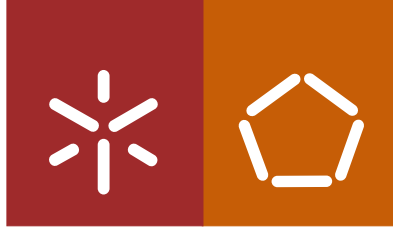


Universidade do Minho
Escola de Engenharia

Pedro Jorge Ribeiro Barbosa

Otimização de locomoção bípede



Universidade do Minho

Escola de Engenharia

Pedro Jorge Ribeiro Barbosa

Otimização de locomoção bípede

Dissertação de Mestrado
Ciclo de Estudos Integrados Conducentes ao
Grau de Mestre em Engenharia Biomédica

Trabalho efetuado sob a orientação da
**Professora Doutora Cristina Manuela
Peixoto dos Santos**

Outubro de 2012

É AUTORIZADA A REPRODUÇÃO PARCIAL DESTA DISSERTAÇÃO APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE;

Universidade do Minho, ___/___/_____

Assinatura: _____

Agradecimentos

Gostava de agradecer à minha orientadora, a professora Cristina Santos, por me ter dado a oportunidade de trabalhar nesta área de investigação e por me ter aconselhado e guiado durante a elaboração da dissertação.

Quero também agradecer ao investigador do laboratório ASBG Miguel Oliveira por todo o apoio e tempo disponibilizado durante estes últimos meses.

Quero agradecer à minha família, namorada e amigos por terem sido um suporte fundamental no meu trilha académico e por tornarem tudo isto possível.

Resumo

Atualmente verifica-se um crescimento exponencial a nível de desenvolvimento de sistemas robóticos móveis havendo um esforço para criar sistemas com propriedades mais eficientes e adaptáveis às exigências do ambiente de trabalho. Neste contexto, têm havido uma preocupação acrescida em desenvolver melhores sistemas de locomoção quer seja locomoção por rodas quer seja por pernas (bípede, quadrúpede e hexapode).

Esta dissertação foca-se na otimização da locomoção bípede a qual é uma área que tem sido alvo de grande atenção uma vez que esta é uma área da robótica que ainda necessita de progredir no sentido de conseguir finalmente uma locomoção tão eficiente como a marcha humana.

Deste modo, a elaboração deste trabalho teve como objetivos principais a criação de uma estratégia de otimização que combinasse a geração de padrões de movimento através de geradores centrais de padrões (CPGs) com um algoritmo de otimização evolucionário (*Non-Dominated Sorting Genetic Algorithm II*). Essa estratégia implicou a determinação de objetivos que correspondem a características da locomoção bípede e que foram otimizados, sendo eles o deslocamento frontal, a altura a que o pé levanta, a força de impacto entre os pés e o chão e a posição do centro de massa.

Os resultados foram obtidos a partir de simulações na plataforma Webots para o robô bípede Darwin-OP. Neste contexto, os resultados foram muito satisfatórios uma vez que o algoritmo foi capaz de gerar locomoção estável e os objetivos propostos foram otimizados. Foi feito também um estudo de sensibilidade que determinou a existência de parâmetros de CPGs que apresentam uma forte correlação positiva com as funções objetivos. Assim, os parâmetros A_{compasso} , frequência ω e O_{Roll} influenciam fortemente o deslocamento e a força de impacto e o parâmetro A_{hPitch} influencia a altura a que o pé levanta.

No futuro seria pertinente aplicar o algoritmo elaborado num robô bípede real e conferir se consegue gerar uma locomoção eficiente em condições reais.

Abstract

Presently there is an exponential increase on the level of development of mobile robotic systems and so there is an effort to create systems with properties more efficient and adaptable to the demands of the work environment. In this context, there has been a heightened concern in developing better systems of locomotion either by wheels either by legs (bipedal, 4-legged or 6-legged).

This dissertation focuses on the optimization of bipedal locomotion which is an area that has been the subject of much attention since this is an area of robotics that still needs to make progress towards finally achieving locomotion as efficient as the human gait.

Thus, this work aimed to create an optimization strategy that combines the generation of movement patterns through central pattern generators (CPGs) with an evolutionary optimization algorithm (Non-Dominated Sorting Genetic Algorithm II). This strategy involved the determination of objectives that correspond to characteristics of bipedal locomotion and that have been optimized, namely the frontal displacement, the ground clearance, the impact force between the foot and the ground and the position of the center of mass.

The results were obtained from simulations in Webots platform for the bipedal robot Darwin-OP. The results were very satisfactory since the algorithm was able to generate stable locomotion and the proposed objectives were optimized. We also made a sensitivity analysis that determined the existence of CPGs parameters that exhibit a strong positive correlation with the objective functions. Thus, the parameters A_{compasso} , the frequency ω and O_{Roll} strongly influence the impact force and displacement as well as A_{hPitch} influences the height to which the foot rises.

In the future it would be appropriate to apply the developed algorithm in a real biped robot and check if it can generate an efficient locomotion in real conditions.

Índice

1.	Introdução	1
1.1	Motivação e Objetivos	2
1.2	Estrutura da Tese	4
2.	Estado da Arte	6
2.1.	Métodos de Geração de Marcha	7
2.1.1.	Geradores Centrais de Padrões.....	7
2.1.2.	Modelo do Pêndulo Invertido	13
2.2	Locomoção Robótica	17
2.2.1	Grupos e Líderes.....	19
2.2.2	Robôs Humanóides	22
3.	Funções de Custo	27
3.1.	Controlo da Estabilidade	29
4.	Métodos de Otimização	33
4.1.	Otimização Multi-objetivo	35
4.1.1	Frente Pareto	37
4.1.2	Hipervolume.....	39
4.2	Computação Evolucionária.....	41
4.2.1	Algoritmos Genéticos	42
4.2.2	Outras Ferramentas de EC	48
4.2.3	NSGA.....	50
4.3	Métodos Não Evolucionários	54
4.3.1	<i>Reinforcement Learning</i>	54
4.3.2	<i>Particle Swarm Optimization</i>	57
5	Proposta de Otimização	59
5.1	Geradores Centrais de Padrões.....	59
5.2	Geração de Ritmos	60
5.3	Geração de Movimento	62
5.3.1	Primitivas de Movimento	63
5.4	Sistema de Otimização	69
5.4.1	Vetor de Parâmetros	71

5.4.2	Evolução por Fases	71
5.5	Determinação dos Objetivos	73
5.5.1	Centro de Pressão	73
5.5.2	Força	74
5.5.3	Deslocamento	77
5.5.4	Ground Clearance	77
5.6	Enquadramento dos Objetivos na Evolução por Fases	78
5.7	Tratamento de Restrições	79
5.8	Algoritmo de Otimização	80
6	Resultados e Discussão	82
6.1	Evolução da Otimização	84
6.2	Hipervolume	89
6.3	Soluções Extremas da Frente Pareto	90
6.4	Análise de Sensibilidade	94
6.4.1	Correlação entre Parâmetros e Funções Objetivo	94
6.4.2	Correlação entre Funções Objetivo	100
6.5	Principal Component Analysis	103
7	Conclusões e Futuros Trabalhos	106
	Bibliografia	109

Lista de Figuras

Figura 2. 1 - Número de artigos por ano que abordam como temática principal o uso de CPGs na robótica na base de dados IEEE Explore, desde 1990 até 2006.[11]	11
Figura 2. 2 - Ilustração do movimento da perna semelhante ao movimento de um pêndulo invertido. O círculo representa o centro de massa e as setas correspondem aos vetores da força aplicada no centro de massa e da velocidade deste. [21].....	14
Figura 2. 3 - Fase de duplo apoio. Transição de um passo para outro passo e redirecionamento da velocidade do centro de massa (à esquerda). Vetores da força de reação e da velocidade aplicada no COM na perna traseira (<i>trail</i>) e na perna dianteira (<i>lead</i>). [22].....	15
Figura 2. 4 - Robôs bípedes da Honda. [31].....	20
Figura 2. 5 - Primeiras versões dos robôs Darwin. Darwin 1.0 à esquerda e Darwin 2.0 à direita. [38]	25
Figura 2. 6 - Diferentes perspectivas do Darwin-OP com as respectivas dimensões. [39] 26	
Figura 3. 1 - Possíveis relações entre ZMP e COP para um pé não rígido. a) Locomoção dinâmica estável, b) Locomoção não estável onde o ZMP não existe e o ponto de atuação da força de reação do chão é o COP, enquanto o ponto onde $M_x = 0$ e $M_y = 0$ está fora do polígono de suporte (FZMP) – o sistema roda á volta da borda do pé, e c) equilíbrio dinâmico com a ponta do pé. [42]	30
Figura 4. 1 - Representação esquemática de operações básicas dos GA's. [71].....	43
Figura 4. 2 - Cálculo da <i>crowding-distance</i> . Os pontos marcados com círculos preenchidos são soluções da mesma frente não dominada. [80]	51
Figura 4. 3 - Esquema representativo do procedimento do NSGA-II. [80].....	53
5. 1 - Unidades geradoras, juntas correspondentes e acoplamentos intra-membros e inter-membros correspondentes. Acoplamento bilateral está representado pela seta bilateral e acoplamento unilateral está representado por setas unilaterais. [96]	61
5. 2 - Primitivas de movimento: A) movimento de <i>balancing</i> ; B) movimento de flexão; E) movimento de compasso. [96].....	63

5. 3 - Fluxograma representativo da proposta de otimização da locomoção.....	70
5. 4 - Diagrama representativo das três fases da estratégia de evolução.	72
5. 5 - Esquema representativo da localização dos sensores de força (T0, T1, T2 e T3) na planta do pé do Darwin e das duas diagonais entres eles	74
5. 6 - Perspetiva de baixo do Darwin. Especificação da posição dos sensores de força embebidos nos pés do robô e de algumas medidas específicas tidas em conta na <i>fitness</i>	76
5. 7 - Esquema representativo do <i>ground clearance</i>	77
6. 1 - Gráfico representativo da evolução por estágios dos indivíduos ao longo das gerações. Este gráfico representa uma média dos resultados obtidos em várias simulações	84
6. 2 - Gráfico representativo da evolução por estágios dos indivíduos ao longo das gerações. Este gráfico representa o exemplo com piores resultados obtidos entre várias simulações.....	85
6. 3 - Gráfico representativo da evolução por estágios dos indivíduos ao longo das gerações. Este gráfico representa o exemplo com melhores resultados obtidos entre várias simulações.....	86
6. 4 - Gráfico representativo da evolução por estágios dos indivíduos ao longo das gerações. Este gráfico representa os resultados obtidos para o Darwin antes de ser aplicada a nossa proposta de otimização.....	87
6. 5 - Fotografias tiradas ao Darwin durante a fase de <i>balancing</i> numa simulação efetuada no <i>webots</i>	88
6. 6 - Fotografias tiradas ao Darwin durante a fase de <i>stepping</i> numa simulação efetuada no <i>webots</i>	88
6. 7 - Fotografias tiradas ao Darwin durante a fase de <i>walking</i> numa simulação efetuada no <i>webots</i>	88
6. 8 - Média (linha azul) e desvio padrão (barras verticais) da medida de hipervolume em 10 simulações teste	89
6. 9 - Comparação da evolução do Pitch, do Roll e do COM (frontal e lateral) entre a solução <i>hand tuned</i> e as soluções não dominadas A, C, E e S. Os retângulos cincentos	

representam o instante em que o pé esquerdo está no chão e os retângulos brancos representam o instante em que o pé esquerdo não está no chão.....	92
6. 10 - Fotografias da simulação de marcha do Darwin no Webots. De cima para baixo: solução hand tuned, solução A (máximo deslocamento), solução E (mínima força) e solução C (máximo <i>ground clearance</i>).	93
6. 11 - Gráfico representativo da relação do parâmetro A_{compasso} com o aumento do ratio (<i>deslocamento</i>)/máximo(<i>deslocamento</i>) a cor azul e com o aumento do ratio (<i>força</i>)/mínimo(<i>força</i>) a cor verde	99
6. 12 - Gráfico representativo da relação do parâmetro ω com o aumento do ratio (<i>deslocamento</i>)/máximo(<i>deslocamento</i>) a cor azul e com o aumento do ratio (<i>força</i>)/mínimo(<i>força</i>) a cor verde	99
6. 13 - Gráfico representativo da relação do parâmetro O_{hRoll} com o aumento do ratio (<i>deslocamento</i>)/máximo(<i>deslocamento</i>) a cor azul e com o aumento do ratio (<i>força</i>)/mínimo(<i>força</i>) a cor verde	100
6. 14 - Gráfico tridimensional com valores das três funções objetivo obtidos a partir das soluções não dominadas da última população de uma simulação teste. Os círculos representam as soluções não dominadas, entre as quais encontram assinaladas as soluções A, C e E correspondentes aos extremos da frente Pareto (máximo <i>deslocamento</i> e <i>ground clearance</i> e mínima <i>força</i>).	102
6. 15 - Gráfico de PCA para as três funções objetivo. O <i>deslocamento</i> e o GC apresentam correlação semelhante com a <i>force</i>	104
6. 16 - Gráfico de PCA para as três funções objetivo. A <i>force</i> apresenta correlação superior com o <i>deslocamento</i> do que com o GC.	105

Lista de Tabelas

Tabela 2. 1 - Especificações (ano de criação, altura, peso, graus de liberdade e velocidade) dos robôs humanoides das séries E, P e Asimo. [31].....	20
Tabela 5. 1 - Valores assumidos por F dependendo das várias posturas dos pés.	75
Tabela 5. 2 - Limites das restrições dos parâmetros (amplitudes).....	79
Tabela 5. 3 - Limites das restrições dos parâmetros (offsets, σ e frequência).	79
Tabela 6. 1 - Valores de funções objetivo e respectivos parâmetros que correspondem a soluções extremas da frente Pareto, a uma solução hand tuned e a uma solução S não dominada.....	90
Tabela 6. 2 - Coeficientes de correlação de Pearson e valores p de significância entre os parâmetros e a função objetivo <i>Deslocamento</i> . Tanto para R como para P são apresentados os valores máximos, mínimos, média e desvio padrão calculados para 10 simulações. Os valores a ter em consideração encontram-se a negrito e sublinhados.	95
Tabela 6. 3 - Coeficientes de correlação de Pearson e valores p de significância entre os parâmetros e a função objetivo <i>Força</i> . Tanto para R como para P são apresentados os valores máximos, mínimos, média e desvio padrão calculados para 10 simulações. Os valores a ter em consideração encontram-se a negrito e sublinhados.	96
Tabela 6. 4 - Coeficientes de correlação de Pearson e valores p de significância entre os parâmetros e a função objetivo <i>Ground Clearance</i> . Tanto para R como para P são apresentados os valores máximos, mínimos, média e desvio padrão calculados para 10 simulações. Os valores a ter em consideração encontram-se a negrito e sublinhados	97
Tabela 6. 5 - Coeficientes de correlação de Pearson e valores p de significância entre as função objetivo.....	100
Tabela 6. 6 - Importância das várias componentes no estudo de PCA.	103

1. Introdução

A locomoção robótica corresponde a todo o conjunto de métodos que permitem aos sistemas robóticos mover-se no seu ambiente. É, deste modo, a base fundamental para a performance de qualquer robô móvel.

Esta dissertação encontra-se inserida no âmbito do Mestrado Integrado em Engenharia Biomédica, e prende-se com o ramo da robótica, mais precisamente com a otimização da locomoção bípede.

A locomoção bípede corresponde ao resultado da relação temporal e espacial entre todas as partes móveis do robô. Neste contexto, o problema da otimização remete para a procura da marcha ideal no sentido de determinar a posição e velocidade ótima para cada junta a cada instante. A determinação de padrões de marcha vai afetar diretamente a estabilidade do robô e tem como objetivo maximizar as métricas de performance. [1]

Neste contexto, o problema de otimização tem um cariz multi-objetivo pois resulta na obtenção de várias soluções que correspondem aos valores que melhor satisfazem vários objetivos. Estas soluções são conhecidas como soluções não dominadas e correspondem a soluções, dentro do espaço abrangido pelas funções objetivo, para as quais não é possível melhorar simultaneamente os valores de todas as funções objetivo, ou seja, a melhoria de um ou mais objetivos requer uma menor performance em um ou mais objetivos. O conjunto de soluções não dominadas constitui a frente Pareto a qual traduz a gama de extensão dos resultados obtidos. [2]

A otimização de sistemas de locomoção requer uma formulação ponderada do problema e conseqüentemente, é necessário definir métodos de geração de marcha, especificar parâmetros e objetivos, e traduzi-los em funções que vão ser minimizadas ou maximizadas, e por fim, definir o tipo de algoritmo que vai servir como ferramenta de otimização.

A proposta de otimização apresentada nesta dissertação foi aplicada no robô humanoide Darwin-OP cujas características são especificadas no capítulo 2. Os testes foram realizados na plataforma de simulação *Webots*.

1.1 Motivação e Objetivos

Nas últimas décadas, têm sido explorados novos métodos e modelos bio-inspirados para a otimização da marcha através do estudo e compreensão dos princípios do controlo motor do ser humano e dos animais para aplicação em robôs autónomos, conferindo-lhes capacidade de locomoção e de aprendizagem autónoma.

A descoberta ou otimização de novos algoritmos e métodos de geração de locomoção revela-se de extrema importância no âmbito da obtenção de sistemas móveis ou de apoio à locomoção eficientes e estáveis. Geralmente, o elevado número de parâmetros inerente à locomoção robótica e a falta de conhecimento sobre o comportamento dinâmico do robô e a sua relação com o ambiente à sua volta torna a modelação dos parâmetros à mão uma missão muito difícil. Assim, este trabalho remete para uma estratégia capaz de ultrapassar essas dificuldades possibilitando ao robô pesquisar a gama de valores que melhor respeitam os objetivos definidos para a otimização da marcha.

A pesquisa de novas soluções e métodos de otimização na área da locomoção bípede pode corresponder a um passo inicial na vasta investigação que tem sido feita a nível de técnicas de reabilitação. Atualmente verifica-se um aumento da incidência de pessoas que recorrem a reabilitação de modo a recuperar a funcionalidade dos membros inferiores. Por exemplo, existem em média um milhão de pessoas por ano no espaço da União Europeia que sofrem um Acidente Vascular Cerebral (AVC) e cerca de 50% das vítimas desta patologia sobrevivem com perdas de controlo motor ou até mesmo amputações de membros. A recuperação de funções dos membros inferiores é um processo complexo que envolve intervenção terapêutica. Neste contexto, novas técnicas de reabilitação com recurso a dispositivos robóticos têm sido empregues, nomeadamente exoesqueletos, ortóteses e próteses robóticas. [3]

Deste modo, ao fornecer resultados para o desenvolvimento de estratégias de otimização no âmbito da geração de marcha eficiente, este trabalho representa uma contribuição preciosa para no futuro tornar possível a aplicação de algoritmos de otimização multi-objetivo que permitam gerar padrões de marcha com elevada qualidade traduzindo-se numa substancial melhoria da qualidade de vida para os pacientes.

O estudo realizado nesta dissertação apresenta-se ainda de grande utilidade para o desenvolvimento de sistemas robóticos de assistência em hospitais ou outro tipo de unidades hospitalares onde certamente existem pacientes com problemas de mobilidade derivada de problemas como a velhice, obesidade ou patologias com consequências no controlo motor. A logística de muitos hospitais não permite por vezes dar o devido apoio a essas pessoas. Neste contexto, o recurso a sistemas robóticos móveis dentro dos hospitais é uma perspetiva que tem vindo a ganhar força nos últimos anos e pode vir a resolver o problema de falta de auxílio a pessoas que não consigam andar sem algum tipo de apoio. [4] [5]

No que diz respeito aos objetivos principais desta dissertação é de salientar a pretensão de criar uma metodologia de otimização de marcha num robô simulado Darwin-OP no sentido deste conseguir:

- Deslocar-se a maior distância possível num determinado intervalo de tempo;
- Diminuir a força de impacto dos pés quando estes entram em contato com o chão;
- Colocar os pés numa posição que favoreça a estabilidade do robô;
- Aumentar a altura correspondente ao deslocamento vertical do pé durante o ciclo da marcha no sentido de não arrastar os pés.

Deste modo, pretende-se com este trabalho determinar uma frente ótima Pareto, cujas soluções, para além de serem as mais robustas, assegurem também uma boa diversidade ao longo desta frente. Além disso, exige-se que seja feita uma análise de sensibilidade, à medida que são feitos os testes, relativamente aos parâmetros e objetivos definidos e daí adquirir resultados e conhecimentos que permitam a reformulação e adaptação das funções que se pretende maximizar ou minimizar.

1.2 Estrutura da Tese

Esta dissertação divide-se em sete capítulos diferentes nos quais são descritos diferentes tipos de conteúdo que dizem respeito às várias etapas da elaboração do trabalho realizado ao longo do ano.

Deste modo, neste capítulo fez-se o enquadramento do trabalho realizado e uma apresentação dos objetivos propostos.

No capítulo 2 é feita uma descrição dos principais métodos de geração de marcha. É ainda feita uma descrição dos robôs humanoides mais evoluídos do mercado e são enumerados os grupos e líderes mundiais que se têm destacado no avanço de robôs móveis.

No capítulo 3 é descrito o estado de arte para as funções objetivo usadas na otimização da locomoção bípede. Neste âmbito é de referir um certo destaque para o controlo da estabilidade, o qual foi mais pormenorizado.

No capítulo 4 são esmiuçados os principais métodos de otimização existentes na área da locomoção bípede.

No capítulo 5 é descrita a estratégia de otimização elaborada para este trabalho, referindo qual o método de geração de movimento e algoritmo de otimização utilizado. É ainda especificado os objetivos e funções inerentes assim como descrito o tratamento de restrições na estratégia de otimização.

No capítulo 6, inicialmente, é detalhado o protocolo experimental fornecendo detalhes sobre as condições em que foi levada a cabo a proposta descrita no capítulo 5. Posteriormente, são apresentados os resultados obtidos os quais são sujeitos a uma análise de sensibilidade no sentido de determinar importantes correlações e redundâncias entre parâmetros e funções objetivo. É também feito um estudo de PCA sobre as funções objetivo no sentido de determinar as relações entre elas. É feita uma análise da frente Pareto assim como do hipervolume. É ainda estudada a evolução por estágios da estratégia utilizada.

Por fim, no capítulo 7 é realizada uma reflexão em tom de conclusão e referidos possíveis futuros trabalhos no seguimento do que foi feito nesta dissertação.

No âmbito desta dissertação, foi elaborado e submetido um artigo que se encontra em processo de avaliação o qual se encontra referido com a referência [6]:

- M. Oliveira, V. Matos, C. Santos, L. Costa e P. Barbosa, “Multi-objective Parameter CPG Optimization for Gait Generation of a Biped Robot,” *2013 IEEE International Conference on Robotics and Automation*, 2012 (submetido).

2. Estado da Arte

Uma vez que o ambiente onde os humanos operam é imprevisível, o robô tem de gerar em tempo real a marcha adequada baseada na informação recebida pelo sistema sensorial. O robô deve caminhar com diferentes comprimentos de passo, ultrapassar obstáculos, subir degraus, etc. [7]

Os robôs que usam locomoção com pernas possuem maior mobilidade em ambientes irregulares e não estruturados do que os robôs que se movimentam por rodas. Essa qualidade deve-se ao facto de poderem usar pontos de apoio discretos mais adequados para ultrapassar obstáculos ou subir e descer escadas.

Deste modo, a marcha é um movimento cíclico e periódico das juntas de um robô com pernas, implicando a sequenciação ou coordenação das pernas de modo a conseguir uma locomoção viável. [1, 8]

A geração de marcha é um problema de otimização multi-objetivo e multi-condicionado, representando um grande desafio para robôs com membros pois estes possuem sistemas de locomoção com múltiplos graus de liberdade e uma estrutura mecânica variável, resultando num número elevado de parâmetros que têm de ser estabelecidos. [1]

2.1. Métodos de Geração de Marcha

Fornecer boas capacidades locomotoras aos robôs é de primária importância no sentido de projetar robôs que possam levar a cabo tarefas úteis em diversos ambientes. A relevância da locomoção quer para a biologia quer para a robótica levou a múltiplas interações interessantes entre estes dois campos. As interações têm acontecido maioritariamente numa direção, com a robótica a inspirar-se a partir da biologia em termos de morfologias, modelos de locomoção, e mecanismos de controlo. Por outro lado, a robótica começa a dar algum retorno, e já começam a ser usados robôs como ferramentas científicas para testar hipóteses biológicas. [9] [10] [2]

2.1.1. Geradores Centrais de Padrões

Nos sistemas biológicos, o mecanismo gerador de ritmo denominado por Gerador Central de padrões (CPG) está envolvido em várias atividades rítmicas, tais como a locomoção, respiração, batida cardíaca, entre outras. A geração de locomoção através de CPGs atrai muita atenção pois quer o sistema nervoso quer o sistema músculo-esquelético interagem um com o outro e são atingidas dinâmicas oscilatórias com redes de CPGs. [10]

No início do século passado, Brown et al. sugeriu que os ritmos são gerados centralmente, isto é, por redes neuronais que não requerem *inputs* da periferia (como por exemplo, neurónios sensoriais) para gerar comportamentos cíclicos. Na altura propôs um modelo conceptual no qual duas populações de neurónios que são acoplados mutuamente com conexões inibitórias e que possuem um mecanismo de fadiga produzem atividade rítmica. Atualmente tornou-se evidente que os ritmos são gerados centralmente sem requerer informação sensorial. Todavia este feedback sensorial, embora não seja necessário para gerar ritmos, assume um papel importante em modelar os padrões rítmicos. Isto é fundamental para manter os CPGs e os movimentos do corpo coordenados. [11] [12] [13]

Em suma, o CPG é um circuito neuronal encontrado quer em animais invertebrados quer em vertebrados que podem produzir padrões rítmicos de atividade

neuronal sem receber *inputs* rítmicos. O termo central indica que o feedback sensorial (proveniente dos sistema nervoso periférico) não é necessário para gerar ritmos.

Os CPGs apresentam diversas propriedades interessantes tais como controlo distribuído, habilidade para lidar com redundâncias, e permite a modulação da locomoção por simples sinais de controlo. Estas propriedades, quando transferidas para modelos matemáticos, tornam os CPGs alicerces interessantes para controladores de locomoção em robôs. Isto traduziu-se num aumento exponencial de estudos para a introdução de modelos com CPGs para a locomoção robótica. Estes estudos incluem modelos conetores, mapas vetoriais e sistemas de osciladores acoplados. Virtualmente, todas as implementações envolvem conjuntos de equações diferenciais acopladas que são numericamente integradas (num microcontrolador ou processador). Provavelmente as únicas exceções são os CPGs que são diretamente executados em *hardware*, isto é, num *chip* ou outro dispositivo eletrónico.

Existem diversas propriedades que tornam os modelos de CPGs uma das alternativas mais adequadas e promissoras para a locomoção robótica: [9] [11] [14]

- O propósito dos modelos de CPG é exibir comportamentos cíclicos, isto é, produzir padrões rítmicos estáveis. Quando isso acontece, o sistema rapidamente retorna ao seu comportamento rítmico normal após perturbações transientes das variáveis de estado. Isto fornece robustez contra perturbações.
- Os CPGs são adequados para implementação distribuída, a qual é importante para robôs modulares e robôs reconfiguráveis.
- Os modelos de CPGs possuem tipicamente poucos parâmetros de controlo que permitem modulação da locomoção, por exemplo a velocidade e direção ou ainda o tipo de marcha. Um modelo de CPG devidamente implementado reduz a dimensionalidade do problema de controlo de tal modo que controladores de elevado nível (ou algoritmos de aprendizagem) não precisam de produzir diretamente comandos motores multidimensionais precisando apenas de sinais de controlo. Relacionado com este fator, modelos de CPG produzem geralmente modulações suaves

das trajetórias produzidas mesmo quando os parâmetros de controlo são alterados de forma abruptas, uma vez que equações diferenciais atuam tipicamente com filtros de primeira ou segunda ordem. Esta propriedade é útil para a geração *online* de trajetórias a qual evita possíveis danos nos motores ou caixa de velocidades devidos a alterações abruptas nos comandos motores.

- Os CPGs são ideais para integrar sinais de feedback sensorial (o qual pode ser adicionado como termos acoplados em equações diferenciais).
- Os modelos de CPGs oferecem, geralmente, um bom substrato para a aprendizagem e otimização de algoritmos.

