

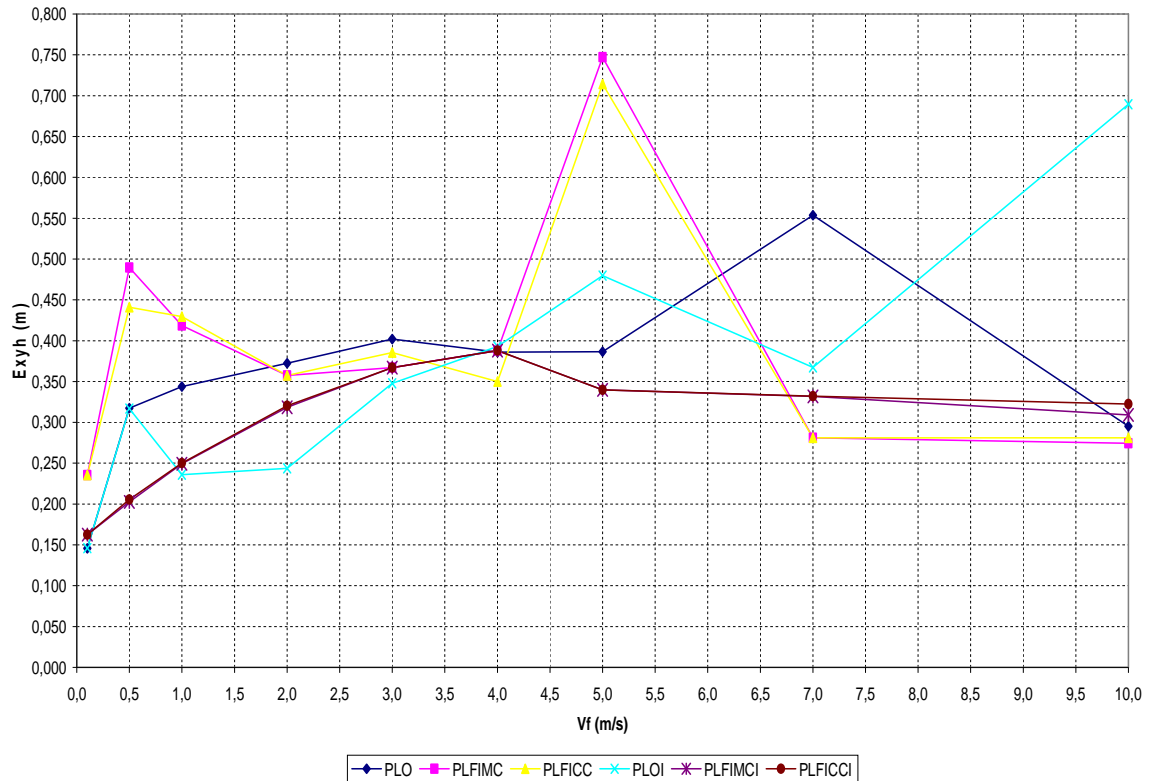


Figura 118 Valor do índice ϵ_{xyH} em função da velocidade

8.4. OPTIMIZAÇÃO DOS PARÂMETROS E DA VELOCIDADE DE DESLOCAMENTO DO ROBÔ

No terceiro, e último, conjunto de experiências, foi adicionado mais um gene (velocidade de deslocamento do robô) ao cromossoma já existente e foi executado novamente o AG, de maneira a serem obtidos os valores ótimos dos parâmetros do sistema robótico e também o valor ótimo da velocidade de deslocamento, para cada um dos padrões de locomoção.

Os valores ótimos dos índices E_{av} e ϵ_{xyH} e da velocidade de deslocamento são apresentados na Tabela 8. Relativamente aos parâmetros ótimos do robô hexápode, obtidos pelo AG, podem ser vistos na Tabela 9 (para os padrões de locomoção directos) e na Tabela 10 (para os padrões de locomoção inversos).

Tabela 8 Comparação dos valores óptimos dos índices E_{av} , ε_{xyH} e velocidade para cada padrão de locomoção

	E_{av}		ε_{xyH}		Distância		Velocidade
	48 Genes	49 Genes	48 Genes	49 Genes	48 Genes	49 Genes	49 Genes
PLO	334,1 J/m	684,0 J/m	0,344 m	0,180 m	0,789 m	1,091 m	0,998 m/s
PLFIMC	566,7 J/m	570,6 J/m	0,419 m	0,415 m	0,990 m	0,979 m	0,894 m/s
PLMICC	572,3 J/m	556,1 J/m	0,429 m	0,433 m	0,990 m	0,979 m	0,887 m/s
PLOI	476,2 J/m	318,7 J/m	0,236 m	0,297 m	1,241 m	0,967 m	0,823 m/s
PLFIMCI	750,4 J/m	570,6 J/m	0,249 m	0,415 m	1,346 m	0,979 m	0,894 m/s
PLFICCI	752,7 J/m	556,1 J/m	0,250 m	0,433 m	1,406 m	0,979 m	0,887 m/s

Como se pode concluir, as diferenças não são muito significativas face aos resultados das experiências anteriores porque, ao fixar-se o valor da velocidade em 1 m/s, já se estava bastante próximo do valor óptimo da velocidade de deslocamento do sistema robótico.

Tabela 9 Parâmetros e velocidade otimizados do robô (padrões de locomoção directos)