Ainda não existe uma metodologia de implementação de CPGs na locomoção robótica bem definida, e os investigadores têm explorado um vasto leque de abordagens inclusive *designs* baseados em teorias de sistemas dinâmicos e algoritmos de aprendizagem e otimização. Todavia, alguns fatores devem ser tidos em conta quando se constrói um modelo de CPGs neste âmbito. Primeiramente deve-se definir a arquitetura geral do CPG. Isto inclui o tipo e número de osciladores ou neurónios. Num robô, isto também implica escolher entre controlo de posição (os outputs do CPG são os ângulos desejados para as juntas que são fornecidos a um controlador de feedback) ou controlo de torque (os outputs dos CPG controlam diretamente o torque produzido pelos motores). O tipo e topologia dos acoplamentos também devem ser bem definidos uma vez que isto vai determinar as condições para a sincronização entre osciladores e as marchas resultantes, isto é, as relações de fase estáveis entre osciladores. Posteriormente deve-se também ter em conta as formas de onda. Este fator vai determinar que trajetórias são atualmente executadas por cada junta durante o ciclo. As formas de onda são claramente dependentes no formato do ciclo limite produzido pelo oscilador escolhido, embora possa ser transformado pela adição de filtros. Por fim, deve-se ter em conta o efeito dos sinais de *input*, ou seja, como podem os parâmetros de controlo modular variáveis como a frequência, amplitude, atrasos de fase, ou formas de onda (por exemplo, ajustar independentemente fases de *swing* e *stance*), assim como se deve ter em conta o efeito dos sinais de feedback provenientes

do corpo, os quais podem afetar a atividade do CPG, como por exemplo acelerar ou desacelerar depender das condições das condições do ambiente. [9] [2] [11] [15]

A aplicação de CPGs em algoritmos de otimização depende também da natureza do próprio método de otimização. Algoritmos de aprendizagem e de otimização podem ser usados de duas formas diferentes. As abordagens podem-se dividir em duas categorias: aprendizagem supervisionada e aprendizagem não supervisionada. A primeira pode ser aplicada quando o padrão de ritmos desejado que o CPG tem de produzir é conhecido. O padrão desejado pode então ser usado para definir a função de erro explícita para ser minimizada. Tais técnicas podem, as vezes, serem usadas para desenhar CGPs, embora estejam restringidas a situações onde padrões adequados são avaliados. Exemplos de técnicas incluem algoritmos de aprendizagem *gradient-descent* para redes neuronais recorrentes, aprendizagem para campos vetoriais, e algoritmos de aprendizagem estatística para sistemas dinâmicos.

Técnicas de aprendizagem não supervisionada são usadas quando o comportamento desejado do CPG não está definido por um padrão específico pretendido tal como acontece na aprendizagem supervisionada, mas por um critério de performance de alto nível, como por exemplo caminhar o mais rápido possível. Entre as técnicas de aprendizagem não supervisionadas, algoritmos de otimização estocásticos baseados em populações tais como algoritmos evolucionários tem sido extensivamente usados no *design* de modelos de CPG. Outra abordagem não supervisionada consiste em algoritmos de aprendizagem por reforço (*reinforcement learning*) e de otimização heurística tal como o método de Powell. [11]

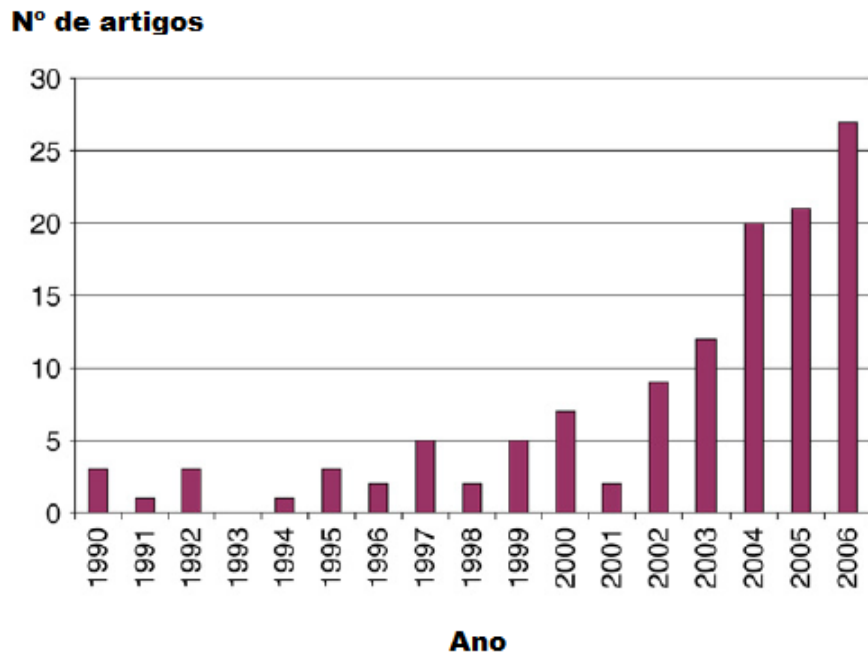


Figura 2. 1 - Número de artigos por ano que abordam como temática principal o uso de CPGs na robótica na base de dados IEEE Explore, desde 1990 até 2006. [11]

Têm sido vários os autores a explorar o recurso a CPGs para a geração de locomoção em robótica. O gráfico da figura 2.1 demonstra o crescimento de interesse e de produção científica que tem havido em relação aos CPGs nas últimas décadas. Neste contexto, é de destacar Matsuoka et al. que propôs uma rede de CPG para locomoção bípede em que cada junta possui um oscilador com dois neurónios, um extensor e outro fletor. Propôs um modelo matemático, e forneceu algumas condições matemáticas para redes de inibição mútuas representadas por um modelo neuronal contínuo para gerar oscilação. Taga et al. propôs um princípio de controlo adaptativo do sistema de locomoção, onde os sistemas nervoso, músculo-esquelético e sensorial tiveram de cooperar para se adaptar a ambientes imprevisíveis. Os resultados da marcha bípede resultante na simulação 2-D mostraram robustez contra a variação de declives. Miyakoshi et al. expandiram a simulação 2-D a movimento 3-D, e a simulação da marcha foi realizada em simulações demonstrando estabilidade. Hase et al. propuseram um método de otimização de parâmetros usando um método conhecido por Algoritmo Genético (GA), e analisaram o movimento 3-D com um sistema músculo-

esquelético preciso e um sistema nervoso hierárquico. O controle de movimento usando osciladores não lineares foram também propostos e verificados em robôs quadrúpedes (Kimura et al., Tsujita et al.) e robôs hexápodes (Cruse et al., Akimoto et al.). [16] [17] [18] [19] [20]

2.1.2. Modelo do Pêndulo Invertido

Têm sido propostos muitos métodos para o controle da locomoção de robôs humanoides. Muitos investigadores têm recorrido ao modelo do pêndulo invertido para gerar algoritmos de controle para locomoção. O pêndulo invertido corresponde a uma classe de sistemas sub-atuados, não lineares de elevada ordem, que são caracterizados por um ponto de equilíbrio instável. Pode ser usado na análise e controle da estabilidade de sistemas similares tais como: robôs móveis com rodas com um centro de gravidade alto, manipuladores robóticos e robôs bípedes, entre outros. [21]

A marcha é muitas vezes vista como o movimento acoplado de dois pêndulos, uma vez que a postura da perna se comporta como um pêndulo invertido em torno do pé de apoio, e a perna que faz o *swing* comporta-se também como um pêndulo normal em torno da anca. A analogia é usada para explicar a conservação da energia durante a marcha. A mecânica da marcha requer a transição entre fases semelhantes às de um pêndulo. Ao contrário de uma roda, um pêndulo invertido apenas pode transportar o centro de massa (COM) uma distância limitada. Os humanos continuam a sua marcha através da transferência da postura pendular de uma perna para a outra perna. Kuo et al. verificou que parte significativa da energia metabólica gasta é consequência desta transição. Tem de se executar trabalho mecânico positivo e negativo no COM de modo a conseguir executar essas transições, e este trabalho requer um custo metabólico proporcional. Este custo é uma consequência inevitável do comportamento do pêndulo invertido. A figura 2.2 retrata o movimento da perna semelhante ao movimento de um pêndulo invertido. [22] [23]

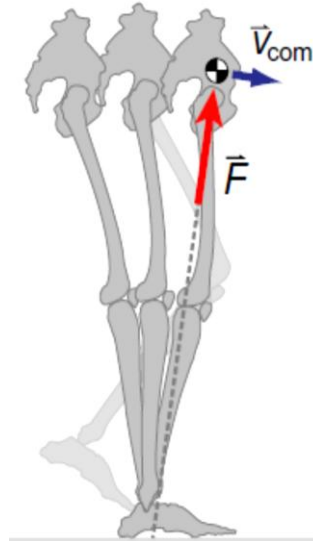


Figura 2. 2 - Ilustração do movimento da perna semelhante ao movimento de um pêndulo invertido. O círculo representa o centro de massa e as setas correspondem aos vetores da força aplicada no centro de massa e da velocidade deste. [22]

Uma perna comportando-se como um pêndulo invertido rígido, durante o suporte com apenas uma perna, conserva a energia mecânica, e conseqüentemente, não é necessário trabalho para mover o COM. A perna pode também ser mantida em máxima extensão com força muscular mínima de modo a que as fases de suporte com apenas uma perna não implique custos em termos de trabalho ou de força muscular. Todavia a fase de apoio único apenas dura a distância de um passo e, mesmo que o seguinte passo desempenhe igualmente o pêndulo invertido e haja também conservação de energia, a transição entre passos, no qual um pêndulo para e outro se inicia, implica trabalho mecânico. Esta transição implica a redireção da velocidade do COM e simultaneamente trabalho negativo e positivo por parte das duas pernas (figura 2.3). No fim de um passo, o COM é movido para a frente, mas com uma componente de velocidade para baixo, tal como o arco pendular o descreve. No início do seguinte passo, o COM tem de ser redirecionado para se mover para cima, de modo a seguir o arco prescrito pela perna dianteira. Para manter a velocidade da marcha estável, a magnitude da velocidade do COM tem de ser a mesma no início e no fim da fase de apoio único. Ainda assim, mudar a direção da velocidade do COM requer força, produzida separadamente pelas duas pernas, e direcionada ao longo de cada perna.

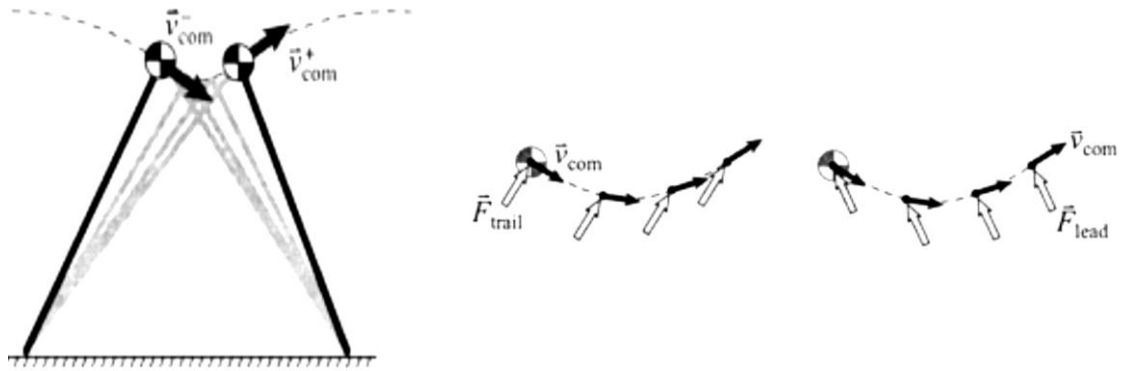


Figura 2. 3 - Fase de duplo apoio. Transição de um passo para outro passo e redirecionamento da velocidade do centro de massa (à esquerda). Vetores da força de reação e da velocidade aplicada no COM na perna traseira (*trail*) e na perna dianteira (*lead*). [23]

A perna traseira desempenha um trabalho positivo no COM e a perna dianteira executa um trabalho negativo (figura 2.3). A taxa de trabalho é igual ao produto escalar entre os vetores velocidade e força. A força da perna traseira está direcionada num ângulo agudo com a velocidade do COM, e conseqüentemente, apresenta um trabalho positivo. A perna dianteira está direcionada num ângulo obtuso com a velocidade do COM, e logicamente o trabalho é negativo. A transição entre passos é ótima quando o trabalho exercido por ambas as pernas possui igual magnitude. Teoricamente, torna-se mais custoso se as pernas não executarem a mesma quantidade de trabalho durante a fase de apoio duplo por causa do trabalho adicional necessário durante a fase de apoio único para manter a velocidade da marcha estável. Por outro lado, o ideal é minimizar o tempo e o deslocamento da transição entre passos. Um maior deslocamento do COM durante o redirecionamento implica mais trabalho por parte de cada perna. [24] [23] [22]

Kajita et al. sugeriu o Modo de Pêndulo Invertido Linear (LIPM), o qual assume uma massa concentrada no torso e negligencia todas as outras distribuições de massa. Kajita aplicou controlo multi-condicionado de modo a que o corpo do robô se movimente numa linha reta e rode a uma velocidade angular constante. Isto torna a dinâmica do centro de massa completamente linear, e passa a não ser necessário binário (*torque*) no tornozelo e o ZMP permanece no tornozelo. O torque do tornozelo é usado para controlar o movimento horizontal do corpo para incluir o efeito das massas das pernas e lidar com perturbações externas.

Park et al. propôs um modelo de pêndulo invertido com compensação gravítica (GCIPM). Ao invés de assumir apenas uma massa como no método LIPM, o modelo do robô consiste em duas massas: uma para a *base link* e para a perna de suporte, e a outra massa é para a outra perna. Usando esta técnica, eles desenvolveram um método de trajetórias *online* que aumenta a estabilidade da locomoção, baseado na equação do ZMP e na informação sensorial do ZMP. Park et al. estendeu, posteriormente, esta estratégia para ser usada também durante a fase de duplo apoio. [25]

Albert et al. apresentou um modelo de pêndulo invertido com múltiplas massas (MMIPM). O utilizador determina o movimento do pé e da perna que faz o *swing* e as trajetórias restantes são então calculadas iterativamente. Simulações e medidas reais do ZMP pelos sensores de força mostraram que este modelo conduz a uma maior estabilidade no que diz respeito ao ZMP, resultado do fato do modelo ser mais preciso e realista. [26] [27]

2.2 Locomoção Robótica

Os termos robô móvel descrevem um sistema robótico capaz de desempenhar tarefas em sítios diferentes e que consiste numa plataforma movida por elementos de locomoção. A escolha do sistema de locomoção depende primeiramente do ambiente no qual o robô vai operar. Este pode ser aéreo, aquático ou terrestre. Nos ambientes aquáticos e aéreos, os sistemas locomotores são geralmente hélices ou parafusos, embora em solos oceânicos também sejam usadas pernas robóticas. A escolha de um sistema locomotor em ambiente terrestre é mais complexa devida á variedade inerente, recorrendo quer a pernas quer a pneus. A mobilidade fornece robôs com uma capacidade de operação melhorada e abre uma nova série de áreas de investigação. [28]

Na época em que os robôs industriais foram introduzidos no processo de produção, robôs móveis foram instalados na indústria. Isto aconteceu no fim da década de 60, e os robôs eram maioritariamente veículos automatizados guiados (AGVs), veículos que transportam ferramentas e que seguem uma trajetória pré-definida. O processo de locomoção autónoma envolve diversos estágios tais como perceção do ambiente e geração do movimento. Em ambientes estruturados, este processo de perceção possibilita a geração de mapas ou modelos do mundo que são usados para a localização do robô e para o planeamento do movimento. Em ambientes não estruturados, o robô já precisa de aprender como adaptar a sua locomoção, sendo esta uma das áreas principais de estudo no âmbito da robótica, onde aprendizagem, raciocínio e resolução de problemas vêm juntos. [28] [29]

O processo de localização permite ao robô saber onde está relativamente ao ambiente a cada momento. Neste sentido, são usados sensores que permitem que medições relativas ao estado do robô e ao ambiente que o rodeia sejam feitas. Todavia estes sensores acumulam erros e fornecem medições com ruído inerente. Neste contexto, muitos investigadores ligados a esta temática, tentam melhorar o processo de estimativa de localização por meio da integração das medições obtidas pelos vários tipos de sensores através do uso de filtros Kalman.

A localização pode ser local ou global, sendo a local uma solução mais fácil, onde o robô, de forma incrementada, corrige a sua posição relativa a uma localização

inicial, enquanto na posição global, a posição inicial não é necessária. Além disso, este processo de localização pode ser baseado na identificação sensorial de pontos de referência no ambiente cuja localização é bem conhecida, ou pode ser baseada em mapas e modelos do ambiente e identificar elementos característicos do ambiente mapeado. Os algoritmos de localização, na literatura, geralmente derivam todos do filtro de Bayes, uma equação recursiva que permite estimar a postura do robô a partir do modelo perceptual e do modelo do movimento. [28] [30]

2.2.1 Grupos e Líderes

Nas últimas 3 décadas tem havido um esforço a nível mundial para conseguir desenvolver métodos de locomoção para os robôs quer a nível de locomoção sobre rodas quer a nível de locomoção bípede, quadrúpede e hexapóde.

Categorizando os líderes e grupos de investigadores pelos tipos de locomoção, nomeadamente na locomoção por rodas, é de referir a empresa iRobot, a NASA, e organização do DARPA *Grand Challenge* e autores de investigações como Poultier e os investigadores de *Swiss Federal Institute of Technology Zurich*, entre outros.

Relativamente a locomoção com pernas, o Japão é certamente o país onde a locomoção tem recebido mais atenção e onde tem mais historial. A equipa do professor Kato da Universidade de Waseda no Japão tem desenvolvido uma família inteira de robôs (*Waseda legged robots*) com pernas durante as últimas décadas. Começaram com o WL-1 em 1967 e foram dando seguimento e, em 1972 o WL-5 apresentava já uma marcha bípede automática e capaz de variar a direção, usando um mini-computador como controlador. Em 1983, o WL-10R foi desenvolvido com mais um grau de liberdade no eixo *yaw* na junta da bacia. Marcha plana assim como marcha lateral, rodar, caminhar para a frente e para trás foram ações adquiridas por este sistema humanoide onde foram introduzidos servo-atuadores rotativos. A pesquisa em robôs antropomórficos inteligentes foi iniciada em 1970 com o desenvolvimento do WABOT (Waseda Robot). O WABOT-1 era capaz de comunicar em japonês e de medir distâncias e direções usando recetores externos, orelhas, boca e olhos artificiais. Era também capaz de agarrar objetos e transportá-los com as mãos. Os resultados excelentes provenientes da Universidade de Waseda levaram à criação do *Humanoid Robotics Institute* em 2000, o qual tem sido uma das principais fontes de desenvolvimento na área. [31]

No Japão, é de destacar também os engenheiros da marca Honda que desde 1986 têm conseguido avanços tecnológicos nesta área bastante significativos, primeiro com o EO que teve seguimento até ao E6 e depois com a série P (P1, P2 e P3). Por fim, criaram um dos robôs bípedes mais inteligentes denominado por ASIMO (Advanced Step in Innovative Mobility). As especificações da série de humanoides da Honda são apresentadas na tabela 1. [32]

Tabela 2. 1 - Especificações (ano de criação, altura, peso, graus de liberdade e velocidade) dos robôs humanoides das séries E, P e Asimo. [32]

Series	Year	Weight (kg)	Height (cm)	DOF	Walking (Kmh ⁻¹)	Running (Kmh ⁻¹)
E0	1986	016.5	101.3	06	5.00 seconds between steps	No
E1	1987	072.0	128.8	12	0.25	
E2	1989	067.7	132.0	12	1.20	
E3	1991	086.0	136.3	12	3.00	
E4	1991	150.0	159.5	12	4.70	
E5	1992	150.0	170.0	12	Slow	
E6	1993	150.0	174.3	12	Slow	
P1	1993	175.0	191.5	30	Slow	
P2	1996	210.0	182.0	30	2.00	
P3	1997	130.0	160.0	28	2.00	
ASIMO	2000	052.0	120.0	26	1.60	6.0 (straight) 5.0 (circling)
	2005	054.0	130.0	34	2.70	

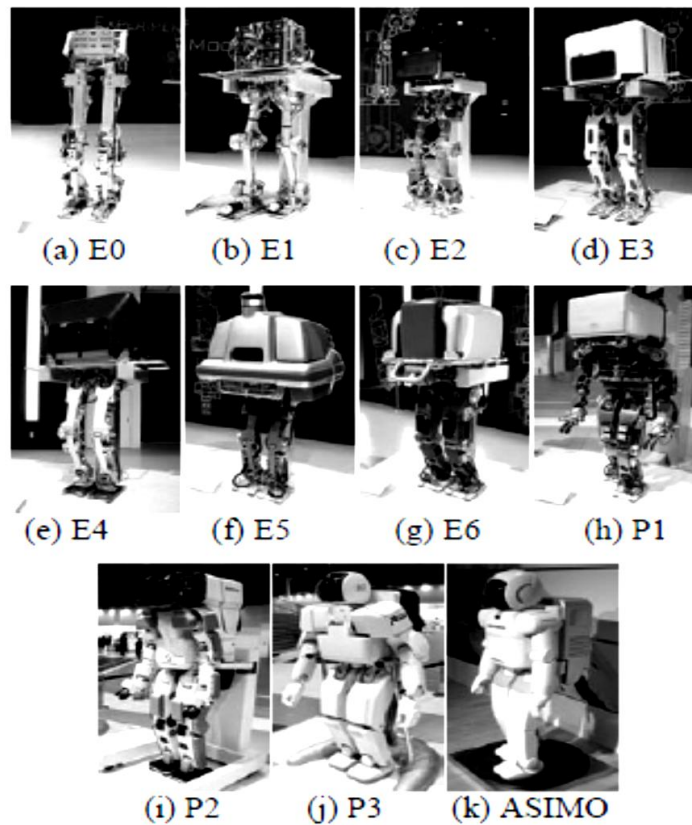


Figura 2. 4 - Robôs bípedes da Honda. [32]

Outra referência na locomoção bípede proveniente do Japão é a Universidade de Tóquio, a qual se começou a destacar coma criação do sistema humanoide chamado SAIKO em 1997, o qual possuía um custo reduzido. Criaram também o conhecido H6 no ano de 2000 o qual se apresentou como uma plataforma de pesquisa na integração percepção-ação em sistemas humanoides. Por outro lado, sob a custódia do ministério da economia, comércio e indústria japonês foi criada uma série de robôs que ainda hoje são uma referência – Humanoid Robot Project (HRP). Por fim, é de referir também a empresa mundialmente conhecida Fujitsu que também tem tido um papel ativo na locomoção robótica destacando-se com a série de robôs HOAP.

A nível mundial há muitas outras referências a destacar. Em França, tem sido apresentado excelentes resultados, nomeadamente na Universidade de Estrasburgo e o seu laboratório LSIT, o instituto nacional INRIA, os laboratórios Rhone-Alpes e LMS Poitiers, o CNRS em Grenoble (conhecido pelo seu projeto RABBIT) e o laboratório LIRIS na Universidade de Versailles. Recentemente, os franceses da Aldebaran Robotics (Paris) tiveram bastante sucesso com a criação do Humanóide NAO.

Outros sítios de referência a ter em conta são: a Universidade Técnica de Berlim, a Universidade de Queensland na Austrália, a Universidade Nacional da Singapura, a Universidade de Génova em Itália. A Coreia também tem contribuído com a série de robôs KHR desenvolvidos no Korea Advanced Institute of Science and Technology (KAIST). O KHR-3 tornou-se mundialmente conhecido sob o nome de HUBO. Os Estados Unidos são também uma fonte de desenvolvimento na área de locomoção destacando-se o Leg Laboratory do MIT. Outras referências nos EUA são a Universidade de Virginia e a Universidade de Pennsylvania. Estas duas universidades tiveram uma contribuição preciosa, juntamente com a empresa coreana ROBOTIS, para a criação de um dos humanoides com mais sucesso na atualidade – o Darwin-OP.

No que diz respeito a locomoção quadrúpede, durante os últimos anos, a Liga RoboCup Four-Legged tornou-se uma boa plataforma para diferentes abordagens de optimização de marchas. A razão para tal é provavelmente a plataforma padrão AIBO da SONY que permite aos seus investigadores comparar facilmente e aplicar os seus resultados.

A nível da locomoção hexapóde é de referir, entre outros o grupo de investigação da Universidade de Chiba no Japão que criou a série de robots COMET; o

projeto iniciado por Espenschied – Robot I, o qual teve continuidade com Robot II e Robot III; os projectos da NASA como por exemplo a saga de robots ATHLETE; e o Instituto Canadiao de Robótica e Sistemas Inteligentes (IRIS) que promoveu a criação de robots hexapódes anfíbios denominados de AQUA. [32] [33] [34] [35]

2.2.2 Robôs Humanóides

O desenvolvimento de robôs humanóides é um dos tópicos mais abordados da atualidade na robótica. O enorme sucesso gerado pelos robôs humanóides da HONDA despoletou o interesse a nível mundial nesta área. Desde a criação do segundo protótipo da HONDA, o P2, em 1996, têm sido desenvolvidos imensos humanóides.

Na universidade técnica de Munique têm vindo a desenvolver um robô bípede humanóide com 22 DOF's designado por LOLA e o qual é capaz de movimentar-se á semelhança dos humanos.

No Korea Advanced Institute of Science and Technology desenvolveram também um robô humanóide designado por HUBO com 41 DOF. Neste é adoptado uma arquitetura de controlo através de CAN (*Controller Area Network*). A sua marcha atinge os 1.25 km/h.

A empresa francesa Aldebaran-Robotics obteve reconhecimento com a criação do humanóide NAO. Com apenas 4.5 Kg e 0.57 m, este robô possui um índice de massa corporal de 13.5 [kg/m²], sendo mais leve que a maioria dos outros humanóides que possuem uma altura semelhante. Possui um total de 25 graus de liberdade, 11 para a parte inferior (pernas e pélvis) e 14 para a parte superior (tronco, braços e cabeça). Possui algumas características inovadoras tais como: a pélvis é composta por duas juntas acopladas que são orientadas por apenas um motor; a motorização usa módulos atuadores que incluem juntas universais personalizadas, mecanismo de engrenagem personalizado e *servoboard* personalizada, e a arquitetura do *software* assenta num sistema modular e distribuído que é fácil de gerir. O humanóide NAO é uma plataforma aberta e fácil de manusear na qual o utilizador pode mudar o *software* embebido ou apenas adicionar algumas aplicações para que o robô adote comportamentos específicos. [36]

Os engenheiros da Honda criaram uma última versão do ASIMO com 34 graus de liberdade que contribuam para que ele caminhe e desempenhe funções de forma muito semelhante à do ser humano. Construído com materiais de peso leve, como a estrutura de liga de magnésio, combinado com computadores poderosos e 34 servo motores, ajuda o ASIMO a mover-se suavemente com facilidade. Este humanoide, possui 130 cm, pesa 54 Kg e pode atingir uma velocidade de 2.7 km/h. O robô ASIMO consegue caminhar e correr em superfícies irregulares, mudar de direção com suavidade, subir escadas e agarrar objetos. O ASIMO consegue compreender e responder a comandos de voz, reconhecer a cara de um grupo específico de indivíduos. Usando, as câmaras dos olhos, este humanoide consegue mapear o ambiente à sua volta e registrar objetos estacionários, assim como evitar obstáculos em movimento à medida que caminha no seu ambiente. [37]

Os laboratórios da Fujitsu, no Japão, desenvolveram a série de robôs humanoides designados por HOAP (*Humanoid for Open Architecture Platform*). O primeiro desta série, o HOAP-1, criado em 2001, pesava 6Kg e possuía 480 mm de altura e possui 20 DOFs (6 em cada perna e 4 em cada braço). Esta plataforma fornece algumas vantagens aos investigadores, nomeadamente elevada portabilidade e modularidade, um ambiente de desenvolvimento simples, interface comum, componentes e códigos de fonte reutilizáveis, fácil seleção e integração de dispositivos de *hardware* e uso fácil. Posteriormente, criaram-se mais robôs no seguimento do HOAP-1, dando origem ao HOAP-2 e HOAP-3, sendo este o modelo mais recente e mais desenvolvido. Este último modelo é ligeiramente maior (60cm) e possui câmeras CCD, microfone, *speaker*, e LEDs para mostrar expressões. Adicionaram um eixo móvel para a cabeça e para as mãos, e sensores de força e de distância que permitem ao robô agarrar objetos. Adquiriu capacidade de reconhecimento de voz e de imagens e síntese de discurso. Todas as funções do HOAP-3 podem ser controladas via *Wireless*. [38]

Robô Humanoide Darwin

O famoso dinâmico antropomórfico robô com inteligência (Darwin) é um humanoide sofisticado e consiste numa plataforma de *hardware* sofisticada capaz de executar marcha bípede e movimentos à semelhança dos humanos, tendo sido desenvolvida como uma plataforma para estudar marchas bípedes e tendo servido como plataforma de base para a primeira entrada da Virginia Tech na Robocup em 2007. Foram desenvolvidas cinco versões do Darwin, notando-se uma evolução de versão para versão. A mais recente é o Darwin-OP e representa a versão mais evoluída relativamente às restantes versões 4.0, 3.0, 2.0, 1.0 e 0.0 (a mais antiga). Esta primeira versão foi usada como um estudo de *design* para determinar a viabilidade de criar um robô humanoide. O estudo inicial de *design* na versão 0.0 do Darwin demonstrou uma ótima integração de software e controlo de motor de um robô humanoide com 21 graus de liberdade, sendo cada junta atuada por motores DC via controlo distribuído com. O Darwin 0.0 conseguia levantar-se e caminhar todavia não usava giroscópios nem sensores de força para feedback no controlo do movimento. Todavia, a competição Robocup obrigou a um maior investimento de modo a conseguir um *hardware* mais sofisticado. A segunda versão (1.0) possui marchas e *software* melhorados em relação ao Darwin 0.0. A versão Darwin 1.0 possui uma estrutura leve de alumínio com um cobrimento em plástico fabricado por prototipagem rápida. Os braços e pernas estão ligados ao corpo por três juntas esféricas (3 DOF) as quais possibilitam uma vasta gama de movimentos. [30] [39]

A versão seguinte, além de possuir melhor *software* e apresentar marchas mais eficientes que as duas primeiras versões, também possui inteligência artificial. A plataforma foi vastamente utilizada principalmente como ferramenta de pesquisa para estudar locomoção bípede e humanoide. Além disso, foi adaptada para a competição internacional Robocup, a qual tem-se tornado um foco de desenvolvimento na área da locomoção humanoide. Por outro lado, o Darwin 2.0 foi um passo evolucionário na medida em que possui cérebro, o qual é composto por um computador PC104+, 1394 câmaras, um giroscópio, e protocolo de comunicação rede *wireless* 802.11, o qual possibilitou ao robô maior autonomia e menos restrições quando este está em operação. Este *software* inteligente, usado no Darwin, identifica e localiza objetos, e

permite ao Darwin desempenhar tarefas de alto nível, como por exemplo, jogar futebol. A inteligência artificial corre no PC104+ e usa dados sensoriais como por exemplo os dados provenientes das câmaras para determinar a ação do robô, a qual é enviada para o gerador de marcha. Este recorre a feedback por parte do giroscópio de modo a criar uma marcha modificada estável para enviar aos servo-motores do robô.

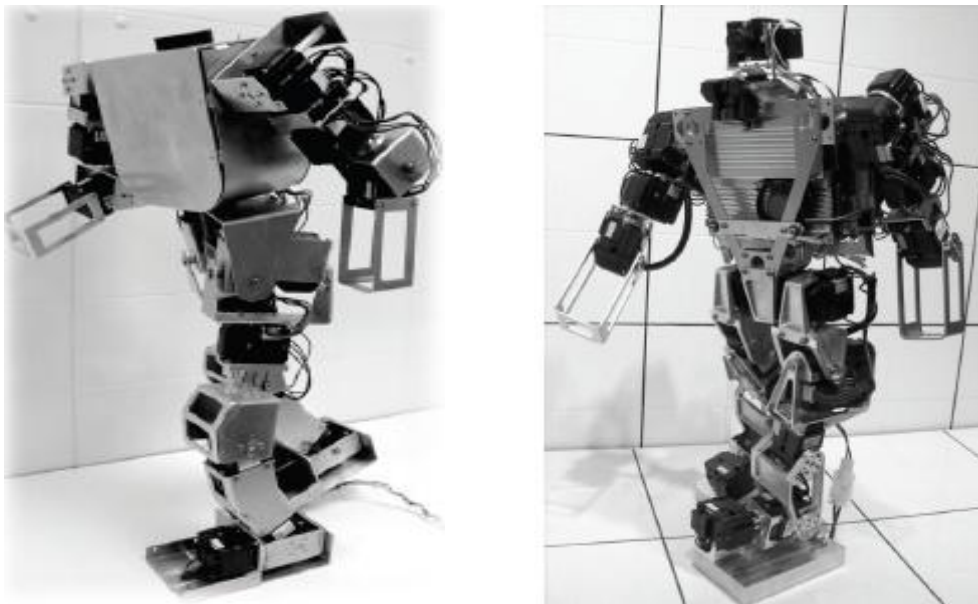


Figura 2. 5 - Primeiras versões dos robôs Darwin. Darwin 1.0 à esquerda e Darwin 2.0 à direita. [39]

Os motores dos robôs são controlados por uma rede série RS-485 e são construídos por potenciômetros, os quais retornam os ângulos das juntas. Enquanto o robô se movimenta, um giroscópio com informação sobre a aceleração e orientação comunica através de uma porta série RS-232 de modo a que o robô consiga modificar a sua marcha e equilíbrio. [39]

Posteriormente fabricaram-se as versões 3.0 e 4.0. A primeira consistiu numa melhoria principalmente a nível de potência e força. A versão 4.0 apresentou algumas mudanças radicais relativamente aos seus antecessores, sendo que a principal medida de inovação consistiu em substituir a plataforma PC104+ por uma Gumstix, possibilitando a redução do tamanho e peso do robô.

A evolução desta série de robôs Darwin, iniciada em 2004, não se ficou por aqui e lançou mais um modelo – o Darwin Open Platform. O desenvolvimento deste modelo foi suportado pela National Science Foundation e levado a cabo pela Virginia

Tech's Robotics & Mechanisms Laboratory (RoMeLa) em colaboração com a Universidade de Purdue e a Universidade de Pennsylvania, sendo o mentor do projeto o Dr. Dennis Hong da equipa de Romela. O Darwin-OP é um pequeno robô humanoide com um poder computacional avançado e sensores sofisticados. Possui 20 graus de liberdade e o seu centro de massa localiza-se na pélvis. Esta localização é ótima para um *balancing* adequado e distribuição adequada de inércia durante a marcha, especialmente nas extremidades. Este último modelo é uma plataforma verdadeiramente maneável onde os utilizadores são encorajados a modificar quer o *software* quer o *hardware*, e consegue correr vários sistemas operativos, desde o Windows ao Linux. A nível de programação aceita várias linguagens inclusive C++, Python, LabVIEW e MATLAB. Tem uma altura de aproximadamente 455 mm e um peso de 2.8 kg. [40]

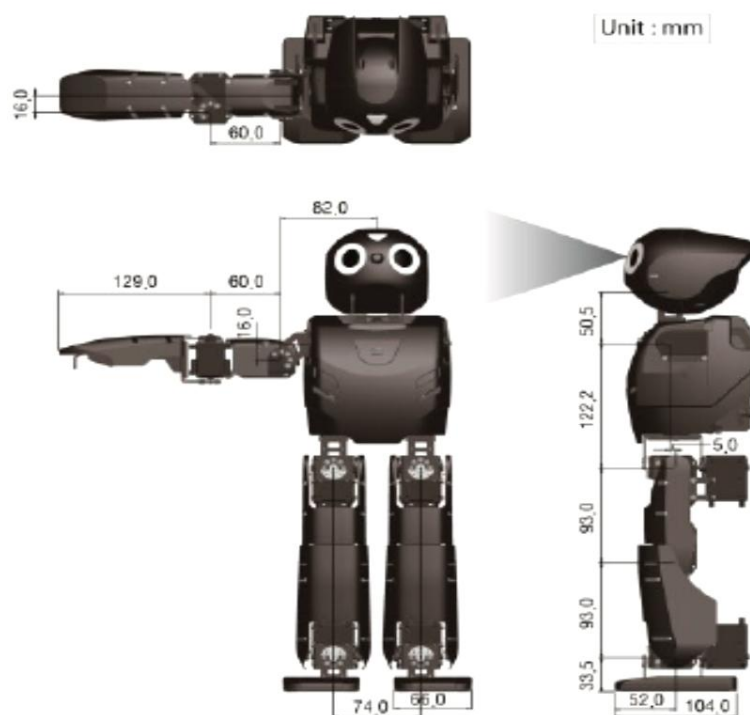


Figura 2. 6 - Diferentes perspetivas do Darwin-OP com as respetivas dimensões. [40]

3. Funções de Custo

Nos algoritmos evolucionários, uma população de indivíduos é usada para encontrar uma solução para um dado problema. Cada indivíduo tem um número de genes que representam possíveis valores para as variáveis do problema. Em cada passo da evolução, a *fitness* é determinada para cada indivíduo, isto é, os valores atuais dos seus genes são usados para resolver o problema e a solução é avaliada por uma função de *fitness* (função objetivo). Posteriormente, baseado na população atual e a *fitness* dos seus indivíduos, é criada uma nova geração que substitui a atual. [41]

Neste contexto, a solução do sistema indeterminado é obtida formulando funções de custo e utilizando uma técnica de otimização. A função custo fornece a base para comparação das diversas soluções, e a melhor solução é obtida pelo algoritmo de otimização.

As principais funções de custo para otimização da marcha são:

- Velocidade máxima: a marcha deve ajudar o robô a atingir velocidade máxima, logo esta corresponde a um índice básico de performance; [1]

- Estabilidade: um robô necessita de ser estável de modo a conseguir movimentar-se num determinado ambiente e não cair. A estabilidade pode ser dinâmica ou estática. A primeira pode ser verificada pelo centro de gravidade (COG), enquanto a dinâmica é muitas vezes verificado pelo ponto de momento zero (ZMP), o qual possui um papel importante na otimização da marcha (principalmente na bípede). [42] [43]

- Consumo mínimo de energia (MCE): este é um dos critérios mais tidos em conta uma vez que um padrão de locomoção eficiente a nível da energia consumida resulta num movimento de marcha mais natural. Reduzindo a energia consumida também permite maximizar o tempo de operação da bateria. [44] [45] [46] [47]

- Variação mínima de torque: este princípio baseia-se na suavidade ao nível do torque, resultando num movimento mais estável devido a uma mudança mais suave na aceleração. [44] [45] [48] [49]

Geralmente, o problema da síntese de marcha, quer seja a caminhar ou a subir ou descer escadas, consiste em determinar as trajetórias angulares das juntas, para conectar a primeira e última postura do robô bípede para a qual a energia consumida e

a variação de torque são mínimas. Capi et al. (2002) demonstrou que a função de custo relativa ao consumo de energia (MCE), pode-se assumir que a energia que controla a posição do robô é proporcional à integração do torque ao quadrado em relação com o tempo, pois o torque da junta é proporcional à corrente. Portanto, minimizando o torque da junta pode resolver o problema do consumo mínimo de energia. [50]

Existem duas formas de desenvolver a marcha de um robô, nomeadamente a evolução *online* e a evolução *offline*. A primeira desenvolve a marcha diretamente no robô, enquanto o método *offline* envolve um simulador e as soluções são avaliadas usando as funções objetivo mencionadas em cima. No caso de evolução *online*, a *fitness* pode não ser diretamente calculada. Em vez disso é determinada através de medições, isto é, as soluções têm de ser testadas deixando o robô realmente caminhar com os parâmetros codificados pelo cromossoma, e a *fitness* para cada indivíduo é avaliada usando os sensores do robô (câmara digital, sensor infravermelho e giro-sensor) ou diretamente pelo utilizador. [1]

3.1. Controlo da Estabilidade

A estabilidade é um fator essencial a ter em conta na otimização da locomoção de um bípede. A maioria dos autores utilizam funções de custo que visam controlar medidas de estabilidade tais como o ZMP, COG ou o COM (centro de massa).

O conceito do ponto de momento zero (ZMP), proposto por Vukobratovic et al., teve origem há mais de três décadas mas não caiu em desuso uma vez que as forças da gravidade continuam a governar a estabilidade da locomoção robótica. [42]

Quando o bípede executa o ciclo de marcha está sujeito às seguintes forças: força da gravidade, forças de reação e forças de inércia. De acordo com o princípio de D'Alembert, se todas as forças forem equilibradas, o movimento do bípede é exequível. Deste modo, segundo este princípio, a força resultante e os momentos que atuam no bípede devem ser nulos. [51]

Neste contexto, o conceito do ZMP tem sido sujeito a diferentes interpretações por parte dos investigadores, as quais são corretas embora possam ser diferentes no grau de detalhe. Por exemplo, Nakamura et al. define o ZMP como o ponto no chão cujo momento devido às forças de gravidade e de inércia não possui componente ao longo do eixo horizontal. Por outro lado, Arakawa et al. define o ZMP como o ponto p onde $T_x = 0$ e $T_y = 0$, sendo T_x e T_y os momentos no eixo x e no eixo y gerados pela força de reação F_x e torque de reação T_x respetivamente. [52] [53]

Por outro lado, é importante estabelecer uma diferença entre o centro de pressão (COP) e o ZMP uma vez que estes dois conceitos são por vezes confundidos. A pressão entre o pé e o chão pode ser substituída por uma força que atua no centro de pressão – COP. Dito de outro modo, o centro de pressão é a localização do ponto correspondente ao vetor vertical da força reação do chão. Representa a média ponderada de todas as pressões sobre a superfície da área em contato com o chão. Se esta força equilibra todas as forças ativas no mecanismo durante o movimento (inércia, gravidade, forças de Coriolis e centrífugas e momentos), o seu ponto de atuação é o ZMP. Assim, no caso de uma locomoção dinâmica estável, o COP e o ZMP coincidem. No caso de uma perturbação que desloca o ponto de atuação da força de reação do chão para a borda do pé, o momento de perturbação provoca rotação do sistema bípede à volta da borda do pé. Neste caso apenas se pode falar do ZMP

fictício, cuja distância até à borda do pé representa o momento de perturbação. Em suma, o ZMP coincide sempre com o COP (locomoção dinâmica estável), mas o COP nem sempre é o ZMP (locomoção dinâmica não estável). Todavia, o ZMP fictício nunca coincide com o COP uma vez que este não existe fora do polígono de suporte. [43] [54]

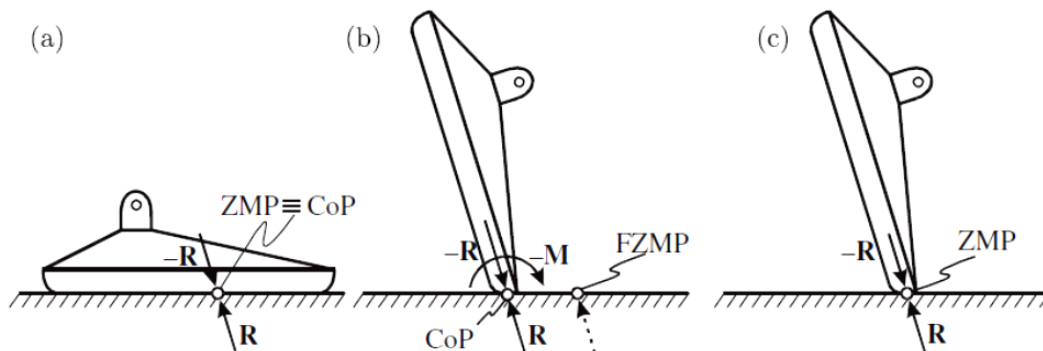


Figura 3. 1 - Possíveis relações entre ZMP e COP para um pé não rígido. a) Locomoção dinâmica estável, b) Locomoção não estável onde o ZMP não existe e o ponto de atuação da força de reação do chão é o COP, enquanto o ponto onde $M_x = 0$ e $M_y = 0$ está fora do polígono de suporte (FZMP) – o sistema roda à volta da borda do pé, e c) equilíbrio dinâmico com a ponta do pé. [43]

Outros fatores de estabilidade que são usados para gerar padrões de locomoção estável são o centro de massa (COM) e o centro de gravidade (COG). O primeiro é um ponto equivalente à massa total do corpo no sistema global de referência (GRS) e é a média ponderada do COM de cada segmento do corpo. A projeção vertical do COM no chão é muitas vezes denominada por centro de gravidade (COG).

Por vezes é necessário controlar o ZMP (*Zero Moment Point*) ou o COG (centro de gravidade) no sentido do robô bípede assumir uma postura estável. A metodologia baseada no ZMP está bem definida e garante a estabilidade do robô, é fácil de implementar e pode ser aplicada a uma marcha dinâmica. Todavia, este método requer modelos precisos do robô e do ambiente, assim como um controlo externo *online* para lidar com perturbações externas. Geralmente, na locomoção bípede, é usada uma restrição que implica que a projeção do COG no chão esteja localizada dentro do plano de suporte de modo a não tropeçar. [55] [56] [57]

Uma vez que o bípede dá passos alternadamente, o COG deve ser deslocado sobre a perna de suporte durante o ciclo da marcha, permitindo à outra perna executar a fase *swing* do passo. Este momento é crucial pois uma eventual falha ao determinar o deslocamento do COG sobre a perna de suporte no *timing* correto pode levar a uma queda.

Uma locomoção estática estável ocorre durante movimento de baixa velocidade e é caracterizada pelo centro de gravidade (COG) do robô e a restrição que implica que a projeção deste deve manter-se dentro da região estável, a qual corresponde ao polígono correspondente ao plano de suporte que diz respeito à zona do chão onde o pé pousa. Deste modo, o movimento do COG no plano frontal é aumentado consideravelmente comparado com o movimento de marcha normal, e torques estáticos (devido à gravidade) possuem um efeito mais forte durante a marcha do que torques dinâmicos (devido à inércia).

Por outro lado, uma locomoção dinâmica estável ocorre durante movimentos de maior velocidade e, ao contrário da locomoção estática, o COG pode estar fora da região estável. Enquanto o robô dá um passo, o centro de massa deve estar fora do pé de suporte, embora o ZMP tem de estar dentro. Quando o movimento de marcha é rápido, o COG pode estar perto ou fora do limite do polígono de suporte, conseguindo-se uma locomoção semelhante à marcha humana. Uma locomoção dinâmica estável é muito importante para robôs bípedes uma vez que, na maior parte do tempo, apenas possuem uma perna de suporte. [43] [51]

A dinâmica de sistemas bípedes é não-linear e difícil de analisar. Em muitos estudos são utilizados modelos bípedes simplificados. O modelo mais popular e amplamente usado é o modelo do pêndulo invertido. Neste modelo, tal como foi descrito anteriormente, o corpo todo é substituído por uma massa concentrada no centro de massa. Este modelo é útil para a análise de estabilidade de bípedes através da computação do ZMP. A abordagem mais comum para a geração de marcha é calcular trajetórias mantendo a estabilidade postural através da dinâmica de sistemas. Desacoplando os subsistemas reduz a complexidade na geração da marcha bípede, uma vez que equações dinâmicas desacopladas e linearizadas simplificam a computação do ZMP. A injeção de torque no tornozelo fornece compensação do ZMP mantendo a estabilidade postural durante várias atividades bípedes. Mantendo o

centro de massa a uma altura específica, o modelo linear de pêndulo invertido serve para criar maior estabilidade na marcha. Este tipo de geração de marcha baseado no ZMP é utilizado pelo bem conhecido humanoide ASIMO. [14]

Muitos dos algoritmos de marcha de sucesso baseiam-se no uso do ZMP como critério para o planeamento *offline* do movimento e para a compensação online do *balancing*. Estes métodos baseados no ZMP têm sido muito eficazes sobretudo em robôs com pés planos. Contudo, requerem que se modele de forma precisa a dinâmica do robô e uma monitorização de controlo de trajetória de elevado ganho, e os padrões gerados resultam numa “postura de joelho dobrado” de modo a evitar singularidades de cinemática inversa. No que diz respeito a eficiência a nível de energia, tais padrões não são desejáveis uma vez que deve ser continuamente aplicado um torque á junta do joelho de modo a manter a postura de joelho dobrado. [58]

4. Métodos de Otimização

A otimização da marcha é muito importante para robôs com pernas pois consegue determinar a posição ótima, a velocidade e aceleração para cada um dos graus de liberdade em qualquer momento, e o padrão de marcha pode afetar diretamente a estabilização dinâmica do robô.

Existem vários tipos de estratégia de sequenciação e coordenação dos movimentos das pernas para conseguir obter uma marcha natural e eficiente.

Uma das estratégias baseia-se no facto da marcha humana ou mesmo a dos animais serem ideais/ótimas uma vez que estes se movimentam de uma forma que minimiza os custos metabólicos de locomoção. Contudo algumas pesquisas revelam algumas inconsistências cinemáticas entre humanos/animais e robôs logo a locomoção biológica não pode ser diretamente utilizada por estes. Isto implica correções cinemáticas quando se calcula ângulos e trajetórias das juntas.

Outras estratégias encaram a geração de uma marcha como um problema de restrições ou condições. Assim, os padrões de marcha são gerados através da otimização de índices de performance tais como a velocidade de movimento, critérios de estabilidade, forças atuadoras e consumo de energia.

As equações dinâmicas relativas à locomoção de um robô com pernas são de elevada ordem, fortemente acopladas e não-lineares, e a otimização da marcha requer a pesquisa de um conjunto de parâmetros num espaço altamente multidimensional e irregular. [1]

Nas primeiras investigações, a marcha aplicada nos robôs humanóides era gerada com base na informação tirada do movimento humano (Vukobratovic et al.). A maioria dos trabalhos posteriores (Roussel 1998, Silva e Machado 1998, Channon 1996) consideram o consumo mínimo de energia como critério fundamental para a geração de marcha por parte de humanóides. Channon et al. fez uma análise ao consumo de energia, a qual está fundamentalmente relacionada com a velocidade da marcha e o comprimento da passada. As funções de distribuição de *inputs* de torque são obtidas minimizando os torques das juntas. Os trabalhos levados a cabo por Capi et al. (2003) e Nakano et al. (1999) obtiveram resultados que demonstraram que

marchas geradas consoante os critérios do consumo mínimo de energia e consumo mínimo de torque possuem várias vantagens. A marcha humana gerada com base em consumo mínimo de energia é bastante similar com a dos humanos. Além disso, o tempo de operação é maior quando o robô atua por meio de uma bateria. Por outro lado, a marcha baseada na variação mínima de torque foi mais estável devido à mudança mais suave do torque e dos elos de aceleração. Estas observações mostram que seria vantajoso gerar uma marcha de modo a satisfazer uma série de critérios. Isto traz inerente um problema de otimização multi-objetivo, o qual pode não ter uma solução que seja a melhor no que diz respeito a todos os critérios/objetivos. Geralmente, o objetivo é determinar uma superfície de *trade-off*, a qual é um conjunto de pontos não dominados que correspondem a soluções, conhecidas como frente-Pareto. [44] [59] [60] [61]

4.1. Otimização Multi-objetivo

O problema multi-objetivo é quase sempre resolvido combinando os múltiplos objetivos num objetivo escalar usando coeficientes que dizem respeito ao peso de cada um. Portanto, para combinar diferentes objetivos numa única função de custo, é necessária uma decisão *à priori* sobre a importância relativa dos diversos objetivos, dando ênfase a um tipo particular de solução. Estas técnicas muitas vezes requerem informação específica relativa ao problema em questão, tal como a gama total que cada objetivo engloba. Em problemas que dizem respeito à geração de marcha humanoide, esta informação é raramente conhecida em avanço, tornando a seleção dos pesos singulares de cada objetivo difícil. Além disso, não existe uma base racional para determinar pesos adequados e a função objetivo assim formada pode perder alguma significância devido ao facto de se combinar objetivos não-comensuráveis. De modo a evitar esta dificuldade, Becerra et al. apresentaram um método que é baseado na otimização do objetivo mais preferido e considerando os outros objetivos como condições limitadas por alguns níveis admissíveis. Estes níveis são depois alterados de modo a gerar o conjunto inteiro da frente-Pareto. A maior desvantagem desta abordagem, para além de consumir bastante tempo, é o facto de tender para encontrar soluções não-dominadas fracas. Posto isto, Coello et al. apresentaram um algoritmo evolucionário multi-objetivo (MOEA) cuja vantagem reside no facto de quando corre o algoritmo uma única vez, são imediatamente geradas tipos de marchas com características completamente diferentes e, portanto, o robô pode trocar entre as marchas de modo a se adequar melhor às condições do ambiente em que se encontra. [50] [62] [63]

Nos últimos 15 anos, a otimização evolucionária multi-objetivo tem-se apresentado como um campo de pesquisa e aplicação cada vez mais popular e útil. Os algoritmos de otimização evolucionária usam uma abordagem, baseada na população, e na qual participa mais que uma solução numa iteração e desenvolve uma nova população de soluções em cada iteração. Esta metodologia apresenta vantagens tais como o facto de não necessitar de informação derivativa, de serem relativamente simples de implementar, flexíveis e ter uma aplicabilidade vasta. Para resolver problemas de otimização com apenas um objetivo, particularmente na procura de uma

única solução ótima, o uso de uma população de soluções pode parecer redundante, todavia para problemas de otimização multi-objetivo, os procedimentos da otimização evolucionária são a escolha ideal. Os problemas de otimização multi-objetivo, por natureza, dão origem a um conjunto de soluções Pareto, o qual posteriormente necessita de um processamento de modo a alcançar-se uma única solução preferencial. De modo a conseguir-se a primeira parte, torna-se uma premissa bastante normal utilizar otimização evolucionária, uma vez que o uso de população numa iteração ajuda a encontrar simultaneamente múltiplas soluções não-dominadas, o que retrata um trade-off entre objetivos no correr de uma única simulação. Uma solução que seja extrema dentro da frente-Pareto relativamente a um objetivo requer um compromisso com outros objetivos. Isto proíbe a escolha de uma solução que seja ótima em relação a unicamente um objetivo, sugerindo dois objetivos principais para a otimização multi-objetivo: obter um conjunto de soluções que assente na frente-Pareto e que sejam suficientemente divergentes para representar a gama inteira da frente-Pareto. [64] [65] [66]

4.1.1 Frente Pareto

Estes algoritmos têm vindo a tornar-se o método de eleição para problemas de otimização que sejam muito complexos para serem resolvidos com métodos determinísticos, tal como programação linear ou métodos de gradiente, e são capazes de encontrar *trade-offs* ideais no sentido de conseguir um conjunto de soluções que sejam ótimas num contexto global. Em contraste com espaços de pesquisa escalares totalmente ordenados, os espaços de pesquisa multidimensionais são apenas parcialmente ordenados, ou seja, duas soluções diferentes têm duas possíveis formas de se relacionar: ou uma domina a outra, ou então nenhuma é dominada.

Considerando em sentido geral, o seguinte o problema de otimização multi-objetivo com m variáveis x de decisão (parâmetros) e n objetivos i :

Maximizar

$$y = f(x) = (f_1(x_1, \dots, x_m), \dots, f_n(x_1, \dots, x_m))$$

Onde

$$x = (x_1, x_2, \dots, x_m) \in X$$

$$y = (y_1, y_2, \dots, y_m) \in Y$$

e onde x é chamado vetor de decisão, X é o espaço de parâmetros, y o vetor objetivo e Y o espaço objetivo.

Um vetor de decisão $a \in X$ é suposto dominar um vetor de decisão $b \in X$ ($a > b$) se e apenas se:

$$\forall i \in \{1, \dots, n\}: f_i(a) \geq f_i(b) \quad \wedge \quad \exists j \in \{1, \dots, n\}: f_j(a) > f_j(b)$$

O vetor de decisão a é chamado Pareto ótimo se e apenas a é não dominado em relação a todo o espaço de parâmetros X . Se o conjunto X_0 não é especificado explicitamente, todo o espaço de parâmetros está implícito. Vetores de parâmetros Pareto ótimos não podem ser melhorados em nenhum objetivo sem prejudicar pelo menos um dos outros objetivos, e neste sentido, representam soluções ótimas a nível

global. É de referir que um conjunto Pareto ótimo não contém necessariamente todas as soluções ótimas Pareto em X .

Quando se pretende estender algoritmos evolucionários a casos multi-objetivo, surgem dois problemas principais que devem ser discutidos, nomeadamente como realizar o exercício de *fitness* e seleção no sentido de chegar ao conjunto Pareto ótimo, e como manter uma população diversificada no sentido de prevenir uma convergência prematura e alcançar uma frente de *trade-off* bem distribuída e ampla. [65] [67]

4.1.2 Hipervolume

Um dos critérios de performance mais usados em algoritmos de otimização é a medida de hipervolume. Este indicador permite orientar a pesquisa em direção a regiões do espaço objetivo definidas pelo utilizador e, simultaneamente possui a propriedade de refinar a relação de dominância Pareto.

A medida de hipervolume é frequentemente usada para comparar resultados de algoritmos de otimização multi-objectiva evolucionária (EMOA), sendo vista como uma medida para a qualidade de um conjunto de pontos não dominados. Os investigadores têm empregado o hipervolume em dois tipos de abordagens diferentes: Fleischer et al. sugeriu reformular o problema de otimização multi-objetivo para um problema de um único objetivo maximizando o hipervolume de um conjunto finito de pontos não dominados. Knowles et al. utilizou o hipervolume dentro de uma estratégia de arquivamento para EMOA.

A medida de hipervolume foi originalmente proposta por Zitzler et al., que lhe chamaram tamanho de espaço dominado. Estes investigadores constataram que esta medida prefere regiões convexas a regiões não convexas.

Zitzler et al. aferiram que no caso bidimensional, cada solução ótima Pareto x cobre uma área, um retângulo, definido pelos pontos $(0,0)$ e $(f_1(x), f_2(x))$. A junção de todos os retângulos cobertos pelas soluções ótimas Pareto constitui o espaço abrangente, sendo o tamanho deste usado como medida. Este conceito pode ser canonicamente estendido a múltiplas dimensões. Basicamente, o hipervolume é usado como um critério de seleção de modo a descartar o indivíduo que menos hipervolume contribui para a pior frente Pareto. Emmerich et al. aplicou um algoritmo de otimização evolucionário multi-objetivo em que cada novo indivíduo gerado apenas entra na população se, ao substituir um membro dela, vai aumentar o hipervolume coberto pela população. Deste modo, novos indivíduos juntam-se à população se substituírem indivíduos dominados e assim contribuir para uma maior qualidade da população. Neste contexto, este critério impede que um indivíduo não-dominado seja substituído por um indivíduo dominado.

Este indicador é diferente do critério usado geralmente no NSGA denominado por *crowding distance*. Este é definido como a soma dos comprimentos dos lados de

um cuboide formado por soluções vizinhas em caso de soluções não extremas na frente Pareto. Ao contrário da medida do hipervolume, é suposto distribuir as soluções uniformemente na frente Pareto enquanto que o uso do hipervolume, prende-se com o fato de se pretender distribuir as soluções de tal modo que consiga maximizar o hipervolume coberto. [68] [69] [70]

4.2 Computação Evolucionária

Na década de 50 e 60, vários cientistas na área dos computadores estudaram sistemas evolucionários com a ideia que a evolução pode ser usada como uma ferramenta de otimização para problemas de engenharia. A ideia nesses sistemas era desenvolver uma população de soluções candidatas para um dado problema usando operadores inspirados pela variação natural genética e pela seleção natural. [71]

A Computação Evolucionária (EC), incluindo o Algoritmo Genético (GA), a Programação Genética (GP), a Programação Evolucionária (EP), e a Estratégia Evolucionária (ES), é uma escolha natural para a otimização da marcha. A EC baseia-se na teoria da evolução natural de Darwin, ou seja, os mecanismos de locomoção resultaram de uma seleção natural e da interação entre os indivíduos e do ambiente natural. Esta estratégia tem apresentado resultados eficazes na otimização de problemas de larga dimensão, multi-objetivos e multi-condicionados.