PLO	PLFIMC	PLFICC
Ls= 1,092 m Hb= 0,769 m Beta= 46,231 % Fc= 0,171 m L11= 0,429 m L12= 0,570 m L21= 0,481 m L22= 0,518 m L31= 0,439 m L32= 0,560 m O1= -0,663 m O2= -0,143 m O3= -0,132 m Mb= 70,012 kg M11= 4,432 kg M12= 5,400 kg M21= 4,832 kg M22= 5,243 kg M31= 3,679 kg M32= 6,398 kg Kxh= 98278,515 N/m Kyh= 10768,572 N/m Bxh= 819,719 N/m Byh= 100,164 N/m T11min= -194,630 Nm T11max= 229,909 Nm T12min= -213,356 Nm T12max= 282,904 Nm T21min= -217,231 Nm T21max= 255,773 Nm T22min= -113,630 Nm T22max= 212,406 Nm T31min= -188,314 Nm T31max= 244,742 Nm T32min= -191,005 Nm T32max= 246,835 Nm Kp11= 4909,057 Kd11= 435,714 Kp12= 3795,410 Kd12= 468,567 Kp21= 4942,468 Kd21= 408,138 Kp22= 3129,041 Kd22= 315,288 Kp31= 4184,318 Kd31= 440,307 Kp32= 3903,521 Kd32= 395,379 Velocidade= 0,998 m/s	Ls= 0,985 m Hb= 0,726 m Beta= 40,929 % Fc= 0,174 m L11= 0,618 m L12= 0,381 m L21= 0,294 m L22= 0,705 m L31= 0,500 m L32= 0,499 m O1= -0,420 m O2= -0,522 m O3= -0,754 m Mb= 71,100 kg M11= 3,565 kg M12= 11,764 kg M21= 3,694 kg M22= 3,221 kg M31= 2,897 kg M32= 3,755 kg Kxh= 82420,468 N/m Kyh= 9955,521 N/m Bxh= 857,208 N/m Byh= 108,981 N/m T11min= -185,116 Nm T11max= 160,055 Nm T12min= -157,307 Nm T12max= 187,401 Nm T21min= -277,207 Nm T21max= 174,589 Nm T22min= -213,118 Nm T22max= 277,730 Nm T31min= -227,397 Nm T31max= 168,550 Nm T32min= -132,206 Nm T32max= 239,986 Nm Kp11= 3145,000 Kd11= 256,088 Kp12= 2415,507 Kd12= 455,686 Kp21= 2708,788 Kd21= 273,206 Kp22= 2220,573 Kd22= 248,426 Kp31= 5865,435 Kd31= 260,684 Kp32= 2865,677 Kd32= 279,645 Velocidade= 0,894 m/s	Ls= 0,986 m Hb= 0,726 m Beta= 41,081 % Fc= 0,173 m L11= 0,617 m L12= 0,382 m L21= 0,294 m L22= 0,705 m L31= 0,495 m L32= 0,504 m O1= -0,417 m O2= -0,522 m O3= -0,746 m Mb= 71,389 kg M11= 3,566 kg M12= 11,512 kg M21= 3,728 kg M22= 3,221 kg M31= 2,897 kg M32= 3,684 kg Kxh= 82379,304 N/m Kyh= 9725,521 N/m Bxh= 858,405 N/m Byh= 108,804 N/m T11min= -175,096 Nm T11max= 96,416 Nm T12min= -181,288 Nm T12max= 201,025 Nm T21min= -275,444 Nm T21max= 175,724 Nm T22min= -211,579 Nm T22max= 129,014 Nm T31min= -228,025 Nm T31max= 200,180 Nm T32min= -132,306 Nm T32max= 240,113 Nm Kp11= 3144,999 Kd11= 250,009 Kp12= 2437,992 Kd12= 457,229 Kp21= 2611,638 Kd21= 272,864 Kp22= 2220,574 Kd22= 248,426 Kp31= 5952,859 Kd31= 247,737 Kp32= 2865,751 Kd32= 279,646 Velocidade= 0,887 m/s
E_{av} = 684,030 J/m	E_{av} = 570,561 J/m	E_{av} = 556,077 J/m
ϵ_{xyH} = 0,180 m	ϵ_{xyH} = 0,415 m	ϵ_{xyH} = 0,433 m
Distância= 1,091 m	Distância= 0,979 m	Distância= 0,979 m

Tabela 10 Parâmetros e velocidade otimizados do robô (padrões de locomoção inversos)

PLOI	PLFIMCI	PLFICCI
Ls= 0,970 m Hb= 0,727 m Beta= 37,392 % Fc= 0,182 m L11= 0,346 m L12= 0,653 m L21= 0,400 m L22= 0,599 m L31= 0,403 m L32= 0,596 m O1= -0,571 m O2= -0,443 m O3= -0,636 m Mb= 74,620 kg M11= 2,764 kg M12= 7,289 kg M21= 4,873 kg M22= 3,965 kg M31= 2,592 kg M32= 3,892 kg Kxh= 83880,281 N/m Kyh= 10238,614 N/m Bxh= 814,374 N/m Byh= 96,667 N/m T11min= -282,178 Nm T11max= 424,007 Nm T12min= -335,712 Nm T12max= 284,126 Nm T21min= -316,726 Nm T21max= 134,282 Nm T22min= -335,230 Nm T22max= 127,212 Nm T31min= -306,514 Nm T31max= 139,066 Nm T32min= -269,981 Nm T32max= 86,795 Nm Kp11= 2904,722 Kd11= 289,980 Kp12= 2537,508 Kd12= 154,991 Kp21= 3900,385 Kd21= 209,915 Kp22= 2241,516 Kd22= 231,071 Kp31= 2198,007 Kd31= 213,821 Kp32= 2675,184 Kd32= 239,164 Velocidade= 0,823 m/s	Ls= 0,985 m Hb= 0,726 m Beta= 40,929 % Fc= 0,174 m L11= 0,618 m L12= 0,381 m L21= 0,294 m L22= 0,705 m L31= 0,500 m L32= 0,499 m O1= -0,420 m O2= -0,522 m O3= -0,754 m Mb= 71,100 kg M11= 3,565 kg M12= 11,764 kg M21= 3,694 kg M22= 3,221 kg M31= 2,897 kg M32= 3,755 kg Kxh= 82420,468 N/m Kyh= 9955,521 N/m Bxh= 857,208 N/m Byh= 108,981 N/m T11min= -185,116 Nm T11max= 160,055 Nm T12min= -157,307 Nm T12max= 187,401 Nm T21min= -277,207 Nm T21max= 174,589 Nm T22min= -213,118 Nm T22max= 277,730 Nm T31min= -227,397 Nm T31max= 168,550 Nm T32min= -132,206 Nm T32max= 239,986 Nm Kp11= 3145,000 Kd11= 256,088 Kp12= 2415,507 Kd12= 455,686 Kp21= 2708,788 Kd21= 273,206 Kp22= 2220,573 Kd22= 248,426 Kp31= 5865,435 Kd31= 260,684 Kp32= 2865,677 Kd32= 279,645 Velocidade= 0,894 m/s	Ls= 0,986 m Hb= 0,726 m Beta= 41,081 % Fc= 0,173 m L11= 0,617 m L12= 0,382 m L21= 0,294 m L22= 0,705 m L31= 0,495 m L32= 0,504 m O1= -0,417 m O2= -0,522 m O3= -0,746 m Mb= 71,389 kg M11= 3,566 kg M12= 11,512 kg M21= 3,728 kg M22= 3,221 kg M31= 2,897 kg M32= 3,684 kg Kxh= 82379,304 N/m Kyh= 9725,521 N/m Bxh= 858,405 N/m Byh= 108,804 N/m T11min= -175,096 Nm T11max= 96,416 Nm T12min= -181,288 Nm T12max= 201,025 Nm T21min= -275,444 Nm T21max= 175,724 Nm T22min= -211,579 Nm T22max= 129,014 Nm T31min= -228,025 Nm T31max= 200,180 Nm T32min= -132,306 Nm T32max= 240,113 Nm Kp11= 3144,999 Kd11= 250,009 Kp12= 2437,992 Kd12= 457,229 Kp21= 2611,638 Kd21= 272,864 Kp22= 2220,574 Kd22= 248,426 Kp31= 5952,859 Kd31= 247,737 Kp32= 2865,751 Kd32= 279,646 Velocidade= 0,887 m/s
E_{av} = 318,715 J/m	E_{av} = 570,561 J/m	E_{av} = 556,077 J/m
ϵ_{xyH} = 0,297 m	ϵ_{xyH} = 0,415 m	ϵ_{xyH} = 0,433 m
Distância= 0,967 m	Distância= 0,979 m	Distância= 0,979 m

9. CONCLUSÕES E PERSPECTIVAS DE TRABALHO FUTURO

No decorrer deste trabalho foram abordados vários temas e vários aspectos relacionados com a locomoção através de pernas. Como foi referido anteriormente, os sistemas de locomoção com pernas apresentam vantagens significativas relativamente aos sistemas móveis com rodas, mas também apresentam algumas desvantagens sendo uma delas o maior consumo energético necessário para a deslocação e também a estabilidade/suavidade do deslocamento do robô.

Neste trabalho foram optimizados os parâmetros do robô para seis padrões de locomoção. No entanto, as diferenças entre eles, no que diz respeito aos parâmetros óptimos do robô, não são muito significativas e conseqüentemente os valores da densidade de energia absoluta média por distância percorrida (E_{av}), do erro do seguimento da trajectória das ancas (ϵ_{xyH}) e da distância percorrida pelo robô também não diferem de forma significativa que nos leve a indicar um padrão de locomoção específico, como o melhor para implementar a locomoção.

Este trabalho foi mais um, a acrescentar aos muitos já existentes, que tentou minimizar estas desvantagens. A utilização dos AG para obter os parâmetros do robô, de maneira a obter-se um baixo consumo energético e um deslocamento estável, parece ser um campo de pesquisa bastante interessante e que pode dar frutos. Apesar de um elevado número de linhas de código (cerca de 13000), o AG desenvolvido poderia ser ainda melhorado, quer com a introdução de mais métodos dos operadores genéticos, quer com a introdução de mais parâmetros passíveis de serem otimizados.

De referir que os objectivos inicialmente colocados foram atingidos. No entanto após, a análise dos resultados obtidos ao longo das várias experiências efectuadas, duas perguntas surgem:

- A primeira está relacionada com o comportamento dos gráficos quando o valor da velocidade de locomoção do robô ultrapassa os 5 m/s. Será que a trajectória linear planeada para as ancas não é a mais indicada para altas velocidades (deveria ser uma outra trajectória, por exemplo do tipo $A \sin(a \omega t) + B \cos(b \omega t)$)? Ou será que o modelo adoptado para o sistema robótico de locomoção não é o mais indicado para essa gama de valores de velocidade?
- A segunda questão está relacionada com os valores do comprimento das pernas do robô e da velocidade óptima de deslocamento ($L_{i1}+L_{i2}= 1$ m e $V_F= 1$ m/s). Será que o comprimento das pernas do robô está directamente relacionado com o valor da velocidade óptima de deslocamento? Ou seja, caso as pernas do robô apresentassem um comprimento de 2 m, a velocidade óptima de deslocamento seria aproximadamente 2 m/s?

Seria interessante analisar estes dois aspectos num trabalho futuro.

Relativamente às perspectivas para um futuro desenvolvimento deste trabalho, existem alguns outros pontos a melhorar, nomeadamente, no número de parâmetros a otimizar. Apesar de neste trabalho se ter considerado um número significativo de parâmetros (cromossoma com 48 genes), ficaram alguns de fora que seriam interessantes considerar. Entre os parâmetros a ter em conta num futuro trabalho encontram-se os seguintes:

- A complacência entre os diversos segmentos do corpo;
- A complacência entre o pé do robô e o solo;
- O número de pernas do robô;
- O centro de massa das pernas;
- A inércia das pernas;
- O número de gdl das pernas;
- O tipo de trajectória das ancas;
- O tipo de trajectória dos pés.

Referências Documentais

- [1] Motion Control Technology, Space Exploration, Disponível on-line no site: http://nasatech.com/motion/features/feat1_0805.html, última consulta em Julho 2008.
- [2] Aaron Wilson, Dogbot by Aaron Wilson, Disponível on-line no site: <http://www.arrickrobotics.com/robomenu/dogbot.html>, última consulta em Julho 2008.
- [3] Digital Drops, Metamorfose: O Robô Barata que roda Linux, Disponível on-line no site: <http://digitaldrops.com.br/drops/2007/07/metamorfose-o-robo-barata-que-roda-linux.html>, última consulta em Julho 2008.
- [4] Laboratoire de Mécanique des Solides, Marche humaine & mécanique: robot bipède BIP 2000, Disponível on-line no site: <http://www-lms.univ-poitiers.fr/rubrique71.html>, última consulta em Julho 2008.
- [5] Sony Entertainment Robot Europe, AIBO Your artificial intelligent companion, Disponível on-line no site: <http://support.sony-europe.com/aibo/>, última consulta em Julho 2008.
- [6] Nelson G.M.; Quinn R.D.; Bachmann R.J.; *et al.* (1997). Design and simulation of a cockroach-like hexapod robot. In *Proceedings of the IEEE international Conference on Robotics and Automation*, Albuquerque, New Mexico, pp. 1106-1111.
- [7] Kirchner F.; Klaassen B.; Linnemann R.; *et al.* (2002). Biomimetic Walking Robot Scorpion: Control and Modeling. *Special Issue of the Robotics and Autonomous Systems*, vol. 41, n°. 2-3, pp. 69-76.
- [8] Low, K.H.; Leow, W.K.; Ang Jr., M.H. (2002). A Hybrid Mobile Robot Architecture with Integrated Planning and Control, In *Proceedings of the 1st International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS-02)*, Bologna, Italy, pp. 219-226.
- [9] Clarke, R. (1993). Asimov's Laws of Robotics Implications for Information Technology. In *Proceedings of the IEEE Computer Society Press*, vol. 26, n°. 12, pp. 53-61
- [10] KUKA Automation + Robotics, KR 1000 titan, Disponível on-line no site: http://www.kuka.com/united_kingdom/en/products/industrial_robots/heavy/kr1000/, última consulta em Julho 2008.
- [11] Jet Propulsion Laboratory, Mars Pathfinder Mission Status, Disponível on-line no site: <http://mpfwww.jpl.nasa.gov/MPF/rover/sojourner.html>, última consulta em Julho 2008.
- [12] United Press International, iRobot's PackBot on the front lines, Disponível on-line no site: <http://www.physorg.com/news11166.html>, última consulta em Julho 2008.
- [13] The Next Generation – An introduction to medicine, Robotic Surgery, Disponível on-line no site: http://www.nextgenmd.org/vol2-5/robotic_surgery.html, última consulta em Julho 2008.