Por outro lado, este método tem uma capacidade de pesquisa global muito forte sendo independente da população inicial, diminuindo o risco de se equivocar num mínimo local na procura da solução ótima. É também resistente ao ruído na função de avaliação e oferece um modelo que apenas requer feedback do ambiente à sua volta de modo a melhorar a sua performance quando é implementada uma evolução *online* num robô real.

Uma das vantagens da EC consiste no facto de poder recorrer a paralelismos. Dado que a otimização da marcha é muitas vezes um problema de grande escala e a função objetivo e as suas variáveis são muitas vezes complexas, o processo de otimização evolucionária pode consumir demasiado tempo devido ao elevado custo computacional da EC o qual, por sua vez, se deve às avaliações iterativas das soluções candidatas. Neste contexto, é vantajoso usar implementações paralelas da EC de modo a ganhar eficiência e melhorar a qualidade da solução obtida para a otimização da marcha baseada em EC. [1]

4.2.1 Algoritmos Genéticos

O Algoritmo Genético (GA) é a ferramenta mais utilizada para a otimização da marcha. Este método foi inventado por John Holland nos anos 60 e foram desenvolvidos por ele e pelos seus colegas na Universidade do Michigan. Em contraste com estratégias de evolução e com a programação evolucionária, o objetivo principal de Holland não foi criar algoritmos para resolver problemas específicos, mas sim estudar formalmente o fenómeno de adaptação tal como ele ocorre na natureza e desenvolver formas que consigam importar estes mecanismos de adaptação natural para sistemas de computação.

Nos últimos anos tem havido uma interação muito difundida entre investigadores estudando vários métodos de computação evolucionária, e os limites entre os GAs, estratégias de evolução, programação evolucionária, e outras abordagens evolucionárias foram quebrados até um certo ponto. Atualmente, os investigadores utilizam muitas vezes os termos “algoritmo genético” para descrever algo bem distante do conceito original de Holland. [71]

4.2.2.1 Terminações Biológicas

No contexto dos algoritmos genéticos, são usados termos biológicos em analogia com a biologia real, embora as entidades referidas são bem mais simples que as reais.

Todos os organismos vivos consistem em células, e cada célula contem um ou mais cromossomas – cadeias de ADN – que servem como uma “impressão digital” para o organismo. Um cromossoma divide-se em genes que correspondem a blocos funcionais de ADN, sendo que cada um codifica uma determinada proteína. Todos os cromossomas juntos constituem o genoma do organismo. Em algoritmos genéticos, o termo cromossoma refere-se tipicamente a uma solução candidata a um problema, muitas vezes codificado como uma cadeia de bits. Os “genes” são bits individuais ou curtos blocos de bits adjacentes que codificam um elemento em particular da solução candidata. Por exemplo, no contexto de uma função de otimização com vários

parâmetros, os bits que codificam um parâmetro em particular podem ser considerados um gene. [71]

Em suma, algoritmo genético é um algoritmo de pesquisa baseado na mecânica da genética e da seleção natural. Combina a sobrevivência do mais forte com a troca aleatória, embora estruturada, de informação, de modo a formar um algoritmo de pesquisa com uma capacidade inovadora de evolução natural.

Esta estratégia apresenta algumas vantagens: pesquisa baseada numa população onde é possível obter uma solução global ótima; não necessita de nenhuma informação auxiliar tais como gradientes, derivadas, etc.; problemas complexos e multimodais podem ser resolvidos por otimização global; independente do tipo de problema, isto é, adequado para todo o tipo de problemas.

Um GA inicia com a criação aleatória de uma população de *strings* e a partir daí gera sucessivas populações de *strings* que aperfeiçoam ao longo do tempo. Algumas das operações que geram as novas populações são ilustradas na figura 4.1. [72]

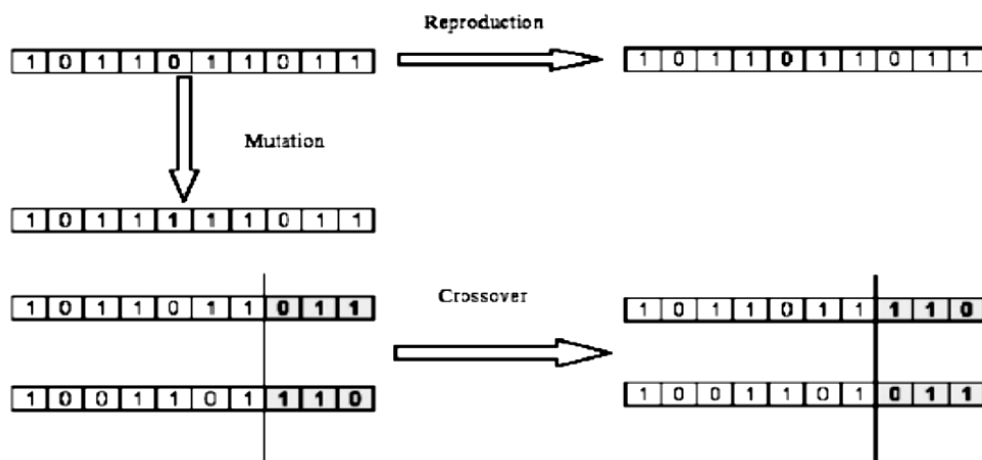


Figura 4. 1 - Representação esquemática de operações básicas dos GA's. [72]

Reprodução é um processo no qual *strings* individuais são copiadas de acordo com os seus valores da função custo. As que possuem melhores valores representativos de aptidão têm a probabilidade de contribuir com uma ou mais *offstrings* para a próxima geração. Uma vez que a *string* tenha sido selecionada para a reprodução, uma réplica exata desta é feita. Esta réplica é depois introduzida numa “piscina genética”, correspondendo a uma nova população experimental para posterior ação destes operadores genéticos.

O operador de *Crossover* consiste tipicamente em trocar material genético entre dois cromossomas pais haploides. Pode proceder-se em dois passos: primeiro, membros de *strings* reproduzidos recentemente na piscina genética são acoplados aleatoriamente. Segundo, cada par de *strings* sujeita-se a um cruzamento da seguinte forma: é seleccionada uma posição "*K*" aleatória entre 1 e o tamanho *l* da *string* menos 1. Duas novas *strings* são criadas pela permuta de todos os caracteres entre as posições (*K+1*) e *l* inclusive. Isto retrata a forma mais simples de *crossover* sendo usado apenas um ponto para delinear a permuta. Todavia, existe *crossover* com mais de um ponto. De fato, o *crossover* em dois pontos reduz o "*bias* posicional" (esquemas que podem ser criados ou destruídos por *crossover* dependem fortemente da localização dos bits no cromossoma) e reduz o "efeito das pontas" que acontece no *crossover* de um só ponto onde os segmentos trocados entre os dois pais contem sempre as pontas das *strings*.

O *crossover* é o maior instrumento de inovação e variação nos GAs. Por outro lado, a mutação assegura a população contra a fixação permanente em qualquer *locus* em particular. O operador de mutação consiste na alteração aleatória do valor de uma posição da *string*. Em GA, a probabilidade de isto acontecer é pequena, uma vez que uma taxa maior poderia levar à perda de informação importante. [71] [72]

Em suma, embora não haja uma definição rigorosa de algoritmo genético que os diferencie dos outros algoritmos evolucionários, pode-se dizer que o GA é composto pelos seguintes elementos: população de cromossomas, seleção de acordo com a *fitness*, *crossover* e mutação aleatória para a criação de uma nova descendência. Holland usava um quarto elemento, designado por inversão, o qual era um operador de reordenamento que consistia em escolher dois pontos numa *string* e reverter a ordem dos bits entre eles. Todavia é raramente utilizado nas implementações atuais pois as suas vantagens, se existirem, não estão bem definidas. O cromossoma corresponde a uma string de bits, e cada locus no cromossoma possui dois alelos possíveis: 1 ou 0. Cada cromossoma pode ser visto como um ponto no espaço de pesquisa de soluções candidatas. Os GAs requerem quase sempre uma função de *fitness* que atribui uma classificação a cada cromossoma na população corrente que depende de quão bem os cromossomas resolvem o problema em questão.

Os algoritmos genéticos correspondem à abordagem mais utilizada, todavia algumas modificações podem ser levadas a cabo de modo a melhorá-lo e a adequar-se a problemas específicos da otimização da marcha. Existe uma adaptabilidade inerente no GA, que pode mudar as probabilidades de cruzamento e mutação, de modo a equilibrar a utilização e exploração global e local para o progresso da otimização evolucionária. Por exemplo, GA adaptativo é usado para otimizar a marcha do robô humanóide enquanto sobe e desce umas escadas, pesquisando parâmetros ótimos de trajetória em polinômios combinados. Mecanismos adaptativos podem também ser aplicados no controlo da taxa de mutação. Este método coloca radiação (esta diminui ao longo do tempo) no meio de uma região onde um grande grupo de indivíduos é agrupado dentro do mesmo local para aumentar drasticamente a taxa de mutação nesta área, levando todos os indivíduos na próxima geração a sofrerem mutações e a dispersarem para outras áreas. Este mecanismo pode ser útil no controlo do comportamento de aprendizagem do GA, assim como torna este mais robusto no que diz respeito a ruído na avaliação de parâmetros na prevenção de convergência prematura para extremos sub-ótimos. [1] [71]

4.2.1.2 Métodos de Seleção

Num GA é necessário definir a forma como se vai realizar a seleção, isto é, como escolher os indivíduos na população que vão criar a descendência para a próxima geração, e quantos descendentes é que vão criar cada um. O propósito da seleção é, logicamente, dar ênfase aos indivíduos mais aptos na população na esperança que a sua descendência, por sua vez, possua uma *fitness* ainda melhor. A seleção tem de ser balanceada com a variação inerente ao *crossover* e à mutação: uma seleção muito forte implica que indivíduos sub-ótimos bastante aptos tomem controlo da população, reduzindo a diversidade que é necessária para haver variação e progresso; uma seleção muito fraca resulta numa evolução muito lenta.

Neste contexto, existem vários métodos de seleção propostos na literatura dos GAs, entre os quais se pode destacar o *sigma scaling*, a seleção de Boltzmann, a seleção por *ranking*, a seleção por torneio, a seleção *steady-state*, entre outros. O método conhecido por *sigma scaling*, mantém a pressão de seleção (o grau para o qual os indivíduos mais aptos estão permitidos mais prole) relativamente constante à medida que o algoritmo corre em vez de depender das variâncias da *fitness* na população. Este método mantém a pressão de seleção todavia muitas vezes é necessário diferentes pressões em tempos diferentes. O método de Boltzmann permite essa diferença de pressão – por exemplo, num estágio inicial será adequado permitir os indivíduos menos aptos a reproduzir a uma taxa semelhante à dos indivíduos mais aptos, mantendo uma seleção lenta enquanto garante diversidade na população. Posteriormente, pode ser mais adequado aumentar a seleção no sentido de dar em ênfase aos indivíduos mais aptos, assumindo que a diversidade inicial derivada da menor seleção tenha permitido à população encontrar a parte certa do espaço de pesquisa.

Outro método conhecido é o método de seleção por *ranking* o qual previne uma convergência demasiado rápida. Este método evita a possibilidade de dar a maior percentagem de descendência a um pequeno grupo de indivíduos com elevada aptidão, e deste modo reduz a pressão de seleção quando a variância de *fitness* é elevada. Por outro lado, também eleva a pressão de *fitness* quando a variância de *fitness* é reduzida. Este método apresenta uma possível desvantagem: diminuir a

pressão de seleção significa que o GA, em alguns casos, vai ser mais lento a encontrar os indivíduos mais aptos.

Os métodos proporcionais à *fitness* descritos anteriormente requerem dois passos em cada geração: um passa por calcular a *fitness* média (e para o sigma scaling, o desvio padrão) e outro passa por calcular o valor esperado para cada indivíduo. O *ranking* requer a classificação da população inteira, tornando este procedimento muito demorado. Neste contexto, um outro método – seleção por torneio – é similar à seleção por *ranking* em termos de pressão de seleção, mas é mais eficiente a nível computacional e mais recetivo a implementações paralelas. O método de seleção por torneio garante pressão de seleção mantendo um torneio entre alguns indivíduos. O vencedor desse torneio é o indivíduo com melhor *fitness* e vai ser inserido na piscina de reprodução. Esta vai ser apenas composta por indivíduos vencedores e, conseqüentemente, vai possuir uma *fitness* média superior à *fitness* média da população. Esta diferença de *fitness* fornece a pressão de seleção, a qual leva o GA a melhorar a *fitness* de geração em geração. Pode garantir-se um aumento da pressão de seleção, aumentando o tamanho do torneio (maior número de indivíduos concorrentes), uma vez que o vencedor de um torneio maior tem, em geral, uma *fitness* superior que um vencedor de um torneio inferior.

Na maioria dos GAs descritos na literatura, em cada geração, a nova população consiste inteiramente em descendência gerada por pais da geração anterior (embora alguma dessa descendência possa ser idêntica aos seus pais). Em alguns esquemas, tais como os esquemas elitistas descritos anteriormente, sucessivas gerações sobrepõem-se até um certo ponto – uma porção da geração anterior é retida para a nova população. Um método de seleção conhecido por seleção *steady-state* possui a característica de apenas alguns indivíduos serem substituídos em cada geração: usualmente um pequeno número de indivíduos menos aptos são substituídos por descendência resultante de *crossover* e mutação dos melhores indivíduos. Este método é mais recorrente em GAs onde é usada uma aprendizagem incremental e onde os membros da população resolvem o problema em questão coletivamente e não individualmente como acontece normalmente. [71] [73] [74]

4.2.2 Outras Ferramentas de EC

Outras ferramentas usadas com regularidade para desenvolver a marcha dos robôs correspondem a Programação Genética (GP) e Evolução Gramática (GE). Na abordagem da GP, a marcha é definida usando as trajetórias dos ângulos das juntas em vez do *locus* dos membros de modo a reduzir o espaço de pesquisa para otimização. É utilizado um mecanismo de arquivamento de elite (EAM) de modo a prevenir a convergência prematura e melhorar a capacidade de pesquisa de GP. Este mecanismo consegue preservar indivíduos de elite num estágio inicial e conduzi-los até um estágio posterior. [1] [75]

A vantagem de GE assenta no facto de permitir ao utilizador especificar e modificar a gramática convenientemente, enquanto ignora a tarefa de desenhar operadores específicos de pesquisa genética. Assim, GE pode ser usado para otimizar dados de movimento pré-existentes ou para gerar novos movimentos. GE é também utilizado para otimizar o problema de redireccionamento da marcha utilizando uma representação Fourier de marcha para codificar o cromossoma e os princípios de similaridade dinâmica como condição.

Uma abordagem híbrida de otimização espaço-tempo e estratégia evolucionária de adaptação por matriz de covariância (CMA-ES) tem sido proposta para gerar marchas e morfologias para a locomoção. Esta abordagem pode combinar a eficiência em espaços de elevadas dimensões e a habilidade para lidar com variáveis gerais da otimização espaço-tempo com a habilidade para lidar com variáveis não-diferenciais e para evitar mínimos locais do CMA. A ideia do CMA-ES básico consiste em amostrar pontos de pesquisa usando uma distribuição normal que está centrada num modelo atualizado da solução ideal. Esta solução ideal pode ser vista como uma média ponderada do melhor sub-conjunto de pontos de pesquisa correntes. A distribuição é também moldada pela matriz de covariância das melhores soluções amostradas na iteração corrente. [1] [76] [77]

Outra ferramenta, nomeadamente os Algoritmos de Estimativa de Distribuição (EDAs) são baseados em modelos probabilísticos que substituem os operadores de mutação e de cruzamento usados em GAs. De fato, é capaz de extrair explicitamente informação estatística global a partir das soluções selecionadas e construir

posteriormente um modelo de distribuição de probabilidades de soluções promissoras baseando-se na informação extraída. A maior vantagem dos EDAs assentam no facto de que o conhecimento sobre o problema adquirido anteriormente poder ser usado para definir o modelo de probabilidade inicial, e a informação estatística global acerca do espaço de pesquisa pode ser extraída diretamente por EDA de modo a modificar o modelo de probabilidade com soluções promissoras. Isto pode reduzir o espaço de pesquisa e obter boas soluções num curto intervalo de tempo. [78]

Em alguns casos de otimização, não se pode medir diretamente nem calcular aspetos da marcha com bases em funções específicas. Nestes casos, preferências, emoções, intuição humana e outros aspetos psicológicos podem ser introduzidos dentro do sistema alvo. Neste contexto, Computação Evolucionária Interativa (IEC) é uma forma de computação onde a função de *fitness* pode ser substituída pelo utilizador. Este método consegue refletir a preferência do utilizador e permite a otimização da solução com uma necessidade mínima de conhecimento sobre o domínio do problema.

Outra estratégia, a Evolução por Estágios, a qual envolve a evolução de resultados num certo número de estágios. Introduzindo um conjunto estagiado de desafios manobráveis, diminui o espaço de pesquisa e deste modo melhora a taxa de convergência da EC e obtém uma rápida evolução do comportamento em direção a um dado objetivo. [1] [75] [79] [76] [80]

4.2.3 NSGA

A presença de objetivos múltiplos num problema, em princípio, dá origem a um conjunto de soluções ótimas (conhecidas como soluções ótimas Pareto), ao invés de uma única solução ótima. Orientando-se para a determinação da região Pareto, um algoritmo evolucionário (EA) pode ser utilizado para descobrir várias soluções Pareto numa única simulação.

O algoritmo genético de classificação não dominada (NSGA) proposto por Srinivas et al. (1995) foi um dos primeiros referidos algoritmos evolucionários. Ao longo dos anos, as maiores críticas em relação à abordagem do NSGA são: [81]

- Elevada complexidade computacional da classificação não dominada: o NSGA correntemente utilizado possui uma complexidade computacional de $O(MN^3)$, onde M corresponde ao número de objetivos e N ao tamanho da população. Deste modo, o NSGA representa um elevado custo computacional para populações de tamanho elevado que deriva da complexidade envolvida no procedimento de classificação não dominada em todas as gerações;
- Falta de elitismo: resultados recentes apresentados por Rudolph et al. e por Zitzler et al. demonstram que elitismo pode acelerar significativamente a performance do GA, podendo também ajudar a prevenir a perda de boas soluções uma vez que sejam encontradas.
- Falta de especificidade do parâmetro de partilha σ_{share} : os mecanismos tradicionais que asseguram a diversidade numa população, no sentido de obter uma ampla variedade de soluções equivalentes, baseiam-se, na maioria, no conceito de partilha. O principal problema é que este aspeto requer uma especificação do parâmetro de partilha (σ_{share}). Este parâmetro está relacionado com a distância métrica escolhida para calcular a medida de proximidade entre dois membros da população.

Deb et al. (2001) propôs uma versão melhorada do NSGA, à qual decidiu designar por NSGA-II. A partir dos resultados de simulações numa série de testes, concluíram que o NSGA-II superava dois outros algoritmos evolucionários de multi-

otimização (MOEAs) contemporâneos, nomeadamente o PAES (Estratégia de evolução Pareto-arquivado) e o SPEA (Algoritmo evolucionário da força-Pareto), ao nível de encontrar um vasto conjunto de soluções e de convergir perto do verdadeiro conjunto ótimo Pareto. No NSGA-II proposto foi substituída a abordagem com função de partilha por uma abordagem *crowded-comparison*, a qual não requer nenhum parâmetro definido pelo utilizador no sentido de garantir diversidade na população. Esta abordagem apresenta menor complexidade computacional e recorre a uma métrica de estimação de densidade e um operador *crowded-comparison*. A estimação da densidade de soluções à volta de uma solução em particular na população é determinada através do cálculo da distância média entre dois pontos em ambos os lados desse ponto ao longo de cada um dos objetivos. Esta distância serve como uma estimativa do perímetro do cuboide formado pelo uso dos vizinhos mais próximos como vértices (*crowding distance*). [82] [83]

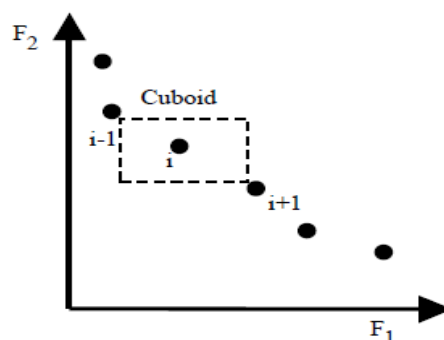


Figura 4. 2 - Cálculo da *crowding-distance*. Os pontos marcados com círculos preenchidos são soluções da mesma frente não dominada. [81]

Na figura 4.2, é possível constatar a *crowding distance* d_t de um ponto i , correspondendo a uma medida do espaço objetivo á volta de i que não é ocupado por qualquer outra solução na população. Neste contexto, calcula-se a quantidade d_t determinando o perímetro do cuboide (a tarcejado) formado usando os vizinhos mais próximos no espaço objetivo. A computação desta variável requer a classificação da população de acordo com cada função objetivo em ordem ascendente de magnitude. O valor global de *crowding-distance* é calculado como a soma de valores individuais de distância correspondentes a cada objetivo.

Com estas inovações, este método possui um carácter elitista, utiliza um mecanismo explícito de preservar a diversidade e dá ênfase a soluções não dominadas.

O elitismo, primeiramente introduzido por De Jong (1975), é um método de seleção que força o GA a reter os melhores indivíduos em cada geração. Deste modo, tais indivíduos não correm o risco de serem perdidos caso não sejam selecionados para reprodução ou de serem destruídos por *crossover* ou mutação. Assim, o carácter elitista garante que as melhores soluções podem passar diretamente para a próxima geração sem ter que se sujeitar a operadores estocásticos. [71] [81] [84]

Neste contexto, o algoritmo do NSGA-II pode ser descrito em vários passos. Inicialmente, uma população aleatória P_0 é criada. Em cada geração t , a população-filho (Q_t) é primeiramente criada usando a população-pai (P_t) e os usuais operadores genéticos. Portanto, as duas populações são combinadas de modo a formar uma nova população (R_t) de tamanho $2N$ e que posteriormente é classificada em diferentes classes de não-dominância. Depois disso, a nova população é preenchida por pontos de várias frentes não dominadas, uma de cada vez, isto é, o preenchimento começa com a primeira frente não dominada e continua com pontos da segunda frente-Pareto, e por aí adiante. Uma vez que a população geral R_t tem um tamanho de $2N$, nem todas as frentes podem ser acomodadas em N *slots* disponíveis para a nova população. Todas as frentes que não podem ser acomodadas são apagadas. Quando a última frente permitida está a ser considerada, podem existir mais pontos na frente que *slots* disponíveis na nova população. Este cenário é ilustrado na figura 4.3. Em vez de descartar aleatoriamente alguns membros da última frente, são selecionados pontos que dêem o máximo nível de diversidade ao conjunto. A classificação por *crowding distance* dos pontos da última frente, que não foi inteiramente acomodada, é conseguida colocando em ordem descendente os seus valores de *crowding distance*, e os pontos que se encontram no topo desta lista são escolhidos. Em cada geração, soluções na população que não são dominadas por qualquer outra solução vão constituir a primeira frente. Depois disso, as soluções da primeira frente são ignoradas temporariamente, e as soluções restantes são processadas. No segundo nível de soluções não dominadas é atribuída um valor de *fitness* superior ao pior valor de *fitness* das soluções da primeira frente. [50] [84] [85] [86] [87]

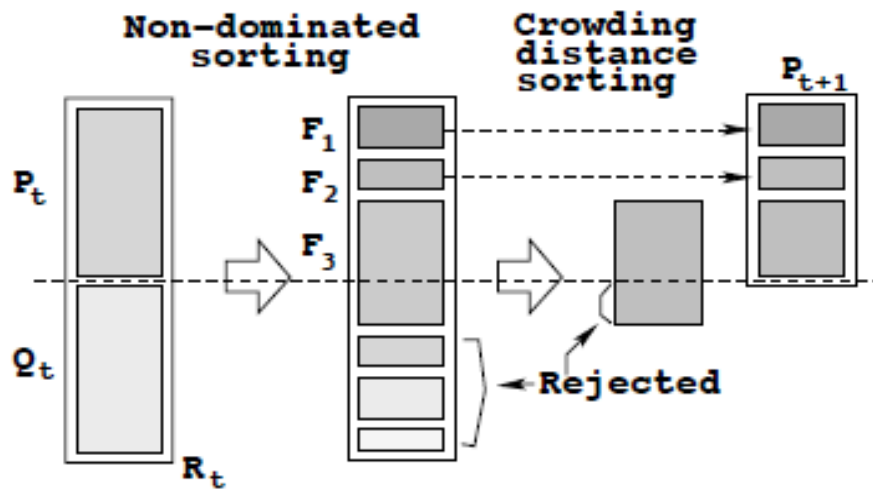


Figura 4. 3 - Esquema representativo do procedimento do NSGA-II. [81]

4.3 Métodos Não evolucionários

Para além dos métodos evolucionários de otimização, existem outras abordagens que visam a pesquisa do ponto ótimo global das funções de custo.

4.3.1 *Reinforcement Learning*

Neste método de otimização, um sistema melhora a sua performance recebendo feedback a partir do que está à sua volta sob a forma de prémio ou penalização proporcional à qualidade da resposta do sistema. Posto isto, o sistema usa este feedback para adaptar o seu comportamento de modo a maximizar a probabilidade de receber tais prémios à medida que o tempo passa. [88]

Dito de outro modo, *Reinforcement Learning* (RL) refere-se a uma classe de algoritmos não supervisionados que visam maximizar um sinal de recompensa numérico. Ao invés de utilizar exemplos de ações corretas como em métodos de aprendizagem supervisionados, este método alcança aprendizagem tentando muitas ações e assimilando as que produzem um sinal com melhor feedback. Um problema geral neste tipo de estratégia é o facto de o sinal de recompensa poder ser atrasado, ou até variável no domínio do tempo, tornando a atribuição de créditos às ações um pouco difícil.

Os elementos necessários para qualquer algoritmo de RL incluem um ambiente, um agente de aprendizagem que tem um objetivo nesse ambiente, um método de selecionar ações, um método de determinar a utilidade imediata de cada ação, e um método de determinar a utilidade a longo prazo das ações realizadas. Além disso, um algoritmo de RL pode incorporar um modelo do seu ambiente. A grande vantagem deste método reside no facto de não requerer exemplos de ações corretas e daí poder ser aplicado em sistemas nos quais estes exemplos ainda não estão disponíveis. Contudo, existem algumas limitações tais como a tendência para terem um processo de aprendizagem muito lento, o que pode conduzir a uma performance muito pobre em ambientes dinâmicos. Outra limitação da RL é o *trade-off* entre o facto de os agentes estarem a tentar atingir o objetivo o mais rápido possível e o facto de estes

deverem procurar aprender mais informação sobre o ambiente à sua volta no sentido de melhorar a sua performance futura. [89]

Neste contexto, dando a possibilidade ao processo de aprendizagem de afetar uma grande porção da marcha, Kohl et al. foram capazes de descobrir configurações de parâmetros que não seriam possíveis descobrir através de configurações à mão.

Além disso, trabalhos recentes sugerem que devido a limitações mecânicas e ruído, o *locus* que cada membro segue é na realidade diferente do *locus* esperado, e uma alteração nos parâmetros que descrevem o *locus* ideal pode não surtir o efeito desejado no *locus* real. Esta discrepância pode tornar a configuração à mão difícil para os utilizadores os quais esperam uma certa correspondência entre os parâmetros e os seus efeitos na marcha. Todavia, os processos de aprendizagem são independentes da semântica dos parâmetros logo estas discrepâncias não são tao relevantes.