- [14] Annika, The motivation behind creating Cog, Disponível on-line no site: <http://www.ai.mit.edu/projects/humanoid-robotics-group/cog/overview.html>, última consulta em Julho 2008.
- [15] American Honda Motor Co. Inc., The world's most advanced humanoid robot, Disponível on-line no site: <http://asimo.honda.com/>, última consulta em Julho 2008.
- [16] Ferrell, C. (1993). Robust agent control of an autonomous robot with many sensors and actuators. In *MIT Artificial Intelligence Lab Technical Report 1443*, pp. 152-165.
- [17] Massion, J. (1997). Cerveau et motricité: fonctions sensori-motrices. *Imprimerie des Presses Universitaires de France*, ISBN 2-13-047591-4, p. 187.
- [18] Cruse, H. (1990). What mechanisms coordinate leg movement in walking arthropods?. In *Trends in neurosciences*, vol. 13, n°. 1, pp. 15-21.
- [19] Celaya, E.; Porta, J. (1998). A Control Structure for the locomotion of a legged Robot on difficult Terrain. In *Proceedings of the IEEE Robotics and Automation Magazine*, vol. 5, n°. 2, pp. 43-51.
- [20] Graham, D. (1985). Pattern and control of walking in insects. In *Advances in Insect Physiology*, vol.18, pp. 131-140.
- [21] Randall, J. (1999). Stable adaptive neural control of systems with closed kinematic chains applied to biologically-inspired walking robots. *These in Philosophy*. Bristol: Faculty of engineering, University of the West of England, Bristol, p. 322.
- [22] Beer, R.D.; Kacmarcik, G.; Ritzmann, R.; et al. (1991). A model of distributed sensorimotor control in the cockroach escape turn. In R.P.Lippmann, J.E.Moody, and D.S.Touretzky (Eds.), In *Neural information Processing Systems*, Morgan Kaufmann, vol. 3, pp. 436-442.
- [23] Camhi, J. (1988). Escape behavior in the cockroach: distributed neural processing. In *Cellular and Molecular Life Sciences (CMLS)*, Birkhäuser Basel, vol. 44, n°. 5, pp. 401-408.
- [24] Cymbalyuk, G.; Borisyuk, R.; Mueller-Wilm, U.; et al. (1998). Oscillatory network controlling six-legged locomotion optimization of model parameters. In *Neural Networks*, vol. 11, n°. 7-8, pp. 1449-1460.
- [25] Labhart, T.; Meyer, E.P. (2002). Neural mechanisms in insect navigation: polarization compass and odometer. In *Current Opinion in Neurobiology*, Elsevier Science, vol. 12, n°. 6, pp. 707-714.
- [26] Lambrinos, D.; Möller, R.; Labhart, T.; et al. (2000). A mobile robot employing insect strategies for navigation. In *Robotics and Autonomous Systems*, vol. 30, n°. 1, pp. 39-64.
- [27] Cruse, H. (1979). A new model describing the coordination pattern of the legs of a walking stick insect. In *Biological Cybernetics*, vol. 32, pp. 107-113.
- [28] Cruse, H. (1976). The function of the legs in the free walking stick insect, *carausius morosus*. In *Journal of Comparative Physiology*, vol. 112, pp. 235-262.

- [29] Cruse, H. (1981). Is the position of the Femur-Tibia joint under feedback control in the walking stick insect?. Force Measurements. In *Journal of Experimental Biology*, vol. 92, pp. 87-95.
- [30] Cruse, H. (1980). A quantitative model of walking incorporating central and peripheral influences: The control of the individual leg. In *Biological Cybernetics*, vol. 37, n°. 3, pp. 131-136.
- [31] Cruse, H. (1976). The control of body position in the stick insect (*carausius morosus*), when walking over uneven terrain. In *Biological Cybernetics*, vol. 24, pp. 25-33.
- [32] Cruse, H. (1980). A quantitative model of walking incorporating central and peripheral influences: The connections between the different legs. In *Biological Cybernetics*, vol. 37, n°. 3, pp. 137-144.
- [33] Kingsley, D.A.; Quinn, R.D.; Ritzmann, R.E. (2003). A cockroach inspired robot with artificial muscles. In *Proceedings of the International Symposium on Adaptive Motion of Animals and Machines*, Kyoto, Japan, p. 7.
- [34] Kirchner, F. (1998). Q-learning of complex behaviours on a six-legged walking machine. In *Robotics and Autonomous Systems*, pp. 253-262.
- [35] Biologically Inspired Robotics Laboratory, Case Western Reserve University, Quinn R.D., Disponível on-line no site: <http://biorobots.cwru.edu>, última consulta em Julho de 2008.
- [36] Quinn, R.D.; Nelson, G.M.; Bachmann, R.J.; et al. (2001). Insect designs for improved robot mobility. In *Proceedings of the 4th International Conference On Climbing and Walking Robots*, Berns and Dillmann (Eds), pp. 69-76.
- [37] Quinn, R.D.; Nelson, G.M.; Bachmann, R.J.; et al. (2003). Parallel strategies for implementing biological principles into mobile robots. In *Journal of Robotics Research*, Sage Publications, vol. 22, n°. 3, pp. 169-186.
- [38] Beer, R.D, Chiel, H. (1993). Simulation of cockroach locomotion and escape. In *Biological Neural Networks in Invertebrate Neuroethology and Robotics*, Academic Press, Boston, pp. 267-286.
- [39] Wilson, D.M. (1966). Insect walking. In *Annual Review of Entomology*, vol. 11, pp. 103-121.
- [40] Espenschied, K.S.; Quinn, R.D.; Chiel, H.J.; et al. (1996). Biologically-based distributed control and local reflexes improve rough terrain locomotion in a hexapod robot. In *Robotics and Autonomous Systems*, vol. 18, pp. 59-64.
- [41] Espenschied, K.S.; Quinn, R.D.; Chiel H.J.; et al. (1993). Leg coordination mechanisms in stick insect applied to hexapod robot. In *Adaptive Behavior*, vol. 1, n°. 4, pp. 455-468.
- [42] Altendorfer, R.; Moore, N.; Komsuoglu, H.; et al. (2001). RHex: A Biologically Inspired Hexapod Runner. In *Autonomous Robots*, vol. 11, n°. 3, pp. 207-213.

- [43] Altendorfer, R.; Saranli, U.; Komsuoglu, H.; et al. (2001). Dynamic Evidence for Spring Loaded Inverted Pendulum Running in a Hexapod Robot. In *Experimental Robotics VII*, Springer Verlag, ISBN 3-540-42104-1, pp. 291-302.
- [44] Saga Science – Portraits Robots, La fourmi, la blatte et le robot!, disponível no site: <http://www.cnrs.fr/>, última consulta em Julho 2008.
- [45] Johannet, A.; Sarda, I. (1995). Behaviour learning by a reward-penalty algorithm: from gait learning to obstacle avoidance by neural networks. In *Proceedings of International Conference Artificial Neural Nets and Genetic Algorithms*, France, pp. 465-467.
- [46] Touzet, C.; Sarzeaud, O. (1992). Application d'un algorithme d'apprentissage par pénalité/récompense a la génération des formes locomotrices hexápodes. *Journées de rochebrune*, AFCET IA. Et cognition, p. 5.
- [47] Nelson, G.M.; Quinn, R.D. (1998). Posture Control of a Cockroach-like Robot. In *Proceedings of IEEE International Conference on Robotics and Automation*, Belgium, vol. 19, n°. 2, pp. 9-14.
- [48] Saranli, U.; Buehler, M. (1999). Modeling and Analysis of a Spatial Compliant Hexapod. *Technical Papers, Department of Electrical Engineering and Computer Science*, McGill University, Montreal, Canada, p. 18.
- [49] Buehler, M.; Saranli, U.; Papadopoulos, D.; et al. (2000). Dynamic locomotion with four and six-legged robots. In *Proceedings of the International Symposium on Adaptive Motion of Animals and Machines*, Montreal, Canada.
- [50] Reichler, J.A.; Delcomyn, F. (2000). Dynamics Simulation and Controller Interfacing for Legged Robots. In *The International Journal of Robotics Research*, vol. 19, n°. 1, pp. 49-58.
- [51] Johannet, A.; Sarda, I. (1994). Gait learning of hexapod robot with neural networks: from simulation to realization. In *2^{ème} congrés Franco-Japonais de takamatsu*, p. 4.
- [52] Porta, J.M.; Celaya, E. (1998). Gait analysis for six-legged robots. *Document Tècnic de l'Institut de Robòtica i Informàtica Industrial*, IRI-DT-9805, Barcelona, p. 10.
- [53] Svinin, M.; Yamada, K.; Ueda, K. (2001). Emergent synthesis of motion patterns for locomotion robots. In *Artificial intelligence in engineering*, vol. 15, n°. 4, pp. 353-363.
- [54] Pratt, J.; Torres, A.; Dilworth, P.; et al. (1996). Virtual actuator control. In *Proceedings of the International Conference on Intelligent Robots and Systems*, IROS, Osaka, Japan, p. 8.
- [55] Torres, A. (1996). Virtual model control of a hexapod walking robot, S.B. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts, p. 55.
- [56] Kodjabachian, J.; Meyer, J. (1998). Evolution and Development of Neural Controllers for Locomotion, Gradient-Following, and Obstacle-Avoidance in Artificial Insects. In *Proceedings of IEEE Transactions on Neural Networks*, vol. 9, n° 5, pp. 796-812.

- [57] Meyer, J.A. (1998a). Evolutionary approaches to neural control in mobile robots. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, San Diego, CA, pp. 2418-2423.
- [58] Meyer, J.A. (1998b). Evolutionary approaches to walking and higher-level behaviors in 6-legged animats. In Gomi (Ed.). *Evolutionary Robotics, From Intelligent Robots to Artificial Life (ER'98)*. AAAI Books, vol. 11, p. 211-217.
- [59] Saranli, U.; Koditschek, D.E. (2003). Template Based Control of Hexapedal Running. In *Proceedings of the IEEE International Conference On Robotics and Automation*, vol. 1, pp. 1374-1379.
- [60] Saranli, U. (2002). Dynamic locomotion with a hexapod robot, PhD. Thesis, The University of Michigan, p. 205.
- [61] Saranli, U.; Buehler, M.; Koditschek D.E. (2000). Design, Modeling and Preliminary Control of a Compliant Hexapod Robot. In *Proceedings of IEEE International Conference on Robotics and Automation*, San Francisco, CA, pp. 2589-2596.
- [62] Brooks, R.D. (1989). A robot that walks, emergent behaviors from a carefully evolved network. In *Neural Computation Archive*, vol. 1, pp. 365-382.
- [63] Brooks, R.D. (1986). A robust layered control system for a mobile robot. In *Proceedings of IEEE Journal of Robotics and Automation*, vol. 2, nº. 1, pp. 14-23.
- [64] Celaya, E.; Porta, J.M. (1996). Control of a Six-Legged Robot Walking on Abrupt Terrain. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, vol. 3, pp. 2731-2736.
- [65] Celaya E.; Porta, J. M. (1995). Force-Based Control of a Six-legged Robot on Abrupt Terrain Using the Subsumption Architecture. In *Proceedings of the 7th International Conference on Advanced Robotics*, Sant Feliu de Guíxols, Spain, pp. 413-419.
- [66] Porta, J.M.; Celaya, E. (2000). Body and leg coordination for omnidirectional walking in rough terrain. In *Proceedings of Third International Conference on Climbing and Walking Robots*, Madrid, Spain, pp. 499-506.
- [67] Ganascia, J.G. (1993). *Inteligência Artificial*, Ed. Ática.
- [68] Boose, J.H. (1984). Personal construct theory and the transfer of human expertise. In *Proceedings of the American Association for Artificial Intelligence*. California, pp. 27-33.
- [69] Simons, G.T. (1988). *Introdução à Inteligência Artificial*, Ed. Classe.
- [70] Losano, M.G. (1992). *História dos autómatos*, Ed. Companhia das Letras, São Paulo.
- [71] Tafner, M.A.; Xerez, M.; Rodrigues, I.W. (1995). *Redes Neurais Artificiais, Introdução e Princípios de Neurocomputação*, Ed. Eko e Furb, Blumenau.
- [72] Rich, E.; Knight, K. (1994). *Inteligência artificial*, Makron Brasil Editora Ltda. (MacGraw-Hill), 2nd edition.