Em suma, o método utilizado por Kohl et al. pode ser considerado como um processo de aprendizagem multi-robô no qual é relativamente claro implementar o algoritmo de aprendizagem sem a utilização de um controlador central. Dado que as avaliações, que cada robô faz, são geradas aleatoriamente, o algoritmo que cada robô executa pode correr quase completamente de forma independente dos outros robôs. Kohl aplicou este conceito nos conhecidos robôs AIBO. O único requisito é que cada AIBO comunique as suas avaliações aos outros. [90]

Neste contexto, na literatura, *reinforcement learning* é vastamente aplicado em marchas de robôs de modo a melhorar o comportamento destes. Kirchner et al. aplica *reinforcement learning* a tempo real num robot com 6 pernas no sentido de desenvolver, primeiro, os movimentos elementares de uma perna em particular, e depois, a marcha “tripé” do robô. Por outro lado, este método é aplicado em determinadas situações onde se tem por objetivo maximizar a velocidade. Kimura et al. visa a aprendizagem de posições angulares sequenciais dos atuadores de um robô quadrúpede no sentido de aumentar a velocidade. O algoritmo do RL foi efetuado num robô no qual duas rodas, anexadas à parte traseira, mede o avanço e fornece o sinal de *reinforcement*. Porta et al. usaram RL para maximização da velocidade e os resultados obtidos demonstraram que a estratégia com RL é mais eficiente que a geração de marcha livre reativa por programação feita à mão. Janusz et al. aplicaram RL

combinado com redes neurais a um robô móvel de modo a selecionar a velocidade das rodas no sentido de evitar colisões. [91] [92]

Svinin et al. usaram RL no sentido de adquirir a aprendizagem dos padrões sequenciais de ação que resultam numa marcha reta e estável. Nesta aplicação, a queda do robô resulta num *reinforcement* negativo, e o algoritmo aprende a evitar a mesma situação. Este autor sugere ainda a implementação do procedimento de aprendizagem num simulador, e depois de conseguida a aprendizagem com êxito, faz-se o *download* no *hardware* do robô. Neste contexto, esta aplicação é puramente simulativa uma vez que procedimentos de aprendizagem podem requerer muito tempo se aplicado num robô atual em tempo real, uma vez que implica a necessidade de um vasto número de ciclos de treino. No sentido de fazer face a este problema, Erden et al. aplica RL, não a partir de uma tábua rasa, mas sim com um algoritmo peculiar de geração de marcha livre que garante estabilidade para todos os estados. Tal abordagem, não garante apenas uma marcha estável em condições normais, mas também provoca uma aprendizagem mais rápida mesmo quando a estabilidade não é garantida quando uma das pernas não está a funcionar correctamente. [91] [93]

A diferença entre este método e o GA é que um GA é um modelo de um processo de evolução de espécies de animais, e basicamente o parâmetro é atualizado selecionando indivíduos com maior aptidão, onde esta é determinada pelas interações com o ambiente durante todo o tempo de vida de cada indivíduo. Por outro lado, *reinforcement Learning* tenta modelar o mecanismo de aprendizagem dos indivíduos e o parâmetro é basicamente atualizado em cada interação com o ambiente. [94]

4.3.2 *Particle Swarm Optimization*

Particle Swarm Optimization (PSO) é uma abordagem de otimização com características estocásticas que imitam o comportamento de grupos de animais. Introduzida por Kennedy e Eberhart, esta estratégia tem como ideia de base a utilização dos comportamentos de grupo de animais para resolver problemas numéricos de otimização em espaços de pesquisa multi-dimensionais. Cada membro do grupo, denominado por partícula, possui uma posição e velocidade dentro dos limites do espaço de pesquisa e exibe uma possível solução para o problema. Em cada iteração, as partículas testam a sua posição e colecionam experiência sobre o espaço de pesquisa. Depois de cada iteração, cada partícula comunica a sua melhor posição conhecida a outras partículas. A nova posição destas na próxima iteração é baseada no seu próprio conhecimento, na informação proveniente das outras partículas, e a sua última posição e velocidade. Através da informação recolhida e a sua utilização por todo o grupo, é alcançada uma pesquisa direcional e paralela pela situação ótima. O peso estocástico das diferentes informações disponíveis fornece o carácter estocástico do processo de otimização. [33] [95]

Alguns autores concluíram que PSO é melhor que os GA como método de aprendizagem para a otimização da marcha num ambiente não-determinístico pois referem que PSO otimiza as séries de Fourier truncadas (TSF) significativamente mais rapidamente e melhor para gerar locomoção mais reta e rápida uma vez que PSO evita mínimos locais onde os GA ficaram retidos.

Todavia, os GAs não precisam de ser necessariamente inferior ao PSO na otimização da marcha, uma vez que o PSO empregue na experiência levada a cabo por esses autores correspondia a PSO adaptativo, o qual tem um parâmetro de inércia não linear e dinamicamente ajustável w que controla o equilíbrio entre exploração global e local. Um maior peso de inércia facilita a pesquisa global, enquanto um menor peso de inércia facilita a pesquisa local. A EC pode obviamente empregar o mesmo mecanismo para melhorar a sua eficiência, uma vez que GA Adaptativo pode variar de forma adaptativa as probabilidades de cruzamento e mutação durante o processo de evolução.

O PSO adaptativo é usado para otimizar as marchas mais rápidas do AIBO. Começando com parâmetros gerados aleatoriamente em vez de parâmetros feitos à mão, são obtidos vários conjuntos de parâmetros com alto nível de performance, sendo que estes exemplos de marchas foram referidos como as mais rápidas alguma vez desenvolvidas para a mesma plataforma do robô.

Alguns autores não recomendam o uso do algoritmo de PSO para resolver problemas de otimização do movimento em larga escala que possuam condições ou termos competitivos na função de custo. E sugerem também que pode ser necessário um algoritmo híbrido (local e global) para o PSO e outros otimizadores globais para resolver eficientemente problemas de movimento em larga escala. [1] [96]

Em suma, tanto EC como a SI já provaram ser ótimas ferramentas para otimização de locomoção de robôs. Quer a abordagem de EC quer a abordagem de *swarm intelligence* (SI) são algoritmos iterativos baseados em populações, embora adotem metáforas diferentes. Assim, ambas partilham as mesmas vantagens e desvantagens, como por exemplo, uma capacidade de pesquisa global e de elevada dimensão similar, capacidade de otimização multi-objetivo, assim como uma série de parâmetros de controlo que requerem monitorização.

Apesar de haver algum empenho em comparar performances entre métodos de EC e outras abordagens de otimização global não evolucionária, nenhum estudo comparativo sistemático tem sido feito. Tal estudo pode não ser necessário ou exequível visto que o objetivo é, na maioria das vezes, encontrar um conjunto de soluções e não uma solução absoluta global ótima. Quer a plataforma do robô quer as funções objetivo da otimização da marcha serão diferentes em cada caso, e conseqüentemente, é difícil encontrar um robô de referência e um conjunto de funções objetivo de referência para otimizar. [1]

5 Proposta de Otimização

Neste capítulo é descrita a nossa proposta de otimização, sendo referidos todos os passos principais, nomeadamente, o método de geração de marcha, as funções objetivo utilizadas e estratégia de otimização.

5.1 Geradores Centrais de Padrões

Os CPGs são modelos neuronais capazes de gerar padrões de atividade rítmica essenciais para a execução de diversas ações rítmicas motoras, como por exemplo a locomoção. Os CPGs na locomoção humanóide consistem numa rede neuronal composta por vários osciladores acoplados, em que cada oscilador orienta uma determinada junta das pernas ou dos braços. [97]

A elaboração deste processo de geração de movimento e respetiva descrição é baseada na literatura [98] e [6].

Neste contexto, a base da nossa proposta de otimização, nomeadamente a geração de padrões de movimento consiste numa rede organizada de unidades geradoras em que cada junta está associada a uma dessas unidades. Deste modo, cada uma destas unidades é implementada por meio de equações diferenciais não lineares e é responsável pela ativação de uma junta. Por outro lado, cada unidade geradora é composta por um gerador de padrão de movimento orientado por osciladores de fase que geram o ritmo de base para o gerador de padrões.

A geração de movimento na nossa proposta considerou como ponto de partida uma série de primitivas de movimento que são incrementadas progressivamente, utilizando trajetórias sinusoidais e em forma de sino para descrever movimentos gerais observados a partir da marcha humana. Esta abordagem visa obter uma marcha básica e eficiente, e é vantajosa na medida em que permite a fácil adição e mudança das primitivas de movimento, o que permite a implementação de mecanismos autónomos tais como aprender com a experiência ou evolução de comportamentos locomotores.

5.2 Geração de Ritmos

As unidades geradoras propostas possuem um oscilador de fase para produzir o ritmo para padrões de movimento, traduzindo-se na seguinte equação diferencial:

$$\dot{\phi}_i = \omega, \quad (1)$$

onde $\phi_i \in [-\pi, \pi]$ (rad.s^{-1}) é a fase do oscilador, a qual aumenta linearmente com ω . Este oscilador sincroniza a geração de movimentos rítmicos de uma perna i (esquerda ou direita).

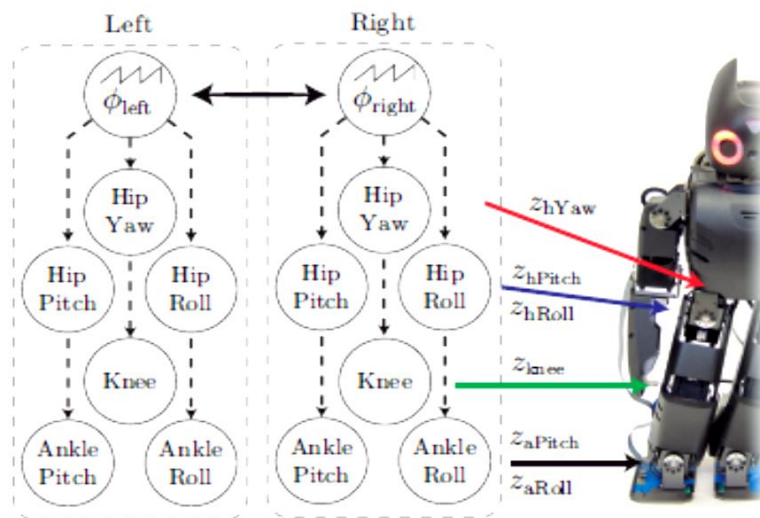
Através do acoplamento dos osciladores de fase, conseguimos manter e especificar diferenças de fase entre as unidades geradoras, adicionando um termo na equação (1), o qual permite especificar uma diferença de fase de φ entre a unidade da perna direita e a unidade da perna esquerda com uma força de acoplamento de k .

$$\dot{\phi}_{dir} = \omega + k \sin(\phi_{dir} - \phi_{esq} + \varphi), \quad (2)$$

$$\dot{\phi}_{esq} = \omega + k \sin(\phi_{esq} - \phi_{dir} + \varphi), \quad (3)$$

onde $\phi_{esq} \in [-\pi, \pi]$ (rad.s^{-1}) é a fase do oscilador da perna esquerda, $\phi_{dir} \in [-\pi, \pi]$ (rad.s^{-1}) é a fase do oscilador da perna direita e $\varphi \in [-\pi, \pi]$ (rad.s^{-1}) corresponde a uma diferença de fase. Na implementação definimos que φ é igual a π .

Na nossa proposta de locomoção, as unidades geradoras estão acopladas entre elas de acordo com a rede apresentada na figura 5.1.



5. 1 - Unidades geradoras, juntas correspondentes e acoplamentos intra-membros e inter-membros correspondentes. Acoplamento bilateral está representado pela seta bilateral e acoplamento unilateral está representado por setas unilaterais. [98]

5.3 Geração de Movimento

A posição da junta $z_i(t)$ é gerada de acordo com a fase corrente do CPG, através da equação não-linear (4).

$$\dot{z}_{i,j} = -\alpha(z_{i,j} - O_{i,j}) + \sum f_j(z_{i,j}, \phi_i), \quad (4)$$

onde $O_{i,j}$ especifica o *offset* do movimento rítmico gerado e α corresponde ao parâmetro de relaxação para o *offset*. Cada função $f_j(z, \phi_i)$ especifica uma primitiva de movimento para o movimento gerado de uma junta, e a trajetória final gerada resulta da soma de todas as primitivas de movimento. Por outro lado, i define se é a perna esquerda ou a perna direita e o j define o gerador de movimento de uma junta na perna, como por exemplo, hip yaw (hYaw), ankle pitch (aPitch), entre outros.

A formação de padrões de movimento recebe um sinal rítmico e gera os padrões motores através da soma de primitivas de movimento, as quais são codificadas como um conjunto de equações não-lineares. São também suave e facilmente moduladas no que diz respeito às suas amplitudes, frequências, e padrões de *offset*.

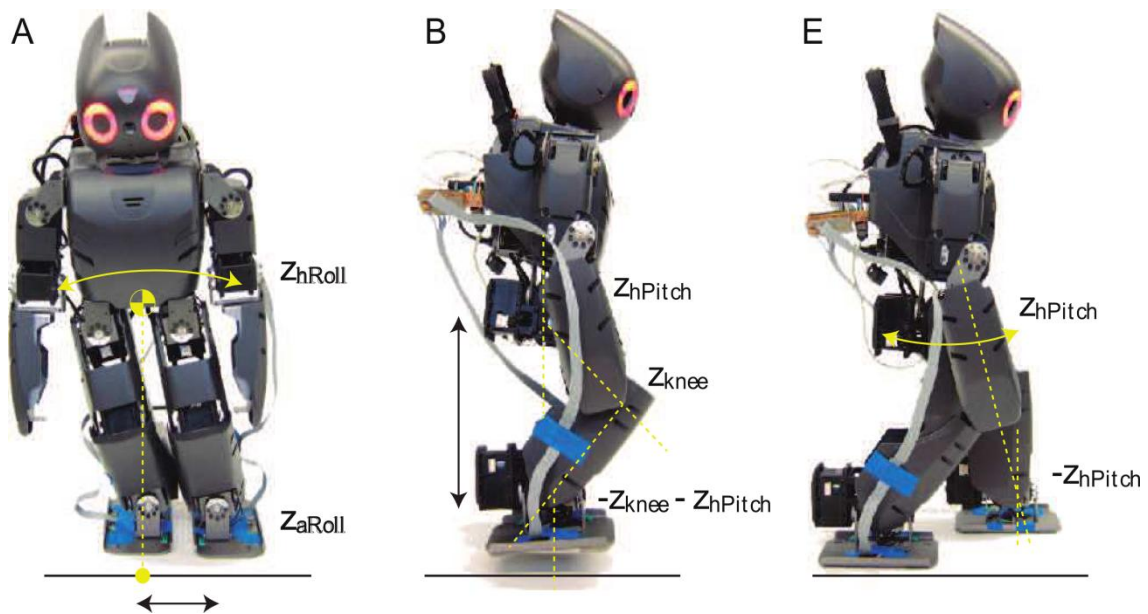
5.3.1 Primitivas de Movimento

Tendo a marcha humana como exemplo, observações e descrições cinemáticas permitem destacar primitivas de movimento as quais constituem um repertório motor básico essencial para conseguir obter comportamentos locomotores básicos e eficazes. Cada primitiva de movimento corresponde a uma movimento básico da locomoção e pode ser representado por equações que se traduzem em curvas sinusoidais e em forma de sino. Alguns desses movimentos são ilustrados na figura 5.2.

Um movimento sinusoidal geral com amplitude A_m e desvio de fase Ψ_m , é gerado da seguinte forma para a junta j :

$$f_j^m = -A_m \omega \sin(\phi_i + \Psi_m), \quad (5)$$

em que m corresponde aos vários tipos de movimento que são feitos e que são descritos a seguir.



5. 2 - Primitivas de movimento: A) movimento de *balancing*; B) movimento de flexão; E) movimento de compasso. [98]

Por outro lado, um movimento em forma de sino pode ser descrito pela seguinte equação:

$$f_j^m = \frac{A_m \omega (\phi_i + \Psi_m)}{\sigma^2} e^{\left(-\frac{(\phi_i + \Psi_m)^2}{2\sigma^2}\right)}, \quad (6)$$

Na equação 6, A_m e σ definem a amplitude e largura da curva, e Ψ_m define o desvio de fase e, m corresponde aos vários tipos de movimento que são feitos. Neste contexto, os movimentos f_j^m são os mesmos para ambas as pernas. Por outro lado, atribuímos movimentos simétricos ao tornozelo em relação à anca de modo a manter sempre os pés paralelos ao chão.

5.3.1.1 Movimento de Balancing

Em [22], este movimento é mencionado como um dos seis determinantes da marcha humana e corresponde a um deslocamento lateral da pélvis visto como o balançar da anca (*balancing*). Assim, este movimento desloca o corpo lateralmente e simetricamente duas vezes durante o ciclo da marcha no sentido de posicionar corretamente o centro de massa. Quando o humanoide se movimenta dá passos alternadamente, e isso implica o posicionamento do COM (centro de massa) sobre o polígono de suporte durante o ciclo da marcha. O correto posicionamento do COM é essencial na medida em que permite a perna contralateral fletir e executar a fase *swing* do passo. Caso não seja alcançado o deslocamento correto do COM sobre a perna de suporte no *timing* correto, o bípede pode cair. É de referir também que na marcha humana, as pernas não estão paralelas, mas sim ligeiramente com um determinado ângulo para dentro. Esta característica reduz o deslocamento lateral do COG que acontece durante locomoção em linha reta.

No Darwin, nós conseguimos obter esse movimento, atuando nas juntas *hip roll* e *ankle roll*, como uma trajetória sinusoidal que faz o robô oscilar lateralmente. Este movimento pode ser descrito nas seguintes equações:

$$f_{hroll}^{balancing} = -A_{balancing}\omega\sin(\phi_i), \quad (7)$$

$$f_{aroll}^{balancing} = -f_{hroll}^{balancing}, \quad (8)$$

onde i especifica se é a perna esquerda ou direita, ϕ_i é a fase do CPG, o parâmetro $A_{balancing}$ determina a amplitude do deslocamento lateral e $\Psi_{balancing}$ é nulo (figura 5.2-A). O ângulo para dentro existente nas pernas referido anteriormente é especificado pelos parâmetros de *offset* $O_{i,hRoll}$, e $O_{i,aRoll}$ nas equações 17 e 18.

5.3.1.2 Movimento de Flexão

Como referido anteriormente, um *balancing* bem executado permite a execução do *swing* da perna que não serve de suporte. Este *swing* corresponde ao levantar da perna contralateral à perna de suporte e implica um movimento de flexão, sendo executado alternadamente entre as duas pernas. O deslocamento vertical derivado do levantar do pé é conseguido através da atuação em três juntas de *pitch*: tornozelo, joelho e anca (figura 5.2-B).

Este movimento ao nível das juntas pode ser descrito como tendo uma curva em formato de sino (equação 6 com $\Psi_m = 0$ e $\phi_i \in [-\frac{\pi}{2}, \frac{\pi}{2}]$), resultando num perfil suave para a trajetória da flexão. Isto é demonstrado na equação (9) para a anca, na equação (10) para o joelho, e na equação (11) para o tornozelo.

$$f_{hpitch}^{flexão} = \frac{A_{hip}\omega\phi_i}{\sigma^2} \times e^{\left(-\frac{\phi_i^2}{2\sigma^2}\right)} \quad (9)$$

$$f_{kpitch}^{flexão} = -\frac{A_{knee}\omega\phi_i}{\sigma^2} \times e^{\left(-\frac{\phi_i^2}{2\sigma^2}\right)} \quad (10)$$

$$f_{apitch}^{flexão} = -(f_{hpitch}^{flexão} + f_{kpitch}^{flexão}) \quad (11)$$

A amplitude da trajetória em forma de sino é determinada pelo parâmetro A_{hip} para a anca e pelo parâmetro A_{knee} para o joelho. A trajetória descrita pelo tornozelo é a soma dos movimentos protagonizados pela anca e pelo joelho e deste modo os pés mantêm-se paralelos. O parâmetro σ especifica a duração da flexão da perna durante o ciclo do passo. Nós pretendemos ter uma fase de *swing* global de cerca de 50% de todo o ciclo do passo, e por isso escolhemos $\sigma = \frac{\pi}{6}$.

5.3.1.3 Movimento Yield do joelho

Este movimento diz respeito ao momento da locomoção em que a perna inicia a fase *stance* e o peso do corpo é transferido para a junta do joelho. Uma característica desse movimento consiste no rendimento (*yield*) do joelho. Adicionando uma pequena flexão na junta do joelho durante a fase do *stance*, reduzimos a trajetória vertical do COM do Darwin.

Este movimento caracteriza-se por um perfil sinusoidal de amplitude A_{yield} , e desvio de fase π para a flexão ocorrer a meio da fase de *stance*. Este movimento é adicionado às juntas do joelho e do tornozelo e é descrito através das equações (12) e (13):

$$f_{kPitch}^{yield} = -A_{yield}\omega\sin(\phi_i + \pi) \quad (12)$$

$$f_{aPitch}^{yield} = -f_{kPitch}^{yield} \quad (13)$$

5.3.1.4 Movimento de Rotação

Este movimento corresponde a uma rotação alternada da pélvis que reduzir a trajetória vertical do COG, e por suavizar as inflexões aquando da variação da direção vertical do COG. Esta rotação pélvica é executada nas juntas yaw da anca, e é descrita como uma trajetória sinusoidal com um desvio de fase de $\frac{\pi}{2}$. A adição deste movimento permite uma locomoção mais eficaz e sobretudo mais semelhante à marcha humana. Este movimento é descrito pela equação (14):

$$f_{hYaw}^{Rotação} = -A_{Rotação}\omega\sin(\phi_i + \frac{\pi}{2}) \quad (14)$$

A amplitude da rotação pélvica descrita na equação (14) é especificada pelo parâmetro $A_{Rotação}$.

5.3.1.5 Movimento de Compasso

O movimento final, responsável por garantir que as pernas dão passos para a frente, consiste em mover as pernas no plano sagital, movendo alternadamente as pernas contralaterais para a frente e para trás, tal como um compasso (figura 5.2-E). Este movimento é descrito com perfis sinusoidais nas juntas *pitch* da anca com amplitude $A_{compasso}$ e desvio de fase de $\frac{\pi}{2}$:

$$f_{hPitch}^{compasso} = -A_{compasso}\omega\sin(\phi_i + \frac{\pi}{2}) \quad (15)$$

$$f_{aPitch}^{compasso} = -f_{hPitch}^{compasso} \quad (16)$$

Similarmente ao movimento de *balancing*, são atribuídos movimentos simétricos às juntas do tornozelo e da anca.

Ao adicionar e ajustar sequencialmente cada movimento, o resultado global de todos os movimentos é uma marcha bípede reta.

O movimento de cada junta é descrito pela equação (4), e por conseguinte é feito a seguir o sumário de todos os movimentos das várias juntas pelas seguintes equações:

$$\dot{z}_{i,hRoll} = -\alpha(z_{hRoll} - O_{hRoll}) + f_{hRoll}^{balancing}, \quad (17)$$

$$\dot{z}_{aRoll} = -\alpha(z_{aRoll} - O_{aRoll}) + f_{aRoll}^{balancing}, \quad (18)$$

$$\dot{z}_{i,hYaw} = -\alpha(z_{hYaw} - O_{hYaw}) + f_{hYaw}^{Rotação}, \quad (19)$$

$$\dot{z}_{i,hPitch} = -\alpha(z_{hPitch} - O_{hPitch}) + f_{hPitch}^{flexão} + f_{hPitch}^{compasso}, \quad (20)$$

$$\dot{z}_{i,kPitch} = -\alpha(z_{kPitch} - O_{kPitch}) + f_{kPitch}^{flexão} + f_{kPitch}^{yield}, \quad (21)$$

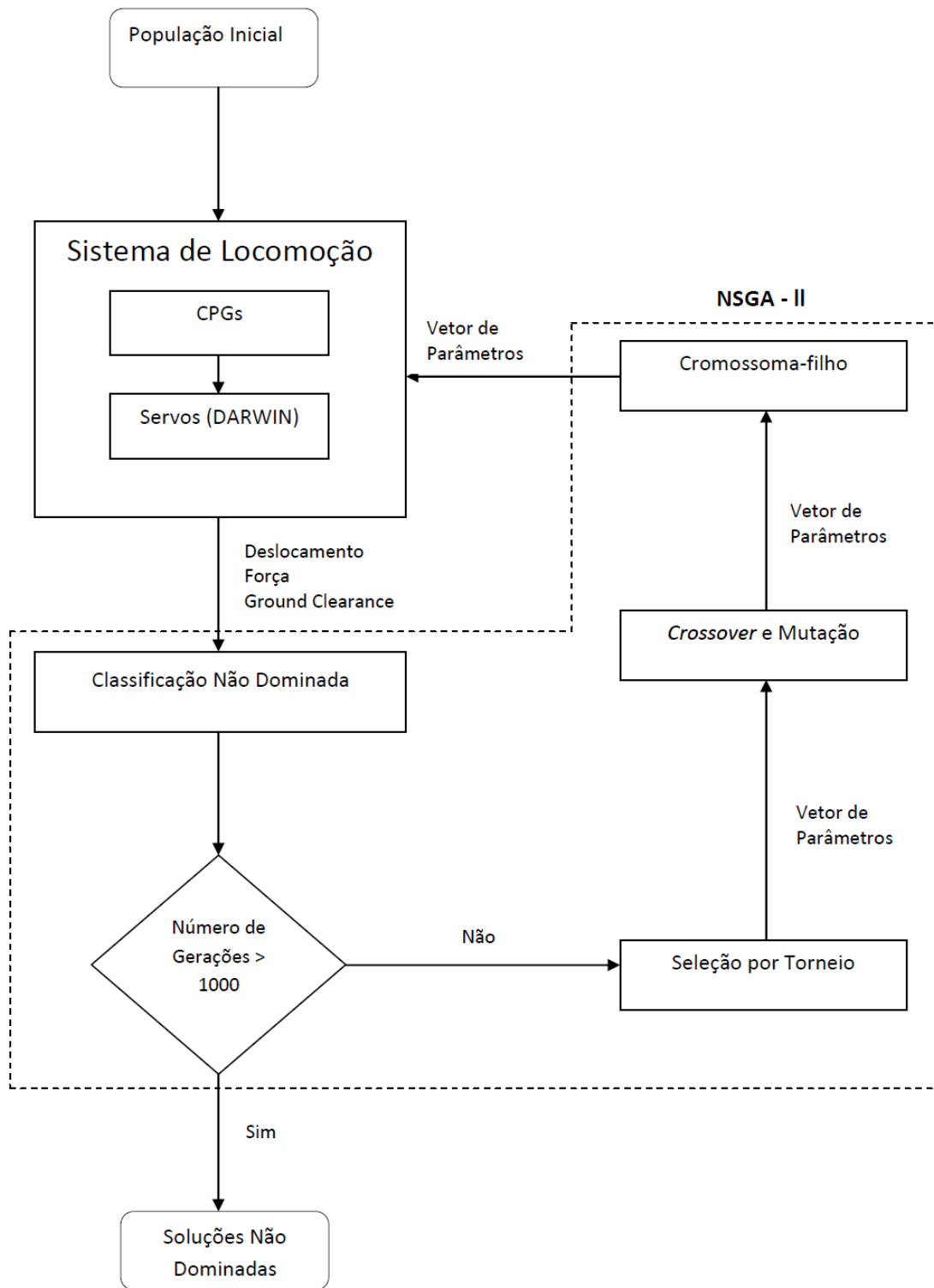
$$\dot{z}_{i,aPitch} = -\alpha(z_{aPitch} - O_{aPitch}) + f_{aPitch}^{flexão} + f_{aPitch}^{yield} + f_{aPitch}^{compasso} \quad (22)$$

5.4 Sistema de Otimização

O objetivo primordial deste sistema de otimização é conseguir gerar trajetórias para as juntas do Darwin que obtenham a melhor marcha possível avaliada de acordo com alguns critérios. No sentido de alterar as trajetórias geradas, é necessário alterar as primitivas de movimento através da variação dos parâmetros CPG. Neste contexto, o comportamento motor final depende na série de parâmetros de CPG referidos anteriormente.

Para conseguir gerar locomoção bípede num robô foi considerado um método de evolução por três fases: fase de *balancing*, fase de *stepping* e fase de *walking*. Cada estágio é executado durante um determinado período de tempo não havendo interrupção entre cada transição de fase. Para cada movimento, existe um ou mais parâmetros a serem otimizados.

No sentido de determinar corretamente os parâmetros, de modo a obter a performance desejada, recorreu-se a uma abordagem evolucionária. Neste contexto, foi utilizado como algoritmo de otimização o NSGA-II (*Non-dominated sorting genetic algorithm*) no sentido de encontrar um conjunto ótimo de parâmetros de CPG, isto é, um vetor de parâmetros otimizado. O sistema de otimização encontra-se esquematizado na figura 5.3.



5.3 - Fluxograma representativo da proposta de otimização da locomoção.

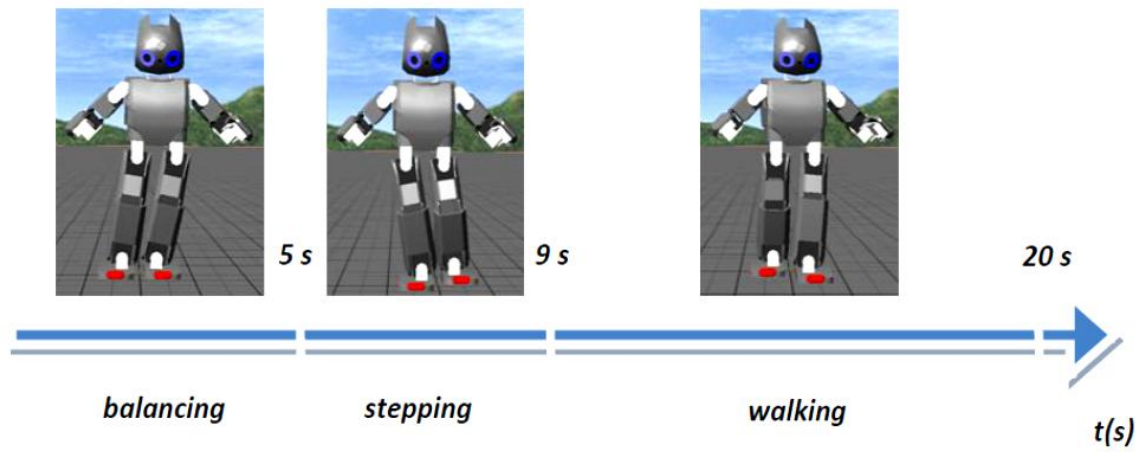
5.4.1 Vetor de Parâmetros

Neste trabalho, o vetor de parâmetros ou cromossoma possui 13 genes: as amplitudes ($A_{balancing}$, A_{hip} , A_{knee} , $A_{Rotação}$, A_{yield} , $A_{compasso}$), e os *offsets* (O_{hRoll} , O_{hPitch} , O_{kPitch} , O_{aRoll} , O_{aPitch}), o σ e a frequência ω .