- [73] Muller, F.M. (1997). Métodos Heurísticos e Metaheurísticos, *II Simpósio Nacional de Informática, Faculdades Franciscanas, Santa Maria – RS.*
- [74] Hokazono, A.F., Grupo de Sistemas Inteligentes, Disponível on-line no site: <http://www.din.uem.br/ia>, última consulta em Julho 2008.
- [75] Darwin C. (1859). On the origin of species by means of natural selection, or the preservation of favoured races in the struggle for life, John Murray, 1st edition, London, UK.
- [76] Holland, J.H. (1975). Adaptation in Natural and Artificial Systems, The University of Michigan Press, Ann Arbor, MI.
- [77] Ackley, D.H. (1987). A Connectionist Machine for Genetic Hillclimbing, Kluwer Academic Publishers, Boston, MA.
- [78] Bhanu B.; Lee, S.; Ming, J. (1991). Self-optimizing image segmentation system using a genetic algorithm. In R. K. Belew and L. B. Booker, editors, In *Proceedings of the Fourth International Conference on Genetic Algorithms and Their Applications*, San Mateo, CA, pp. 362-369.
- [79] Goldberg, D.E. (1989). Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wiley, Reading, MA.
- [80] Dorigo M.; Schnepf, U. (1993) Genetic-based machine learning and behavior-based robotics: a new syndissertation. In *Proceedings of the IEEE Transactions on System, Man, and Cybernetics*, vol. 23, nº. 1, pp. 144-154.
- [81] Goldberg, D.E.; Holland, J.H. (1988). Genetic algorithms and machine learning, In *Machine Learning*, vol. 3, nº. 2-3, pp. 95-99.
- [82] Schaffer, J.D. (1980). Some Experiments in Machine Learning using Vector Evaluated Genetic Algorithms. Ph.D.dissertation, Vanderbilt University.
- [83] Fogel, L.J.; Owens, A.J.; Walsh, M.J. (1966). Artificial Intelligence Through Simulated Evolution. John Wiley & Sons, New York.
- [84] Beasley, D.; Bull, D.R.; Martin, R. (1993). An overview of genetic algorithms: Part 1, fundamentals, In *University Computing*, vol. 15, nº. 2, pp. 58-69.
- [85] Caruana, R.A.; Schaffer, J.D. (1988). Representation and hidden bias: Gray v.s. binary coding for genetic algorithms. In M. B. Morgan, editor, In *Proceedings of the Proceedings of the Fifth International Workshop on Machine Learning*, San Mateo, CA, June. Morgan Kaufmann, pp. 153-161.
- [86] Back, T. (1996). Evolutionary Algorithms in Theory and Practice, Oxford Univ. Press, New York.
- [87] Back,T.; Hoffmeister, F.; Schwefel, H. (1991) A survey of evolution strategies. In R. K. Belew and L. B. Booker, editors, In *Proceedings of the Fourth International Conference on Genetic Algorithms and Their Applications*, San Mateo, CA, pp. 2-9.

- [88] Bramlette, M.F. (1991) Initialization, mutation and selection methods in genetic algorithms for function optimization. In R. K. Belew and L. B. Booker, editors, In *Proceedings of the Fourth International Conference on Genetic Algorithms and Their Applications*, San Mateo, CA, pp. 100-107.
- [89] Goldberg, D.E. (1989). Optimal initial population size for binary-coded genetic algorithms. *Technical Report TCGA Report No. 85001*, University of Alabama.
- [90] Grefenstette, J.J.; Baker, J.E. (1989). How genetic algorithms work: A critical look at implicit parallelism. In J. D. Schaffer, editor, In *Proceedings of the Third International Conference on Genetic Algorithms and Their Applications*, San Mateo, CA, pp. 20-27.
- [91] De Jong, K.A. (1975). An Analysis of the Behavior of a Class of Genetic Adaptive Systems. Ph.D. dissertation, University of Michigan.
- [92] De Jong, K.A. (1988). Learning with genetic algorithms: an overview, In *Machine Learning*, vol. 3, n°. 2-3, pp. 121-138.
- [93] Whitley, D. (1989). The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best, In Schaffer, J.; In *Proceedings of the Third International Conference on Genetic Algorithms*, San Mateo, CA, pp. 116- 121.
- [94] Whitley, D.L. (1992). A genetic algorithm tutorial, Technical Report, Colorado State University, pp. 93-103.
- [95] Beasley, D.; Bull, D.R.; Martin, R.R. (1993) An Overview of Genetic Algorithms: Part 1, Fundamentals. In *University Computing*, vol. 15, n°. 2, pp. 58-69.
- [96] Syswerda, G. (1989). Uniform crossover in genetic algorithms. In J. D. Schaffer, In *Proceedings of the Third International Conference on Genetic Algorithms and Their Applications*, San Mateo, CA, pp. 2-9.
- [97] Dejong, K. (1975). The analysis and behaviour of a class of genetic adaptive systems. PhD thesis, University of Michigan.
- [98] Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*, MIT Press.
- [99] Back, T.; Fogel, D.B.; Michalewicz, Z. (1997). *Handbook of Evolutionary Computation*, Institute of Physics Publishing and Oxford University Press, Bristol, New York.
- [100] Heitkotter, J. (1998). ENCORE The Evolutionary Computation REpository network, Disponível on-line no site: <http://surf.de.uu.net/encore/>, última consulta em Julho 2008.
- [101] Haupt, R.L.; Haupt, S.E. (1998). *Practical Genetic Algorithms*, Wiley-Interscience Publications, NY.
- [102] Davis, L. (1991). *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York.
- [103] Eshelman, L.J.; Shaffer, D.J. (1992). Real-coded genetic algorithms and interval-schemata. In: WHITLEY, D. L. (ed). In *Proceedings of the Foundations of Genetic Algorithms 3*. San Mateo, CA, Morgan Kaufman, pp. 187- 203.