5.4.2 Evolução por Fases

Primeiro, o robô desempenha os movimentos de *balancing* e de rotação da pélvis durante o intervalo de tempo t_b . Se o robô conseguir concretizar a primeira fase com sucesso, a avaliação prossegue para a segunda fase onde o robô executa os movimentos anteriores assim como os movimentos de flexão e de *yield* durante o intervalo de tempo t_s . Caso esta fase seja concretizada com sucesso, o sistema de otimização prossegue para a terceira fase, onde todos os movimentos são executados durante o intervalo de tempo t_w . Deste modo, recorreremos a uma avaliação sequencial onde cada um dos passos possui diferentes funções objetivo. Por outro lado, no sentido de diferenciar os vários passos de avaliação, é introduzida uma restrição de tempo, isto é, cada passo ou estágio de avaliação inicia-se quando o estágio anterior está terminado:

- Fase de *balancing*: durante 5 segundos ($t \leq 5s$) é executado os movimentos de *balancing* e de rotação da Pélvis ($t_b=5s$);
- Fase de *stepping*: posteriormente ($5s \leq t \leq 9s$) são adicionados os movimentos de flexão e de *yield* ($t_s=4s$);
- Fase de *walking*: durante alguns segundos ($9s \leq t \leq 20s$) são executados todos os movimentos ($t_w=11s$);



5. 4 - Diagrama representativo das três fases da estratégia de evolução.

5.5 Determinação dos Objetivos

Foram considerados quatro funções objetivo de modo a avaliar a performance das soluções obtidas para a geração de locomoção: a posição do centro de pressão (COP), a força, o deslocamento vertical dos pés (*ground clearance*) e o deslocamento frontal do robô. A escolha e elaboração destas funções objetivos foram desenhadas para se adequarem aos objetivos que tínhamos definido para otimização da locomoção, nomeadamente melhorar a estabilidade e velocidade, diminuir a força de impacto do pé com o chão e fazer com que o humanoide levante mais os pés. A elaboração das funções objetivo foi também baseada na experiência advinda da realização de testes e posterior análise dos resultados obtidos.

5.5.1 Centro de Pressão

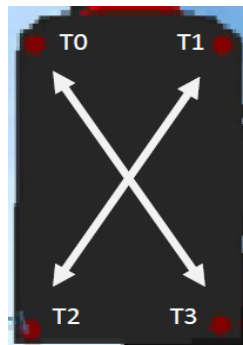
As funções objetivo desta proposta de otimização utiliza uma medida de estabilidade como função objetivo, nomeadamente o centro de pressão (COP). A posição do COP, quando o pé se encontra em contato com o chão, indica se o robô vai cair ou não. Por exemplo, se o pé esquerdo está no chão e o COP está perto do limite do polígono de suporte do pé, o robô pode cair. Assim, durante o primeiro passo da evolução por fases, é maximizada a distância média do COP em relação ao limite do polígono de suporte do pé quando este se encontra na fase de *stance*. O COP é calculado através da equação (22):

$$COP = \sqrt{f^2 + l^2}, \quad (22)$$

onde f e l representam as distâncias designadas por *frontal distance* e *lateral distance* ilustradas na figura 5.6.

5.5.2 Força

Quando inicialmente determinamos uma locomoção com parâmetros ajustados à mão, constatou-se que o impacto dos pés com o chão, na fase inicial do fase de suporte (*stance*), estava a causar oscilações indesejadas no robô bípede. Deste modo, criamos uma nova função objetivo (Força), que diz respeito à força de impacto entre os pés do *Darwin* e o solo. Esta variável é medida através de sensores de força que se encontram na planta dos pés do humanoide (quatro em cada um dos pés). Deste modo, esta função assenta em três objetivos principais: ter o pé paralelo ao chão, minimizar o impacto da força quando os pés batem no chão e fazer com que essa força de impacto esteja igualmente distribuída pelos quatro sensores do pé.



5. 5 - Esquema representativo da localização dos sensores de força (T0, T1, T2 e T3) na planta do pé do Darwin e das duas diagonais entres eles

No sentido de manter o pé paralelo ao chão, é verificado se sensores correspondentes a uma das diagonais estão a tocar no chão. Neste contexto, se os sensores T0 e T3 devolvem valores de força não nulos, significa que todo o pé está em contacto (paralelo) com o solo. Inicialmente apenas se considerou uma das diagonais todavia após uns testes iniciais, foi adicionada a outra diagonal à função de modo a evitar situações em que um dos sensores não funcione corretamente e dê um falso negativo. Assim, verifica-se as duas diagonais e basta ter uma das diagonais ativas (dois sensores posicionados numa das diagonais que estejam assentes no chão) para considerarmos que o pé está paralelo ao chão. Tal é efetuado através da seguinte função, a qual se pretende minimizar:

$$F = (\overline{T0} + \overline{T3}).(\overline{T1} + \overline{T2}).K + S, \quad (23)$$

onde K corresponde a um valor determinado manualmente (tem de ser um valor elevado pois tem de garantir que uma má postura devolva um valor de F elevado) e S é calculado através da equação (24).

Deste modo, ao definirmos um valor de K elevado em relação a S , a função vai assumir um valor aproximado de K nas situações em que o pé não está totalmente pousado no chão, e nas situações em que o pé está assente totalmente no chão, a função assume o valor de S . A tabela 5.1 descreve os valores de F para todas posturas possíveis dos pés. Esta função leva o robô a corrigir a forma como posiciona os pés, pousando-os completamente no chão, uma vez que quando isso não acontece, F assume um valor elevado, e uma vez que esta é uma função de minimização, vai tentar alterar a postura do pé de modo a minimizar o valor de F .

Tabela 5. 1 - Valores assumidos por F dependendo das várias posturas dos pés.

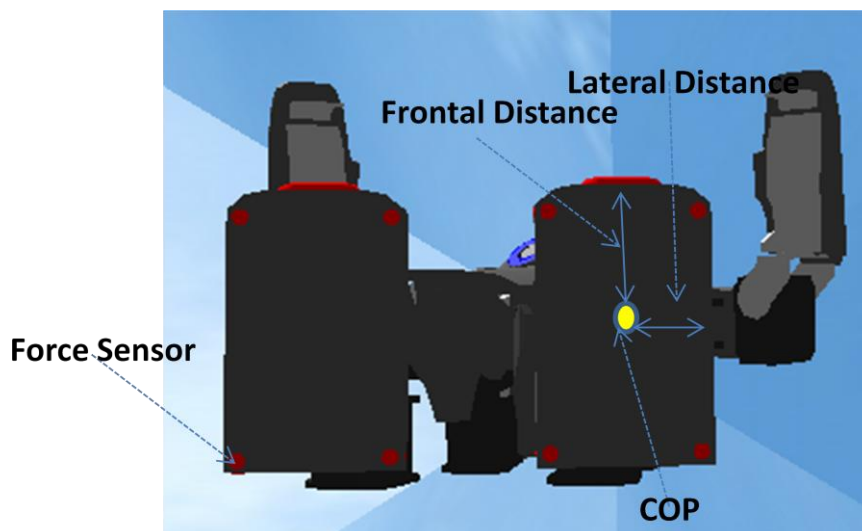
T0	T1	T2	T3	$\overline{T0} + \overline{T3}$	$\overline{T1} + \overline{T2}$	F
0	0	0	0	1	1	$K + S \approx K$
0	0	0	1	1	1	$K + S \approx K$
0	0	1	0	1	1	$K + S \approx K$
0	0	1	1	1	1	$K + S \approx K$
0	1	0	0	1	1	$K + S \approx K$
0	1	0	1	1	1	$K + S \approx K$
0	1	1	0	1	0	S
0	1	1	1	1	0	S
1	0	0	0	1	1	$K + S \approx K$
1	0	0	1	0	1	S
1	0	1	0	1	1	$K + S \approx K$
1	0	1	1	0	1	S
1	1	0	0	1	1	$K + S \approx K$
1	1	0	1	0	1	S
1	1	1	0	1	0	S
1	1	1	1	0	0	S

Por outro lado, a variável S diz respeito ao controlo da força de impacto em duas vertentes: a força de impacto é minimizada e tem de ser igualmente distribuída pelos quatro sensores do pé. O cálculo de S encontra-se descrito através da equação (24):

$$S = |T0 - T3| + |T1 - T2| + \sum |Ft - 286.5| \quad (24)$$

As duas primeiras parcelas verificam se a força de impacto está igualmente distribuída pelos quatro sensores. Deste modo, correspondem à soma das diferenças entre as forças de impacto entre os dois sensores de cada diagonal. Caso a força de impacto esteja igualmente distribuída, esta diferença é praticamente nula, e consequentemente, a soma é nula, logo não vai aumentar o valor de F . Todavia, caso a força esteja mal distribuída obtém-se diferenças entre os sensores $T0$ e $T3$, e/ou os sensores $T1$ e $T2$ e, consequentemente o valor de F aumenta proporcionalmente com essa diferença. Deste modo, o algoritmo vai corrigir essa má distribuição, penalizando essa postura, de modo a minimizar os valores de F .

Quanto ao objetivo final, a força de impacto é minimizada através da minimização da diferença verificada entre o valor correspondente ao somatório das forças medidas nos quatro sensores e o valor do peso total do robot (286.5).



5. 6 - Perspetiva de baixo do Darwin. Especificação da posição dos sensores de força embebidos nos pés do robô e de algumas medidas específicas tidas em conta na *fitness*

5.5.3 Deslocamento

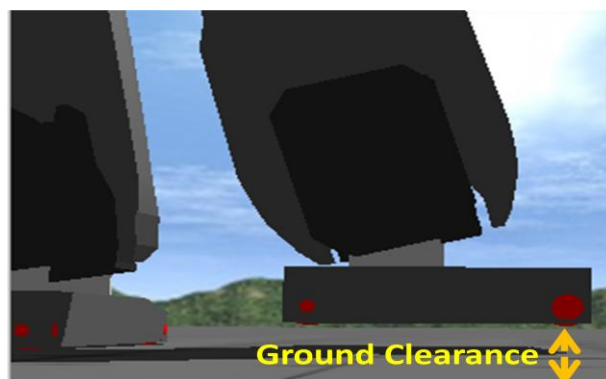
Esta função objetivo tem o papel de otimizar o deslocamento frontal e penalizar eventuais deslocamentos laterais do *Darwin*. Este deslocamento corresponde à distância percorrida durante um intervalo de 11 segundos.

$$\text{deslocamento} = \sqrt{(x_f + x_0)^2 + (z_f + z_0)^2} - \sqrt{(x_f + x_0)^2}, \quad (25)$$

onde x_0 e z_0 são as coordenadas do robô antes de iniciar a marcha e x_f e z_f são as coordenadas finais do robô (x corresponde ao deslocamento lateral e z corresponde ao deslocamento frontal).

5.5.4 Ground Clearance

Por fim, recorreu-se a mais uma função objetivo, designada por *ground clearance*, a qual tem por objetivo maximizar a distância entre o pé e o solo. Deste modo, é calculada a distância entre o sensor do pé, que se encontra a uma altura mais baixa, e o chão, durante o movimento de *swing*, quando o pé se encontra na posição mais afastada do chão.



5.7 - Esquema representativo do *ground clearance*

5.6 Enquadramento dos Objetivos na Evolução por Fases

A performance de cada combinação de parâmetros CPG é avaliada em termos de índices de maximização ou minimização de objetivos. Por outro lado, cada estágio de evolução possui objetivos diferentes e consequentemente, possui diferentes funções objetivo. Neste contexto, para cada fase, temos as seguintes funções objetivo:

- Fase de *balancing*: Centro de pressão (maximização) e Força (minimização);
- Fase de *stepping*: Ground Clearance (maximização) e Força (minimização);
- Fase de *walking*: Ground Clearance (maximização), Deslocamento (maximização) e Força (minimização)

A interação entre estes objetivos dá origem a um conjunto de soluções eficientes conhecidas como soluções ótimas Pareto. As soluções em problemas multi-objetivo são comparadas em termos de dominância sendo este processo descrito na seção 4.1.1. da dissertação. A meta final é conseguir que todos os conjuntos de parâmetros de CPG resultem em movimentos que executam todos os estágios, ou seja, que todos os indivíduos executem as três fases com sucesso.

5.7 Tratamento de Restrições

Na geração de marcha, as novas soluções são geradas com base na otimização dos vetores de parâmetros CPGs. No entanto, estes parâmetros não abrangem um espaço infinito tendo na verdade que respeitar determinados limites que definem o espaço objetivo. No sentido de lidar com simples restrições de limites, cada novo ponto gerado é projetado no sentido de satisfazer os limites das restrições da seguinte forma:

$$x_i = \begin{cases} l_i & \text{se } x_i < l_i \\ x_i & \text{se } l_i \leq x_i \leq u_i \\ u_i & \text{se } x_i > u_i \end{cases}$$

Onde l_i e u_i são o limite inferior e superior da componente i respectivamente. No sentido de lidar com condições com inequações, foi empregue o método de seleção por torneio proposto por Deb em [99]. Para efeitos de comparação, é usado seleção por torneio para ter a certeza que: quando duas soluções viáveis são comparadas, é escolhida a que possui melhor valor de função objetivo; quando uma das duas soluções não é viável, é escolhida a que é fiável; e quando duas soluções não viáveis são comparadas, é escolhida aquela que menos infringe os limites das restrições. Assim, devem ser impostos limites restritivos às variáveis de decisão de acordo com a gama de pesquisa da rede de parâmetros de CPGs. Isto depende da plataforma utilizada, a qual foi o robô DARWIN-OP. Os limites superiores e inferiores são descritos nas tabelas 5.2 e 5.3.

Tabela 5. 2 - Limites das restrições dos parâmetros (amplitudes).

Parâmetro	$A_{\text{balancing}}$	A_{hPitch}	A_{kPitch}	$A_{\text{rotação}}$	A_{yield}	A_{compasso}
Limite inferior	0.01	0.01	0.01	0.01	0.01	0.01
Limite superior	40	80	110	100	8	80

Tabela 5. 3 - Limites das restrições dos parâmetros (offsets, σ e frequência).

Parâmetro	O_{hRoll}	O_{hPitch}	O_{kPitch}	O_{aRoll}	O_{aPitch}	σ	ω (rad/s)
Limite inferior	-50	-19	0	0.01	-50	$\pi/4$	1.57
Limite superior	40	19	100	100	40	$\pi/4$	6.28

5.8 Algoritmo de Otimização

O algoritmo de otimização utilizado baseia-se no NSGA-II (*Non-dominated Sorting Genetic Algorithm*), um algoritmo de otimização multi-objetivo que pertence ao campo da computação evolucionária. Este algoritmo é descrito na secção 4.2.3 desta dissertação. Baseado numa população e em operadores genéticos, nomeadamente mutação, seleção e *crossover*, o objetivo deste algoritmo consiste em obter uma população de soluções candidatas que formam uma frente Pareto condicionada pelas funções de custo descritas anteriormente. Na nossa proposta de algoritmo, no sentido de recombinar e sujeitar os cromossomas a mutação, foram considerados o *Simulated Binary Crossover* (SBX) e a Mutação Polinomial respetivamente que visam simular o trabalho dos operadores binários tradicionais. Para este trabalho, o operador SBX é particularmente adequado no âmbito da nossa proposta de otimização, uma vez que a propagação de soluções-filho a partir das soluções-pai pode ser controlada através do uso de um índice de distribuição. Com este operador, qualquer região contígua arbitrária pode ser pesquisada, desde que exista diversidade suficiente mantida entre as soluções pai viáveis. O funcionamento deste operador está descrito em [99].

No sentido de calcular a *fitness*, este algoritmo recorre a duas ferramentas: classificação (*ranking*) e *crowding distance*, a qual indica a extensão em que a área de espaço objetivo, no qual a solução reside, é populada por outras soluções (medida descrita em 4.2.3).

Por outro lado, num algoritmo genético é necessário definir a forma como se vão selecionar os indivíduos na população que vão criar descendência para a próxima geração e quanta descendência lhes é permitido gerar. Isto permite dar uma certa ênfase aos indivíduos mais aptos na população na esperança que a sua descendência, por sua vez, possua uma *fitness* ainda melhor. O método de seleção escolhido para este algoritmo foi um método de seleção por torneio. Assim, entre duas soluções escolhidas aleatoriamente, aquela que domina a outra é selecionada. Se nenhuma domina, é selecionada a que se encontra na área menos populada.

Neste contexto, o NSGA-II seleciona as soluções com melhor classificação (*rank*), como primeira prioridade e, em caso de haver *ranks* iguais, recorre-se à *crowding distance* para selecionar a melhor solução.

6 Resultados e Discussão

Neste capítulo são apresentados os resultados dos testes efetuados e posteriormente são analisados. Primeiramente será apresentada a evolução das soluções ao longo das gerações de modo a verificar se todos os indivíduos atingiram o objetivo primordial de conseguir cumprir todos os estágios (*balancing*, *stepping* e *walking*) com sucesso. Posteriormente são analisadas as últimas populações dos vários testes e as suas soluções não dominadas, e comparadas com soluções obtidas a partir de parâmetros definidos à mão. Por fim é feita uma análise aos parâmetros e às funções objetivo de modo a detetar eventuais correlações positivas ou redundâncias.

No sentido de validar a nossa proposta de otimização de locomoção para um robô bípede, foram levadas a cabo uma série de simulações na plataforma de simulação Webots. O controlador de locomoção gera trajetórias para o bípede simulado Darwin-OP. O protocolo experimental englobava, inicialmente, o objetivo de testar o algoritmo proposto no robô bípede real todavia isso não foi possível pois o robô sofreu uma avaria na altura em que os testes se iam iniciar.

O Darwin-OP trata-se de um robô humanoide leve com vinte graus de liberdade (DOFs), dos quais, seis graus de liberdade encontram-se nas pernas. É ainda equipado com oito sensores de força na sola dos pés (quatro em cada um). Em cada ciclo sensorial (8ms), é adquirida informação sensorial. O sistema é integrado considerando o método de Euler usando um passo de integração fixo de 1ms. Em todas as experiências, considerou-se uma população de 100 cromossomas e número máximo de 1000 gerações.

Por outro lado, utilizaram-se operadores genéticos dentro do âmbito da estratégia evolucionária, nomeadamente o crossover SBX e a mutação polinomial, com uma determinada probabilidade inerente de 0.9 e 1/7 respetivamente.

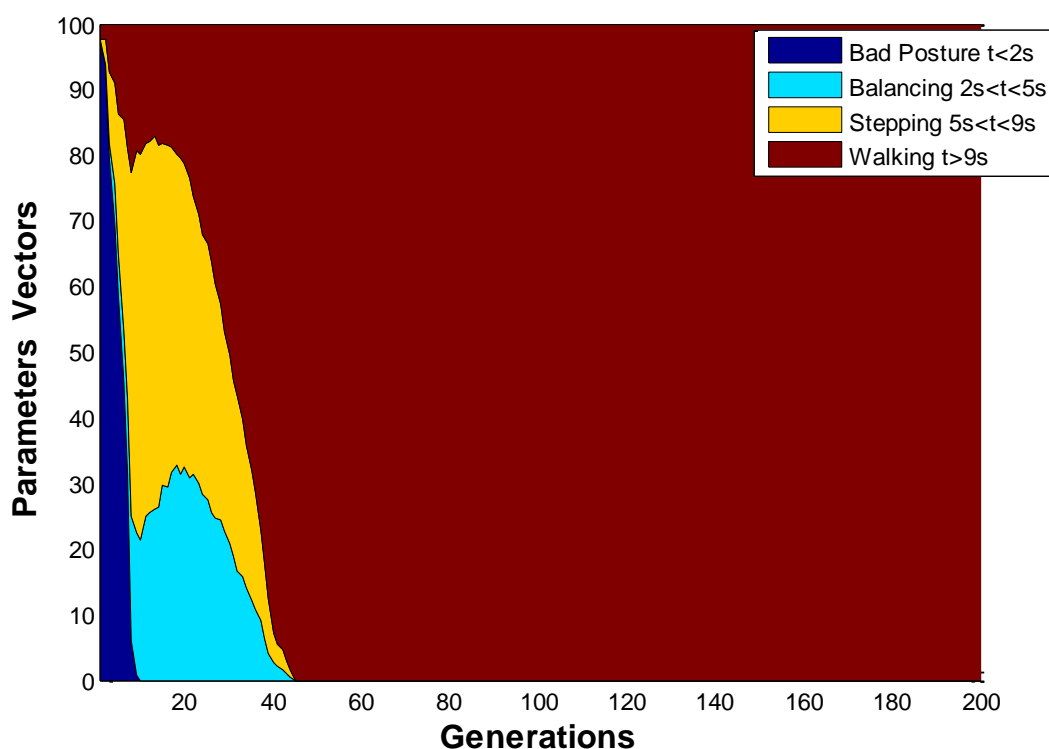
Uma marcha regulada por parâmetros definidos à mão foi incluída, como um indivíduo, numa população inicial aleatória. Cada cromossoma é avaliado durante 20 segundos. No fim da avaliação de cada cromossoma, o robô volta à posição e rotação iniciais de modo a que todos os cromossomas em todas as populações sejam avaliados sob as mesmas condições

As simulações foram executadas num computador pessoal Intel(R) Core™ i3-2330M CPU 2.20GHz (RAM de 4Gb). Todavia neste computador, as simulações demoravam perto de 24 horas. Também se recorreu a outro computador fixo Intel(R) Core(TM) i7- 2600K CPU 3.40GHz (8 GB) o qual conseguia executar as simulações em apenas 12 horas.

Neste contexto, avaliou-se a capacidade da abordagem proposta no sentido de produzir uma marcha com performance melhorada, a partir da marcha inicial programada manualmente, de acordo com critérios específicos.

6.1 Evolução da Otimização

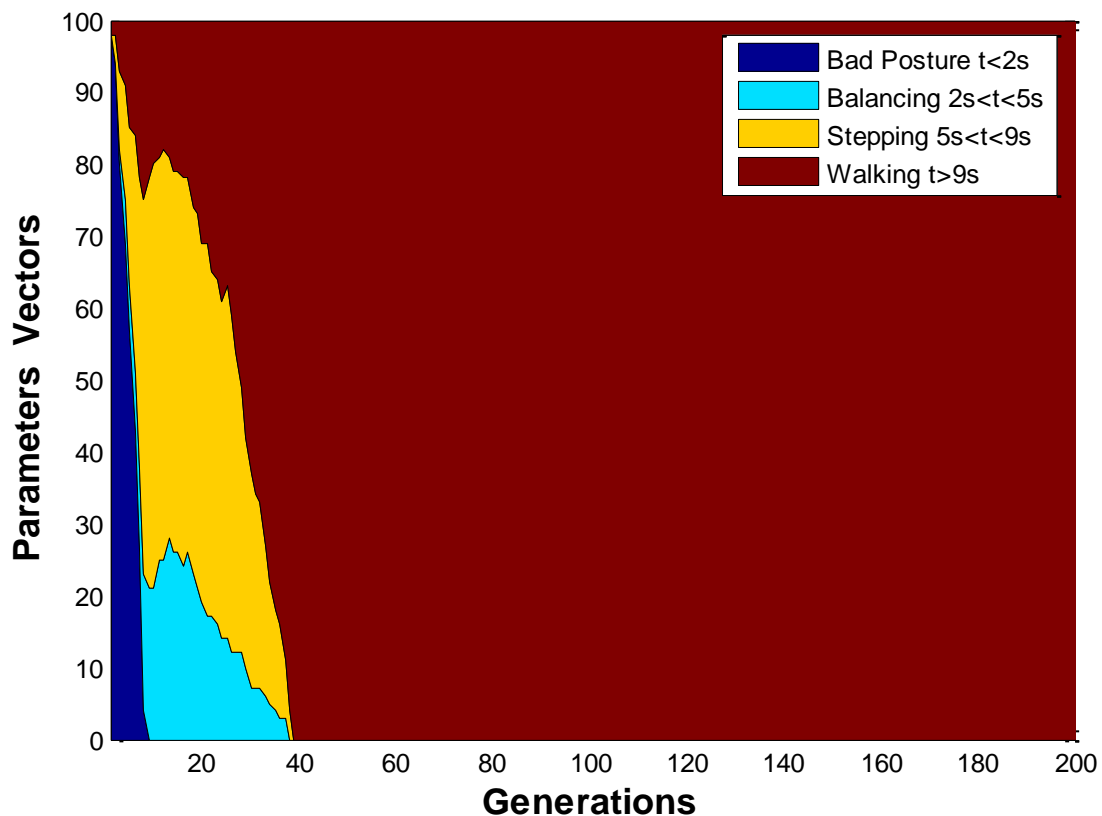
Cada indivíduo ou solução é testado no máximo durante 20 segundos. A partir dos 2 segundos até aos 5 segundos é avaliado o *balancing*. Posteriormente, nos seguintes segundos (até aos 9s) é avaliado o *stepping* e por fim é avaliado o *walking* (entre os 9 e 20 segundos). Estes três movimentos encontram-se ilustrados nas figura 6.5, 6.6 e 6.7. Estas simulações correram ao longo de 1000 gerações todavia os gráficos apenas retratam as 200 primeiras visto que os objetivos são cumpridos logo nas gerações iniciais.



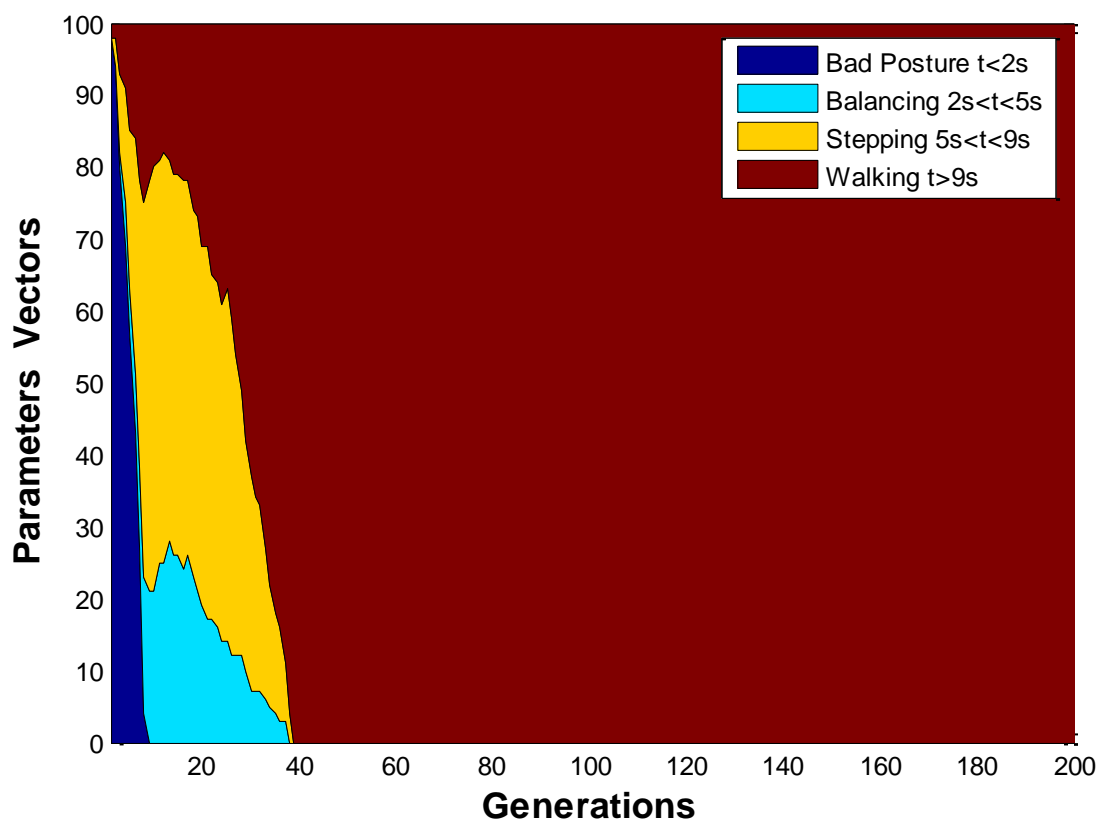
6. 1 - Gráfico representativo da evolução por estágios dos indivíduos ao longo das gerações. Este gráfico representa uma média dos resultados obtidos em várias simulações

Através do gráfico da figura 6.1, o qual representa uma média dos resultados obtidos a partir de várias simulações, é possível constatar que, na primeira geração, apenas 2% dos indivíduos conseguem alcançar o estágio do *walking*, 1% atinge o *stepping* e os restantes indivíduos assumem uma má postura inicial que os leva a cair. Esta percentagem de indivíduos com má postura inicial vai diminuindo bruscamente, e ao fim de 10 gerações nenhum indivíduo assume uma má postura inicial. Entra as 10 e

as 20 gerações, obteve-se em média uma percentagem de indivíduos a executar o *balancing* a rondar entre os 22% e 32%, uma percentagem de indivíduos a executar o *stepping* a rondar entre os 50% e os 55%, e uma percentagem de indivíduos a executar o *walking* a rondar os 18% e os 22%. O número de indivíduos que executam o estágio de *walking* aumenta progressivamente, chegando aos 50% ao fim de 35 gerações. Esta tendência continua e em menos de 45 gerações todos os indivíduos atingem o estágio de *walking*. É de referir ainda que apesar do gráfico 6.1 representar uma média das várias soluções, é muito semelhante ao que se passou em cada uma das simulações, uma vez que se verificou uma variação mínima entre elas. As figuras 6.2 e 6.3 representam o pior e melhor resultado, entre as várias simulações, respetivamente.