- [104] Michalewicz, Z. (1994). *Genetic Algorithms + Data Structures = Evolution Programs*, 3 ed., Springer-Verlag.
- [105] Austin, A.S. (1990). Genetic solutions to XOR problems, In *AI Expert archive*, vol. 5, pp. 52-57.
- [106] Cleveland, G.A.; Smith S.F. (1989). Using genetic algorithms to schedule flow shop releases. In J. D. Schaffer, In *Proceedings of the Third International Conference on Genetic Algorithms and Their Applications*, San Mateo, CA, Morgan Kaufmann, pp. 160-169 .
- [107] Goldberg, D.E. (1989). Sizing populations for serial and parallel genetic algorithms. In J. David Schaffer, editor, In *Proceedings of the Third International Conference on Genetic Algorithms and Their Applications*, San Mateo, CA, Morgan Kaufmann, pp. 70-79.
- [108] Reeves, C.R. (1993). Using genetic algorithms with small populations. In S. Forrest, In *Proceeding of the Fifth International Conference on Genetic Algorithms*, San Mateo, CA, Morgan Kaufmann, pp. 92-99.
- [109] Robertson, G.G. (1988). Population size in classifier systems. In M. B. Morgan, editor, In *Proceedings of the Fifth International Workshop on Machine Learning*, San Mateo, CA, Morgan Kaufmann, pp. 142-152.
- [110] Baker, J. (1987). Reducing bias and inefficiency in the selection algorithm. In Grefenstette, J. ed.; In *Proceedings of the Second International Conference on Genetic Algorithms and Their Applications*, Hillsdale, New Jersey: Lawrence Erlbaum Associates, pp. 14-21.
- [111] Fonseca, C.M.; Fleming, P.J. (1995). An Overview of Evolutionary Algorithms in Multiobjective Optimization. In *Proceedings of the Evolutionary Computation*, vol. 3 n°. 1, pp. 1-16.
- [112] Fonseca. C.M.; Fleming, P.J. (1997). Multiobjective optimization, In Back, T.; Fogel, D.; Michalewicz, Z.; *Handbook of Evolutionary Computation*, Institute of Physics Publishing and Oxford University Press, Bristol, New York, p. 1-9.
- [113] Horn, J.; Nafpliotis, N. (1992). Multiobjective optimization using the niched pareto genetic algorithm. Technical Report, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL.
- [114] Nolfi, S.; Floreano, D. (2000). *Evolutionary robotics: the biology, intelligence, and technology of self-organizing machines*, MIT Press/Bradford Books: Cambridge, USA.
- [115] Meyer, J.A. (1998). Evolutionary approaches to neural control in mobile robots, In *Proceedings of the IEEE International Conference on Systems and Cybernetics*, San Diego, pp. 35-40.
- [116] Horchler, A.D.; Reeve, R.E.; Webb, B.H.; Quinn, R.D. (2004) Robot Phonotaxis in the Wild: a Biologically Inspired Approach to Outdoor Sound Localization, In *Advanced Robotics*, vol. 18, n°. 8, pp. 801-816.

- [117] Terra, M.; Tinós, R. (2001). Fault detection and isolation in robotic manipulators via neural networks - a comparison among three architectures for residual analysis, In *Journal of Robotic Systems*, vol. 18, n°. 7, pp. 357-374.
- [118] Tinós, R.; Navarro-Serment, L.E.; Paredis, C.J.J. (2001). Fault tolerant localization for teams of distributed robots, In *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1061-1066.
- [119] Billard, A.; Ijspeert, A.J.; Martinoli, A. (2000). A multirobot system for adaptive exploration of a fast changing environment: probabilistic modeling and experimental study, In *Connection Science*, vol. 11, n°. 3-4, pp. 357-377.
- [120] Floreano, D.; Urzelai, J. (2000). Evolutionary robots with on-line self-organization and behavioral fitness, In *Neural Networks*, vol. 13, pp. 431-443.
- [121] Jakobi, N. (1997). Half-baked, ad-hoc and noisy: minimal simulations for evolutionary robotics, in P. Husbands and I. Harvey (eds), In *Proceedings of the Fourth European Conference on Artificial Life*, MIT Press, pp. 348-357.
- [122] Thompson, A. (1998). On the automatic design of robust electronics through artificial evolution, in M. Sipper and A. Prez-Urbe (eds), In *Proceedings of the 2nd International Conference on Evolvable Systems: From Biology to Hardware*, Springer, pp. 13-24.
- [123] Floreano, D.; Nolfi, S.; Mondada, F. (2001). Co-evolution and ontogenetic change in competing robots, in M. Patel, V. Honavar and K. Balackishnan (eds), In *Advances in the Evolutionary Synthesis of Intelligent Agents*, MIT Press.
- [124] Marley, G. (1996). Landscapes, learning costs and genetic assimilation, In *Evolutionary Computation*, vol. 4, n°. 3, pp. 213-234.
- [125] Cobb, H.G.; Grefenstette, J.J. (1993). Genetic algorithms for tracking changing environments, in S. Forrest (ed.), In *Proceedings of the 5th International Conference on Genetic Algorithms*, pp. 523-530.
- [126] Branke, J. (2003). Evolutionary approaches to dynamic optimization problems - introduction and recent trends, in J. Branke (ed.), *GECCO Workshop on Evolution Algorithms for Dynamic Optimization Problems*, pp. 2-4.
- [127] Bekey, G.A. (2005). *Autonomous Robots: From Biological Inspiration to Implementation and Control*, The MIT Press, Cambridge, MA, USA.
- [128] Kohl, N.; Stone, P. (2004). Policy gradient reinforcement learning for fast quadrupedal locomotion. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, New Orleans, LA, USA, pp. 2619-2624.
- [129] Reeve, R.; Hallam, J. (2005). An analysis of neural models for walking control. In *Proceedings of the IEEE Transactions on Neural Networks*, vol. 16, n°. 3, pp. 733-742.
- [130] McGhee, R. (1976). Robot locomotion, In *Neural Control of Locomotion*, Plenum Press, New York, pp. 237-264.