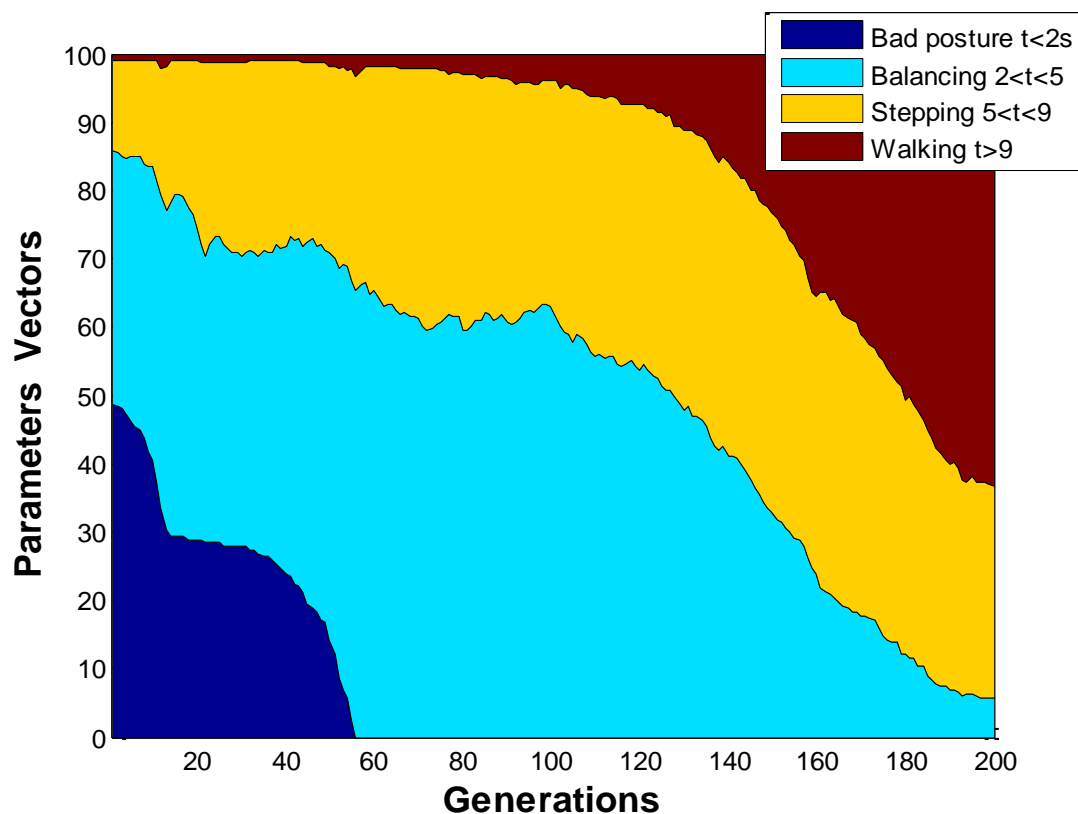
6. 2 - Gráfico representativo da evolução por estágios dos indivíduos ao longo das gerações. Este gráfico representa o exemplo com piores resultados obtidos entre várias simulações



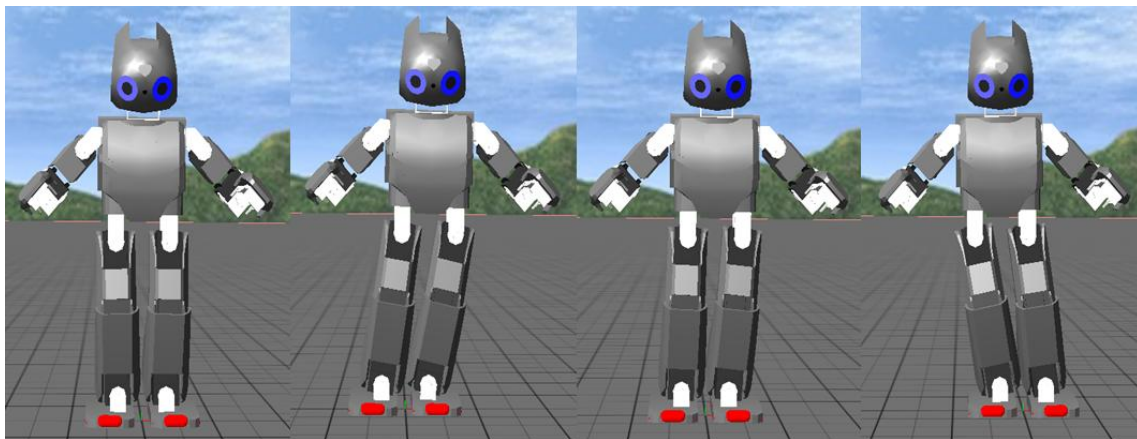
6. 3 - Gráfico representativo da evolução por estágios dos indivíduos ao longo das gerações. Este gráfico representa o exemplo com melhores resultados obtidos entre várias simulações

Por outro lado, é de salientar a melhoria significativa que houve entre a nossa proposta de otimização e o algoritmo inicial aplicado no Darwin que usamos como ponto de partida. Esse algoritmo inicial apresentou os resultados descritos no gráfico da figura 6.4. Embora seja possível constatar que apenas metade dos indivíduos assumiram má postura inicial na primeira geração, verifica-se que ao longo das gerações, existe uma evolução substancialmente menor em relação ao que se passou com os testes finais. De fato, ao fim de 200 gerações ainda se obteve uma percentagem de aproximadamente 35 % de indivíduos que não alcançaram o último estágio (*walking*). Isto revela uma grande melhoria por parte da nossa proposta de otimização. Relativamente à diferença, na primeira geração, de indivíduos a executar uma má postura inicial, deve-se a uma restrição no código que garantia que no mínimo haveria 50% de indivíduos com boa postura. Essa restrição foi posteriormente removida e daí se justificar que, nos testes finais, tenha havido na geração inicial uma percentagem maior de indivíduos com má postura. Todavia, o que realmente interessa

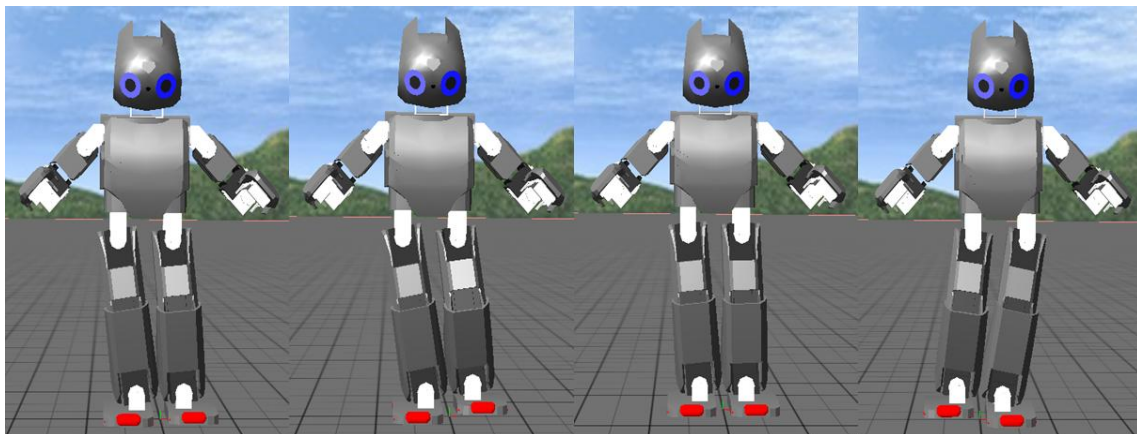
é o fato de haver uma maior percentagem de indivíduos a executar o *walking* e nesse aspeto houve um avanço significativo. Este avanço deve-se não só a uma melhor formulação das funções objetivo, as quais sofreram uma série de *updates* ao longo deste estudo, mas também ao fato de se ter incluído na população inicial uma solução *hand tuned* e outra solução ótima obtida em simulações anteriores, as quais acabaram por ter uma influência na forma como houve uma convergência mais rápida de indivíduos a alcançar a última fase.



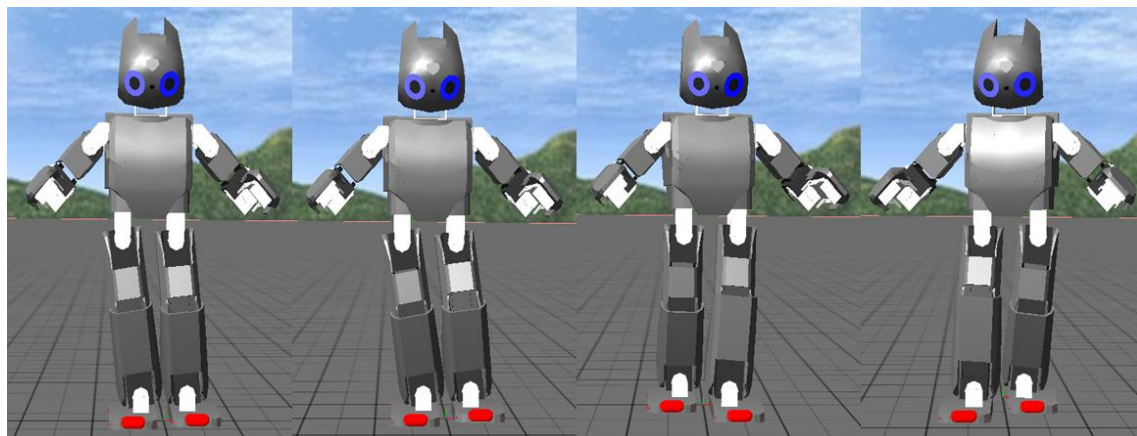
6. 4 - Gráfico representativo da evolução por estágios dos indivíduos ao longo das gerações. Este gráfico representa os resultados obtidos para o Darwin antes de ser aplicada a nossa proposta de otimização



6. 5 - Fotografias tiradas ao Darwin durante a fase de *balancing* numa simulação efetuada no *webots*.



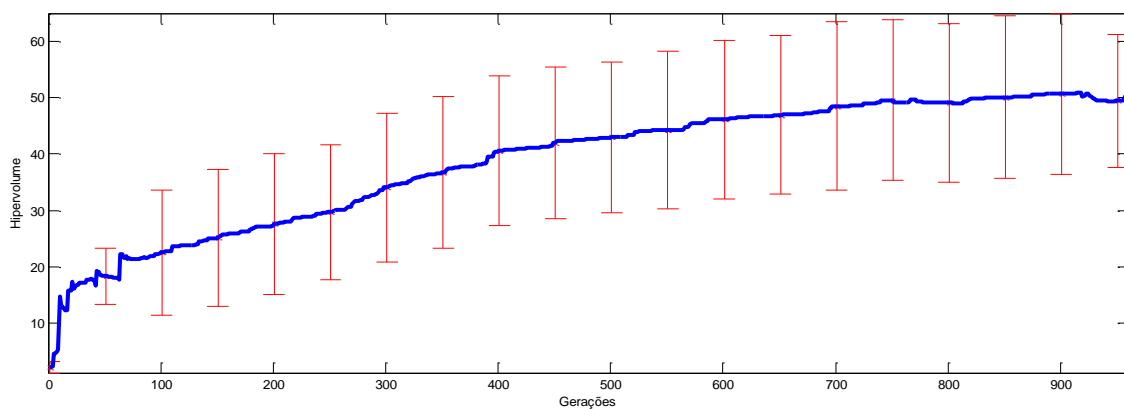
6. 6 - Fotografias tiradas ao Darwin durante a fase de *stepping* numa simulação efetuada no *webots*.



6. 7 - Fotografias tiradas ao Darwin durante a fase de *walking* numa simulação efetuada no *webots*.

6.2 Hipervolume

O hipervolume é um indicador da qualidade da frente Pareto durante o processo de otimização. Neste contexto, foi calculado o hipervolume médio assim como o desvio padrão de 50 em 50 gerações durante as 1000 gerações usando como base dez simulações de teste. No cálculo do hipervolume foi considerada uma aproximação ao ponto de Nadir. Este representa os piores valores de cada função objetivo no conjunto ótimo Pareto. No entanto, dada a dificuldade computacional derivada do cálculo deste ponto, foi utilizado como substituto o ponto de referência $r = \{0, 2000, 0\}$, o qual foi selecionado baseado nas soluções não dominadas presentes nas populações finais. O gráfico da figura 6.8 representa uma média da evolução da medida de hipervolume ao longo das 1000 gerações para um universo de 10 simulações teste. Neste contexto, é possível verificar que há um aumento progressivo do hipervolume até aproximadamente 700 gerações e posteriormente verifica-se uma estagnação. Deste modo, este aumento sugere uma melhor aproximação à frente Pareto ótima. Além disso, as soluções obtidas encontram-se melhor distribuídas no espaço objetivo assim como são mais diversificadas e amplas em termos de objetivos. Por outro lado, é possível deduzir que não era necessário um número tão elevado de gerações visto que o hipervolume não apresenta melhorias a partir das 700 gerações, e deste modo, é possível deduzir que, nesse ponto, já foi determinada a frente Pareto ótima e que esta não vai conseguir melhorias.



6. 8 - Média (linha azul) e desvio padrão (barras verticais) da medida de hipervolume em 10 simulações teste

6.3 Soluções Extremas da Frente Pareto

No sentido de analisar os melhores indivíduos obtidos fez-se uma análise aos extremos da frente Pareto. Deste modo, extraíram-se as soluções não dominadas de uma das simulações e determinou-se os máximos e mínimos obtidos para as várias funções objetivo. Deste modo, na tabela 6.1, são apresentados os valores máximos e os valores mínimos das soluções extremas obtidas para as várias simulações. É também apresentada a solução *hand-tuned* colocada na população inicial mais uma solução S não dominada que representa uma solução com resultados de boa qualidade a nível de todas as funções objetivo.

Tabela 6. 1 - Valores de funções objetivo e respetivos parâmetros que correspondem a soluções extremas da frente Pareto, a uma solução hand tuned e a uma solução S não dominada.

	A	B	C	D	E	F	Hand tuned	S
Deslocamento	1,025	0,915	0,333	0,008	0,013	0,014	0,505	0,8161
Força	694,09	814,42	587,69	251,37	232,32	259,19	693,01	607,01
Ground Clearance	0,018	0,021	0,046	0,013	0,012	0,009	0,022	0,021
Abalancing	10,32	9,25	10,39	9,81	9,62	9,91	10	10,2
Ahpitch	2,36	1,23	32,83	9,69	9,71	9,81	15	8,27
AKpitch	21,47	6,34	0,02	14,84	16,35	15,71	30	15,15
σ	0,53	0,53	0,52	0,52	0,52	0,53	0,52	0,53
Arotação	4,93	0,78	2,35	7,40	5,79	6,05	5	2,86
Ayield	7,65	7,42	7,92	1,58	1,62	6,98	2	7,84
Acompasso	16,66	16,07	17,05	2,74	2,40	2,80	14	19,05
ω	3,02	3,21	2,13	1,91	1,88	1,89	4,2	2,22
Ohroll	3,25	3,04	1,04	-1,58	-1,69	-0,73	-1,5	1,70
Ohpitch	-9,97	-15,22	-10,22	-14,36	-8,13	-8,05	-16	-11,16
Okpitch	0,58	0,05	0,02	0,004	0,15	0,21	30	0,08
Oaroll	16,30	16,38	18,86	17,86	17,68	17,73	10	19,32
Oapitch	-14,66	-14,11	8,54	-16,86	-16,96	-16,87	1,5	6,45

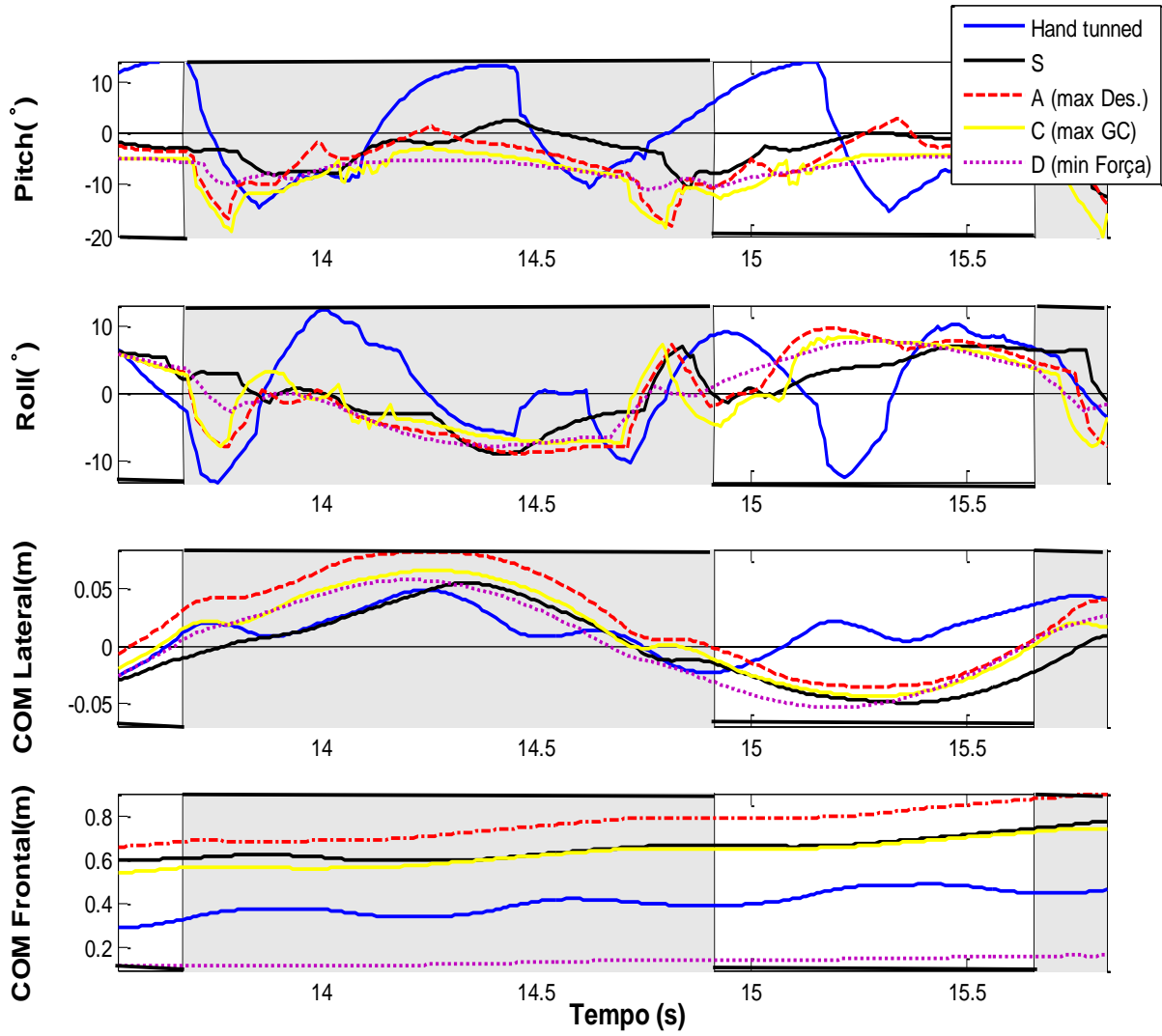
Deste modo, os indivíduos A, C e E correspondem às soluções com maior *Deslocamento*, *Força* e *Ground clearance* respetivamente, uma vez que era pretendido maximizar o deslocamento frontal do robô e altura que os seus pés levantam e minimizar a função objetivo *Força*. Por outro lado, os indivíduos B, D e F representam as soluções com piores valores de *Deslocamento*, *Força* e *Ground Clearance* respetivamente. Através da tabela 6.1 é possível também verificar quais os valores dos parâmetros que definiram cada solução.

Estes indivíduos foram posteriormente utilizados no sentido de avaliar a evolução das trajetórias assumidas pelo centro de massa do robô. Neste contexto, realizou-se uma análise das variações a nível do *pitch*, *roll* e posição do COM do Darwin ao longo de um passo. A figura 6.9 representa a evolução do *pitch* (1º gráfico), do *roll* (2º gráfico) e do COM lateral e frontal (3º e 4º gráficos) para a solução *hand-tuned* (linha azul), e soluções não dominada S (linha preta) e soluções extremas A (tracejado vermelho), C (linha amarela) e E (ponteadado roxo).

Deste modo, é possível constatar uma melhoria significativa em relação à solução *hand tuned* no que diz respeito à evolução do *pitch* e do *roll* do COM. De fato, é visível que a solução *hand tuned* apresenta maiores oscilações traduzindo-se numa locomoção menos eficaz. A solução S apresenta-se como a solução de maior qualidade em geral, com menos oscilações e menor amplitude.

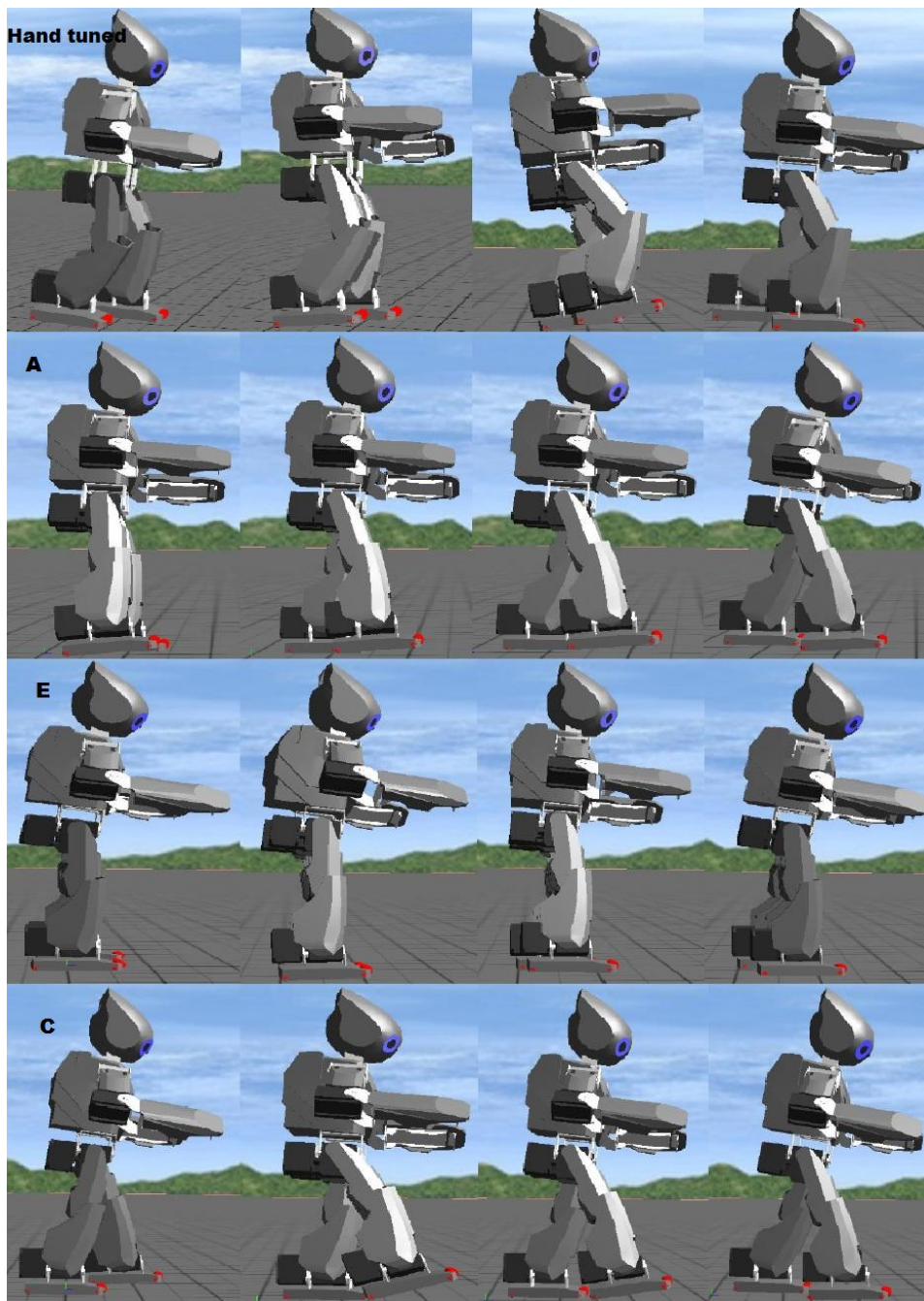
Relativamente á evolução do deslocamento lateral do COM, é perceptível que a solução *hand tuned* apresenta oscilações indesejadas que afetam a qualidade da locomoção. Por outro lado, as soluções não dominadas apresentam um comportamento menos oscilatório e mais constante traduzindo-se num *balancing* mais eficaz.

Por fim, a evolução do deslocamento frontal do COM revelou resultados semelhantes. As soluções não dominadas apresentam um comportamento mais suave que a solução *hand tuned*. Por outro lado, as soluções não dominadas apresentam maior amplitude de deslocamento frontal, traduzindo-se num maior deslocamento. A única exceção é a solução E, correspondente ao valor mínimo de *Força*, que apresentou menor amplitude de deslocamento frontal que a solução *hand tuned*. Isto corresponde à realidade pois a solução E resultou numa marcha lenta e, conseqüentemente com baixo deslocamento.



6. 9 - Comparação da evolução do Pitch, do Roll e do COM (frontal e lateral) entre a solução *hand tuned* e as soluções não dominadas A, C, E e S. Os retângulos cinzentos representam o período em que o pé esquerdo está no chão e os retângulos brancos representam o período em que o pé esquerdo não está no chão

A marcha do Darwin baseado em parâmetros *hand-tuned* assim como uma marcha baseada numa solução não dominada da frente Pareto ilustradas na figura 6.10 pode ser visualizado em [100].



6. 10 - Fotografias da simulação de marcha do Darwin no Webots. De cima para baixo: solução *hand tuned*, solução A (máximo deslocamento), solução E (mínima força) e solução C (máximo *ground clearance*).

6.4 Análise de Sensibilidade

Após a obtenção de resultados, um dos principais objetivos consiste na análise das diversas variáveis testadas, isto é, determinar a forma como elas se correlacionam. Deste modo, podemos chegar a conclusões sobre a influência de determinados parâmetros nas funções objetivo e, logicamente, na locomoção bípede. Por outro lado, isto também é aplicável entre as próprias funções objetivo, ou seja, é importante analisar a relação entre as várias funções objetivo de modo a detetar eventuais objetivos redundantes.

6.4.1 Correlação entre Parâmetros e Funções Objetivo

Deste modo, as tabelas 6.2, 6.3 e 6.4 apresentam os valores de correlação entre parâmetros e funções objetivo. A correlação é feita através do cálculo do coeficiente de correlação de Pearson (R) e dos valores p correspondentes que traduzem a significância estatística da associação. Nas tabelas encontram-se destacados os valores mais significativos (coeficiente de correlação $R > 0.6$ e significância estatística $p < 0.05$) para a análise em questão.

Tabela 6. 2 - Coeficientes de correlação de Pearson e valores p de significância entre os parâmetros e a função objetivo *Deslocamento*. Tanto para R como para P são apresentados os valores máximos, mínimos, média e desvio padrão calculados para 10 simulações. Os valores a ter em consideração encontram-se a negrito e sublinhados.

Parâmetros	R				P			
	Mínimo	Máximo	Média	Desvio P.	Mínimo	Máximo	Média	Desvio P.
Abalancing	-0,1415	0,5125	0,2328	0,2137	0,0000	0,3596	0,0752	0,1412
Ahpitch	-0,4422	-0,1144	-0,2766	0,1338	0,0000	0,4096	0,1156	0,1338
AKpitch	-0,2050	0,5114	0,3062	0,2731	0,0000	0,1820	<u>0,0396</u>	0,2731
σ	0,0337	0,4585	0,1621	0,1576	0,0000	0,8280	0,3501	0,1576
Arotação	-0,0692	0,4975	0,1790	0,2372	0,0000	0,6552	0,3205	0,2372
Ayield	-0,1117	0,5582	0,2683	0,2207	0,0000	0,2685	0,0554	0,2207
Acompasso	0,6245	0,8256	<u>0,7008</u>	0,0692	0,0000	0,0000	<u>0,0000</u>	0,0692
ω	0,4795	0,8777	<u>0,7536</u>	0,1465	0,0000	0,0010	<u>0,0002</u>	0,1465
Ohroll	0,4158	0,7848	<u>0,6043</u>	0,1521	0,0000	0,0097	<u>0,0016</u>	0,1521
Ohpitch	-0,1059	0,3205	0,1404	0,1883	0,0017	0,5871	0,2275	0,1883
Okpitch	-0,2341	0,2965	0,1109	0,1875	0,0059	0,5187	0,1814	0,1875
Oaroll	-0,0918	0,4836	0,1246	0,2182	0,0000	0,8866	0,3735	0,2182
Oapitch	-0,5747	-0,2842	-0,4026	0,1190	0,0000	0,0615	0,0116	0,1190

Tabela 6. 3 - Coeficientes de correlação de Pearson e valores p de significância entre os parâmetros e a função objetivo *Força*. Tanto para R como para P são apresentados os valores máximos, mínimos, média e desvio padrão calculados para 10 simulações. Os valores a ter em consideração encontram-se a negrito e sublinhados.