- [131] Raibert, M. H. (1986). *Legged Robots That Balance*. MIT Press, Cambridge, MA, USA.
- [132] Lewis, M.A.; Fagg, A.H.; Solidum, A. (1992). Genetic programming approach to the construction of a neural network for control of a walking robot. In *Proceedings of the IEEE International Conf. on Robotics and Automation (ICRA)*, Nice, France, pp. 2618-2623.
- [133] Bongard, J.C.; Pfeifer, R. (2002). A method for isolating morphological effects on evolved behaviour. In *Proceedings of the 7th International Conference on Simulation of Adaptive Behaviour (SAB)*, Edinburgh, UK, pp. 305-311.
- [134] VA Linux Systems, Dynamechs, disponível on-line no site: <http://dynamechs.sourceforge.net/>, última consulta em Julho 2008.
- [135] Golubovic, D.; Hu, H. (2003). Ga-based gait generation of sony quadruped robots. In *Proceedings of the 3th International Conference on Artificial Intelligence and Applications (AIA)*, Benalmadena, Spain, pp. 118-123.
- [136] Heinen, M.R.; Osório, F.S. (2006a). Applying genetic algorithms to control gait of physically based simulated robots. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, Vancouver, Canada. IEEE World Congress on Computational Intelligence (WCCI), pp. 1823 - 1830.
- [137] Heinen, M.R.; Osório, F.S. (2006b). Gait control generation for physically based simulated robots using genetic algorithms. In *Proceedings of the International Joint Conference 2006, 10th Ibero-American Conference on AI (IBERAMIA), 18th Brazilian Symposium on AI (SBIA)*, LNCS, Ribeirão Preto - SP, Brazil. Springer-Verlag, pp. 562-571.
- [138] Dudek, G.; Jenkin, M. (2000). *Computational Principles of Mobile Robotics*, Cambridge University Press, Cambridge, UK.
- [139] Wyeth, G.; Kee, D.; Yik, T.F. (2003). Evolving a locus based gait for a humanoid robot. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, USA, vol. 2, pp. 1638-1643.
- [140] Farritor, S.; Dubowsky, S.; Rutman, N.; Cole, J. (1996). A Systems-Level Modular Design Approach to Field Robotics. In *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pp. 2890 - 2895.
- [141] Leger, C. (2000) *DARWIN2K - An Evolutionary Approach to Automated Design for Robotics*. Kluwer Academic Publishers.
- [142] Juarez-Guerrero, J.; Munoz-Gutierrez, S.; Mayol-Cuevas, W.W. (1998). Design of a Walking Machine Structure Using Evolutionary Strategies. In *Proceedings of the 1998 IEEE International Conference on Systems, Man and Cybernetics*, pp. 1427-1432.
- [143] Ishiguro, A.; Kawasumi, K.; Fuji, A. (2002). Increasing Evolvability of a Locomotion Controller Using a Passive-Dynamic-Walking Embodiment. In *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2581-2586.

- [144] Endo, K.; Yamasaki, F.; Maeno, T.; Kitano, H. (2002a). A Method for Co-Evolving Morphology and Walking Pattern of Biped Humanoid Robot. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, pp. 2775-2780.
- [145] Endo, K.; Maeno, T.; Kitano, H. (2002b). Co-Evolution of Morphology and Walking Pattern of Biped Humanoid Robot Using Evolutionary Computation - Consideration of Characteristic of the Servomotors. In *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2678-2683.
- [146] Endo, K.; Maeno, T.; Kitano, H. (2003). Co-Evolution of Morphology and Walking Pattern of Biped Humanoid Robot Using Evolutionary Computation - Evolutionary Designing Method and its Evaluation, In *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 340-345.
- [147] Burrows, M.; Hoyle, G. (1973). The Mechanism of Rapid Running in the Ghost Crab, *Ocypode Cerathophthalma*, In *Journal of Experimental Biology*, Vol. 58, pp. 327-349.
- [148] Herreid, C.F. (1981), Locomotion in Arthropods: An Evolutionary Phantasmagoria, Locomotion and Energetics in Arthropods, In *Plenum Press*, New York, pp. 1-7.
- [149] Luk, B.L.; Cooke, D.S.; Collie, A.A.; White, T.S. (1994). Robug III - A semi-intelligent teleoperated walking and climbing robot for disordered hazardous environments, In *Proceedings of the European Robotics and Intelligent Systems Conference*, vol. 3, pp. 1162-1168.
- [150] Galt, S. (1996). A Survey of the Existing Gait Generation Strategies for Legged Robots, *Internal Report No. 96/0016*.
- [151] Beer, R.D.; Gallagher, I.C. (1992). Evolving Dynamic Neural Networks for Adaptive Behavior, In *Adaptive Behavior*, vol. 1, pp. 91-122.
- [152] Lewis, M.A.; Fagg, A.H.; Solidum A. (1992). Genetic Programming Approach to the Construction of a Neural Network for Control of a Walking Robot, In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2618-2623.
- [153] Fujii, A.; Ishiguro, A.; Otsu, K.; *et al.* (2000). Evolutionary Creation of an Adaptive Controller for a Legged-Robot: A Dynamically-Rearranging Neural Network Approach. In *Proceedings of the International Symposium on Adaptive Motion of Animals and Machines*.
- [154] Hornby, G.; Fujita M.; Takamura S.; *et al.* (1999). Autonomous Evolution of Gaits with the Sony Quadruped Robot. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1297-1304.
- [155] Kodjabachian, J.; Meyer, J-A. (1998). Evolution and Development of Modular Control Architectures for Locomotion in Six-Legged Animats. In *Connection Science*, vol. 10, n°. 3-4, pp. 211-254.
- [156] Casper, J.L.; Micire, M.; Murphy, R.R. (2000). Issues in Intelligent Robots for Search and Rescue, In *Proceedings of the SPIE Ground Vehicle Technology II*, Orlando, Florida, pp. 292-302.

- [157] Latombe, J.C. (1991). Robot motion planning, Kluwer Academic Publisher, Boston.
- [158] Sugihara, K.; Smith, J. (1997). Genetic algorithms for adaptive motion planning of an autonomous mobile robot, In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, vol. 7, pp. 138-143.
- [159] Ashiru, I.; Czarnecki, C.; Routen, T. (1996). Characteristics of a genetic based approach to path planning for mobile robots, In *Network and Computer Applications*, vol. 19, n°. 2, pp. 149-169.
- [160] Gallardo, D.; Colomina, O.; Florez, F.; Rize, R. (1998). A genetic algorithm for robust motion planning, In *Proceedings of the 11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, Benicassim, Spain, vol. 2, pp. 115-121.
- [161] Hocaoglu, C.; Sanderson, C. (2001). Planning multiple paths with evolutionary speciation, In *Proceedings of the IEEE Transactions on Evolutionary Computation*, vol. 5, n°. 3, pp. 169-191.
- [162] Xiao, J.; Michalewicz, Z.; Zhang, L.; Trojanowski, K. (1997). Adaptive evolutionary planner/navigator for mobile robots, *IEEE Transactions on Evolutionary Computation*, vol. 1, n°. 1, pp. 18-28.
- [163] Cohen, J.D.; Lin, M.C.; Manocha, D.; Ponamgi, M.K. (1995). I-collide: An interactive and exact collision detection system for large-scale environments, In *Proceedings of ACM, International 3D Graphics Conference*, pp. 189-196.
- [164] Silva, M.S. (2005). Dissertação sobre Sistemas Robóticos de Locomoção Multipernas, Faculdade de Engenharia da Universidade do Porto, Porto, pp. 109-166.