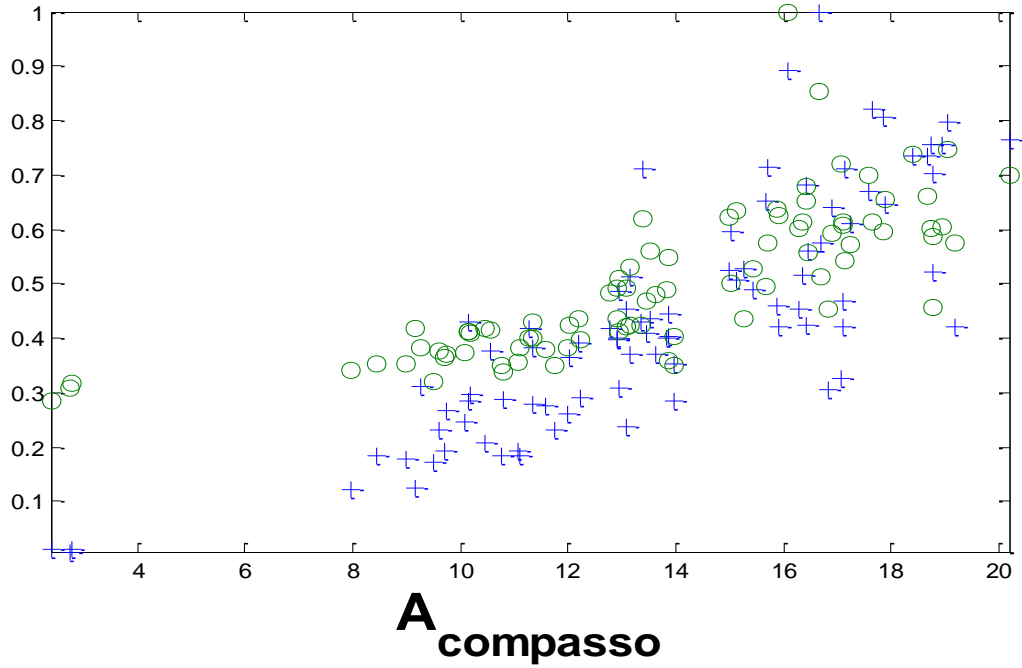
Parâmetros	R				P			
	Mínimo	Máximo	Média	Desvio P.	Mínimo	Máximo	Média	Desvio P.
Abalancing	-0,0953	0,3467	0,0524	0,1617	0,0007	0,9779	0,5200	0,1617
Ahpitch	-0,2666	0,2360	-0,0252	0,1627	0,0227	0,9258	0,4991	0,1627
AKpitch	0,0364	0,4919	0,2836	0,2028	0,0000	0,8147	0,2589	0,2028
σ	-0,0726	0,3978	0,1356	0,1507	0,0001	0,4824	0,2799	0,1507
Arotação	0,0887	0,5515	0,3396	0,2020	0,0000	0,3981	0,1152	0,2020
Ayield	0,0413	0,5526	0,3066	0,1718	0,0000	0,6831	0,1238	0,1718
Acompasso	0,3827	0,7679	<u>0,6119</u>	0,1549	0,0000	0,3224	<u>0,0488</u>	0,2049
ω	0,6526	0,8221	<u>0,7325</u>	0,0654	0,0000	0,0000	<u>0,0000</u>	0,0654
Ohroll	0,7558	0,9075	<u>0,8478</u>	0,0594	0,0000	0,0000	<u>0,0000</u>	0,0594
Ohpitch	-0,0957	0,2260	0,0436	0,1503	0,0375	0,5616	0,3155	0,1503
Okpitch	-0,4576	0,3807	0,0428	0,3040	0,0000	0,5861	0,1829	0,3040
Oaroll	-0,0996	0,3277	0,1076	0,1804	0,0009	0,9497	0,3627	0,1804
Oapitch	-0,2519	0,3052	-0,0845	0,2029	0,0133	0,5044	0,1765	0,2029

Tabela 6. 4 - Coeficientes de correlação de Pearson e valores p de significância entre os parâmetros e a função objetivo *Ground Clearance*. Tanto para R como para P são apresentados os valores máximos, mínimos, média e desvio padrão calculados para 10 simulações. Os valores a ter em consideração encontram-se a negrito e sublinhados

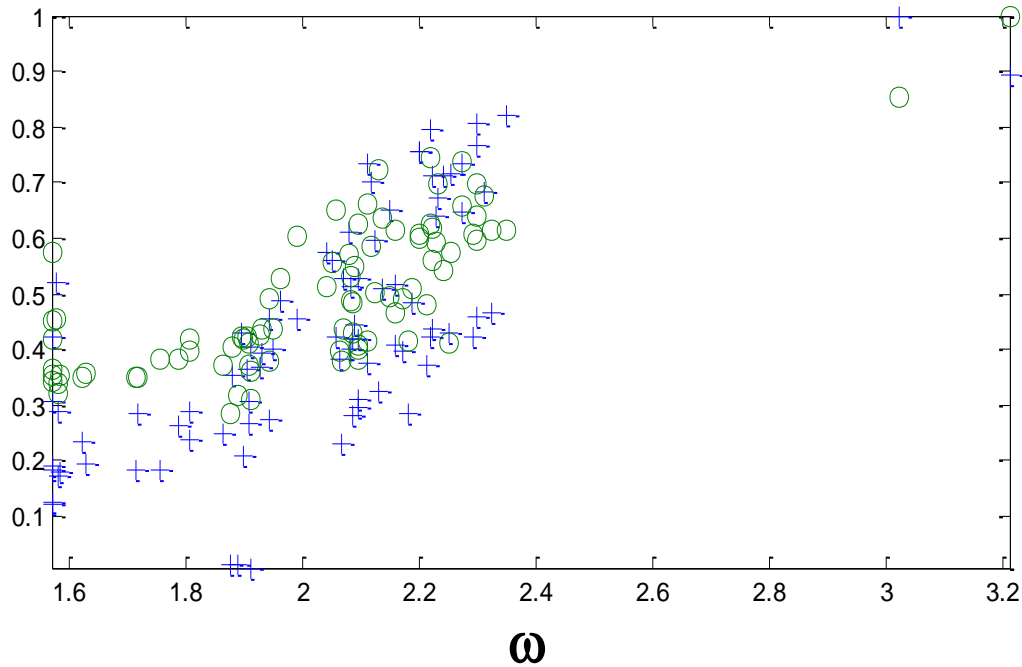
Parâmetros	R				P			
	Mínimo	Máximo	Média	Desvio P.	Mínimo	Máximo	Média	Desvio P.
Abalancing	-0,5009	0,2269	-0,1730	0,2494	0,0000	0,3985	0,1531	0,2494
Ahpitch	0,3407	0,8436	<u>0,6187</u>	0,1957	0,0000	0,0011	<u>0,0002</u>	0,1957
AKpitch	-0,4185	-0,0040	-0,2076	0,1374	0,0001	0,9713	0,2206	0,1374
σ	-0,3148	0,2792	0,0420	0,2172	0,0018	0,4041	0,1922	0,2172
Arotação	-0,1720	0,2047	0,1145	0,1433	0,0602	0,2266	0,1348	0,1433
Ayield	0,0061	0,2465	0,1229	0,1171	0,0173	0,9687	0,4526	0,1171
Acompasso	-0,1412	0,2621	0,1143	0,1467	0,0366	0,6893	0,2210	0,1467
ω	-0,4068	0,0999	-0,1011	0,1804	0,0061	0,6551	0,3280	0,1804
Ohroll	-0,1632	0,5515	0,2538	0,3002	0,0000	0,9361	0,2098	0,3002
Ohpitch	-0,4171	0,0012	-0,2647	0,1489	0,0001	0,9906	0,1734	0,1489
Okpitch	-0,2939	0,0714	-0,0765	0,1530	0,0037	0,8809	0,4179	0,1530
Oaroll	-0,1158	0,4485	0,1544	0,2140	0,0023	0,6335	0,1995	0,2140
Oapitch	0,2534	0,8345	0,4604	0,2328	0,0000	0,0179	<u>0,0049</u>	0,2328

Após analisarmos as tabelas 6.2, 6.3 e 6.4 com atenção, é possível constatar que existem alguns parâmetros que apresentam uma forte correlação com as funções objetivo. Neste contexto, o *deslocamento* apresenta uma forte correlação com os parâmetros A_{compasso} , ω e O_{hRoll} . O parâmetro A_{compasso} apresenta um coeficiente de correlação médio de aproximadamente 0.7 e uma significância estatística média igual a 0, representando um forte indicador de correlação. Este resultado era relativamente previsível uma vez que este parâmetro controla o tamanho dos passos durante a marcha. Relativamente ao ω e ao O_{hRoll} , estes apresentaram coeficientes de correlação médios de 0.75 e 0.60 respetivamente com umas significâncias estatísticas médias praticamente nulas. Deste modo, deduz-se também que a frequência do oscilador (ω) e o *offset* (O_{hRoll}), o qual controla o ângulo entre as pernas, influenciam a performance da locomoção a nível de *deslocamento*. Estes parâmetros evidenciaram também ser os fatores que mais se correlacionam com a função objetivo *force*, apresentando um coeficiente de correlação médio de aproximadamente 0.61 (A_{compasso}), 0.73 (ω) e 0.85 (O_{hRoll}), e umas significâncias estatísticas igual a zero (exceto para o A_{compasso} que foi aproximadamente 0.05). Por fim, a função objetivo *Ground Clearance* apresenta uma forte correlação com o parâmetro A_{hPitch} , o qual controla o movimento da anca durante a fase do *stepping*, com um coeficiente médio de aproximadamente 0.62 e p quase nulo. Os gráficos das figuras 6.11, 6.12 e 6.13 demonstram a influência de alguns dos parâmetros evidenciados nas funções objetivo.

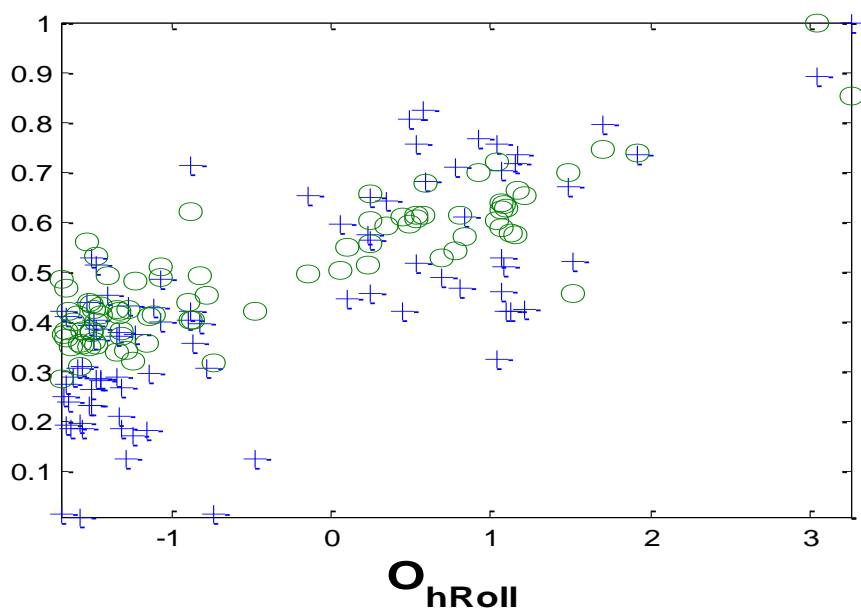
É de referir que as correlações tidas em conta apresentam valores positivos, e que quanto maior os valores verificados maior é a influência que estes têm na função objetivo em questão. Por outro lado, esses mesmos parâmetros apresentaram um valor de p quase nulo, revelando possuir uma forte significância estatística relativa à associação que é feita entre o parâmetro e a função objetivo. Estes valores de p perto de zero garantem fiabilidade, ou seja, no caso de se voltarem a repetir os testes, os resultados obtidos serão os mesmos.



6. 11 - Gráfico representativo da relação do parâmetro A_{compasso} com o aumento do ratio (deslocamento)/máximo(deslocamento) a cor azul e com o aumento do ratio (força)/mínimo(força) a cor verde



6. 12 - Gráfico representativo da relação do parâmetro ω com o aumento do ratio (deslocamento)/máximo(deslocamento) a cor azul e com o aumento do ratio (força)/mínimo(força) a cor verde



6. 13 - Gráfico representativo da relação do parâmetro O_{hRoll} com o aumento do ratio (deslocamento)/máximo(deslocamento) a cor azul e com o aumento do ratio (força)/mínimo(força) a cor verde

6.4.2 Correlação entre Funções Objetivo

Relativamente às funções objetivo, foi também realizado o mesmo estudo de sensibilidade. A tabela 6.5 apresenta os valores de correlação de *Pearson* e respetivos valores de p (nível de significância estatística).

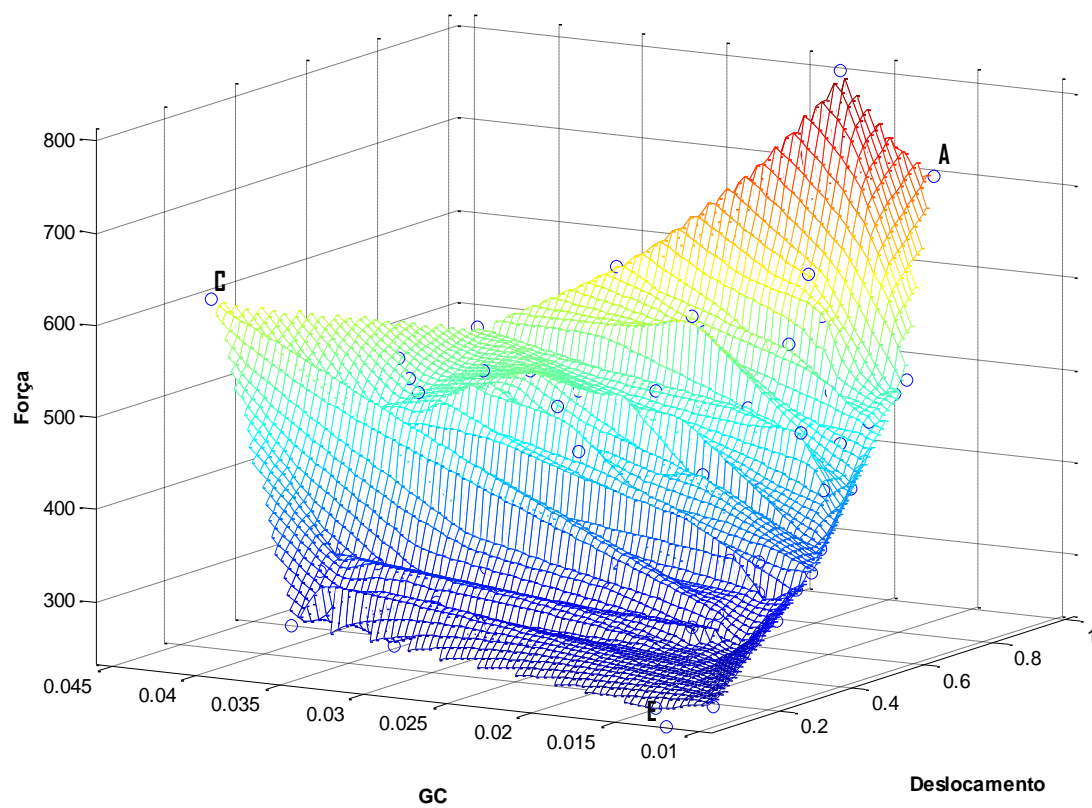
Tabela 6. 5 - Coeficientes de correlação de *Pearson* e valores p de significância entre as função objetivo.

	Força		Ground Clearance	
	R	P	R	P
Deslocamento	0.7054	0.0000	-0.0406	0.6994
Força	-	-	0.6092	0.0000

Após analisar estes valores, é possível deduzir uma conclusão óbvia – a função objetivo *força* e encontra-se estreitamente ligada às outras duas funções (*deslocamento* e *ground clearance*). Os valores de correlação R foram superiores a 0.6 (0.7054 em relação ao *deslocamento* e 0.6092 em relação ao *ground clearance*) e os valores de p foram nulos. Assim, quanto maior o deslocamento vertical (*ground clearance*) maiores são os valores obtidos para a força. Isto é relativamente previsível uma vez que quanto mais os pés levantam maior é a força de impacto ao tocar no chão. Além disso, a função objetivo *force* é uma função cumulativa, isto é, corresponde a um somatório de forças de impacto ao longo da marcha. Logo um maior deslocamento do bípede (*deslocamento*), pode corresponder a um maior número de passos e, conseqüentemente, mais pontos de contato com o chão, contribuindo com forças de impacto para o somatório da força, ou então, pode-se dever a passos com maior amplitude que por sua vez se traduzem em maior forças de impacto com o chão.

Por outro lado, esta análise de sensibilidade patenteia a ausência de relação entre o *deslocamento* e o *ground clearance*. De fato, os valores de correlação de Pearson e de significância estatística ($R=-0.0406$ e $p=0.6994$) demonstram que as duas funções objetivo são completamente independentes uma da outra.

O gráfico da figura 6.14 permite constatar com alguma clareza a relação entre as várias funções objetivo, uma vez que se torna visível que os valores de *força* aumentam com o *deslocamento* e com o *ground clearance*, e que estes dois últimos não apresentam qualquer tipo de relação mostrando-se independentes um do outro. Este gráfico permite ainda constatar que as soluções não dominadas formam uma frente Pareto ampla com uma grande diversidade de soluções não dominadas.



6. 14 - Gráfico tridimensional com valores das três funções objetivo obtidos a partir das soluções não dominadas da última população de uma simulação teste. Os círculos representam as soluções não dominadas, entre as quais encontram assinaladas as soluções A, C e E correspondentes aos extremos da frente Pareto (máximo deslocamento e ground clearance e mínima força).

6.5 Principal Component Analysis

No sentido de avaliar a relação entre as várias funções objetivo foi também realizado um estudo de PCA (*principal component analysis*). Neste contexto, a tabela 10 demonstra que a componente 1 juntamente com a componente 2 explicam 98.4 % da variação possibilitando a elaboração de um *biplot* (figura 6.15).

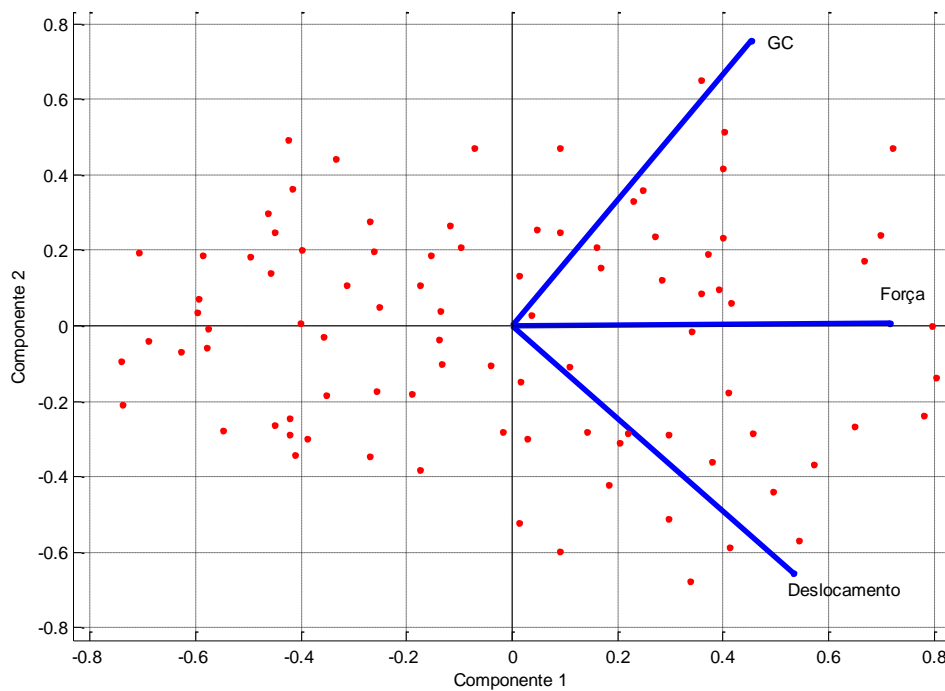
Tabela 6. 6 - Importância das várias componentes no estudo de PCA.

	Componente 1	Componente 2	Componente 3
Deslocamento	0.5325	-0.6534	-0.5381
Força	0.7147	0.0064	0.6994
Ground Clearance	0.4536	0.7570	-0.4703
Eigenvalues	1.9122	1.0401	0.0477
Cumulative proportion	0.6374	0.9841	1.0000

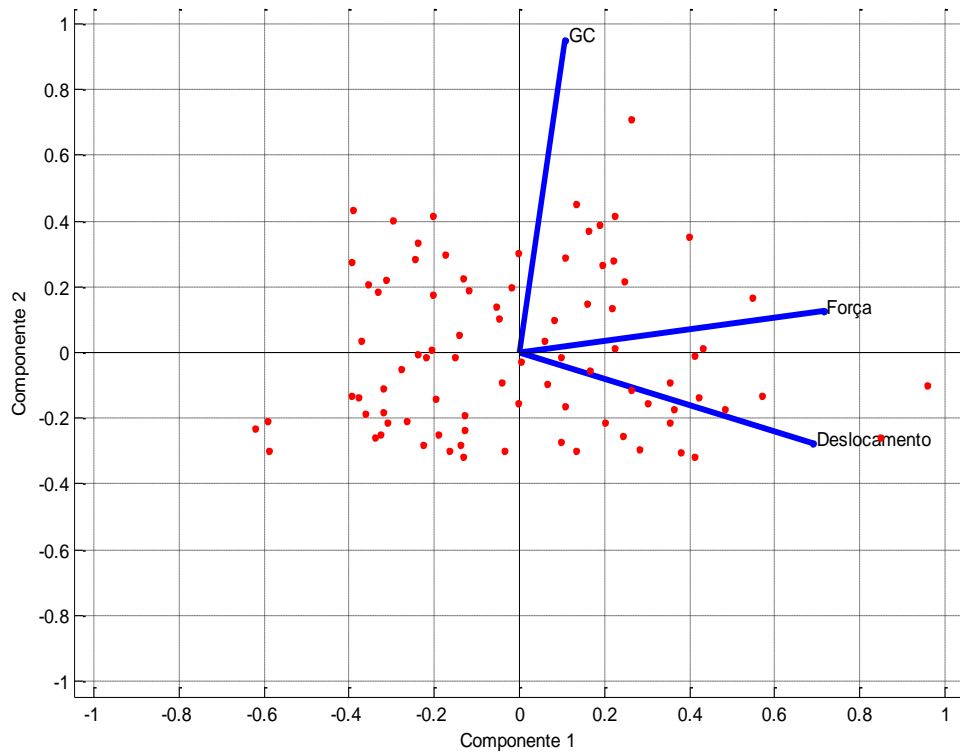
Deste modo, o gráfico da figura 6.15 valida as conclusões deduzidas anteriormente uma vez que traduz as correlações entre as funções objetivo. A correlação entre as três variáveis é representada, no gráfico, pelo cosseno do ângulo entre os vetores, ou seja, a correlação de duas variáveis varia entre -1 e 1 para ângulos entre 180 e 0 graus respetivamente. Neste contexto, existe uma correlação positiva entre a função objetivo *deslocamento* e a função objetivo *força*, assim como entre esta e a função objetivo *ground clearance*, uma vez que se verifica um ângulo inferior a 90°. É de referir que, embora neste estudo de PCA o *deslocamento* e o *ground clearance* aparentam ter uma correlação com a *force* muito semelhante, em outros testes verificou-se uma ligeira diferença constatando-se que a *force* possui uma relação positiva com o *deslocamento* mais forte do que com o *ground clearance*, traduzindo-se num menor ângulo entre os vetores (figura 6.15).

Por outro lado, o ângulo de 90° observado entre a função do *deslocamento* e a função relativa ao *ground clearance*, traduz a independência entre elas, isto é, não existe nenhuma relação aparente entre elas.

Este estudo revela-se de extrema importância na medida em que permite verificar se alguma das funções objetivo assume um papel redundante nesta proposta de otimização. Neste contexto, os resultados apresentam claros sinais de dependência expressos por coeficientes de correlação que permitem deduzir que nenhuma das funções objetivo é redundante assumindo um papel preponderante na otimização da locomoção bípede.



6. 15 - Gráfico de PCA para as três funções objetivo. O *deslocamento* e o GC apresentam correlação semelhante com a *força*.



6. 16 - Gráfico de PCA para as três funções objetivo. A *força* apresenta correlação superior com o *deslocamento* do que com o GC.

7 Conclusões e futuros trabalhos

Esta dissertação apresenta a elaboração de uma nova estratégia de otimização de locomoção bípede cujos resultados revelaram o sucesso da abordagem utilizada e constituem uma fonte de informação útil para quem trabalha no sentido de desenvolver locomoção bípede cada vez mais eficiente e à imagem da marcha humana.

A metodologia de otimização pode ser dissecada em várias etapas principais, nomeadamente a geração de movimento, a especificação dos objetivos e respetivo enquadramento numa evolução por fases, e aplicação de um algoritmo de otimização. Deste modo, foi proposto um sistema de otimização que passa pela combinação entre um controlador de locomoção baseado em geradores centrais de locomoção e um algoritmo de otimização evolucionário.

Um dos pontos fulcrais desta investigação passou pela determinação das funções objetivo, e a importância desta etapa não se prende apenas com a escolha dos objetivos que devem ser otimizados mas sobretudo com a atualização das funções à medida que são feitos testes. As funções foram selecionadas na medida em que fazem parte das funções mais recorrentes na literatura e que à partida nos era mais acessível implementar. Todavia, a obtenção de resultados insatisfatórios numa fase inicial de laboração levou-nos a fazer uma série de reformulações das funções objetivo de modo a conseguir finalmente uma locomoção bípede semelhante à marcha humana. Deste modo, os objetivos propostos foram: posicionar o centro de pressão do bípede no centro do polígono de suporte através da maximização da distância média em relação aos limites do polígono (função Centro de Pressão); maximizar o deslocamento frontal do robô num determinado intervalo de tempo e penalizar eventuais deslocamentos laterais de modo a conseguir uma marcha reta (função Deslocamento); minimizar a força de impacto entre os pés e o chão durante a marcha e manter os pés paralelos ao chão assim como distribuir essa força de impacto uniformemente pelos pés (função Força); e maximizar a altura a que o pé levanta durante o ciclo de marcha (função Ground Clearance).

A locomoção proposta foi estruturada de forma a evoluir por fases, obrigando o robô a iniciar primeiramente um movimento de deslocamento lateral da pélvis

(*balancing*), e posteriormente um movimento de levantar os pés (*stepping*) juntamente com o movimento anterior. Por fim, uma terceira e última fase em que o robô começa a dar passos em frente (*walking*). Os resultados obtidos permitiram-nos concluir que o algoritmo elaborado por nós conseguiu uma evolução por fases com uma taxa de sucesso de 100% ao fim de poucas gerações (aproximadamente 40 gerações). Esta rápida convergência de indivíduos a conseguir executar as três fases revela uma melhoria significativa em relação aos resultados que foram obtidos numa fase primária deste trabalho (ao fim de 200 gerações ainda havia aproximadamente 40% de indivíduos que não conseguiam executar o *walking*). Isto demonstra o sucesso das nossas alterações no algoritmo de otimização ao longo deste trabalho e conseqüentemente, demonstra a qualidade de toda a estratégia de otimização proposta nesta dissertação.

Neste contexto, o algoritmo de otimização foi capaz de criar diferentes soluções correspondentes a diferentes locomoções cujas diferenças se devem a diferentes compromissos (*trade-offs*) entre os vários objetivos. De fato, após uma análise às soluções não dominadas, concluiu-se que o algoritmo utilizado consegue criar uma frente Pareto ótima bem definida e bastante ampla com uma boa distribuição das soluções no espaço abrangido pelos objetivos. Esta conclusão foi posteriormente reforçada com o cálculo da medida de hipervolume que nos permitiu constatar que ao longo das gerações houve uma aproximação à frente Pareto ótima. Concluiu-se também que esta é alcançada ao fim de aproximadamente 700 gerações verificando-se uma estagnação desta para gerações adicionais.

Além disso, uma análise feita às soluções extremas da frente Pareto a nível de evolução do *Pitch* e *Roll* assim como da posição centro de massa (COM) do bípede permitiu concluir que houve melhorias significativas em relação a uma solução *hand tuned* que foi retirada de simulações feitas em trabalhos anteriores à conclusão deste novo sistema de otimização. Verificou-se assim que as novas soluções suprimiram parte das oscilações que se verifica para a solução *hand tuned*, resultando numa locomoção com movimentos mais suaves e com menos oscilações e de menor amplitude. Por outro lado, as novas soluções permitem um melhor posicionamento do COM resultando num movimento de *balancing* mais eficaz e conseguindo um maior deslocamento por parte do robô.

É de referir ainda a execução de uma análise de sensibilidade que se revestiu de grande importância na medida que permitiu tirar ilações relevantes sobre as funções objetivos e os parâmetros de CPGs. Deste modo, a partir desta análise foi possível constatar a existência de correlações positivas entre alguns parâmetros e funções objetivo: o Deslocamento e a Força encontram-se intrinsecamente ligado aos parâmetros A_{compasso} , frequência ω e O_{Roll} e o *Ground Clearance* apresenta uma forte correlação com o parâmetro A_{hPitch} . Esta análise permitiu assim descobrir quais os parâmetros mais importantes para a obtenção de uma locomoção bípede eficiente. Por outro lado, permitiu concluir que as funções objetivo Deslocamento e *Ground Clearance*, embora não apresentam uma correlação entre si, apresentam uma forte correlação com a função Força.

No sentido de validar esta ultima ilação foi realizado um estudo de PCA que permitiu concluir que nenhuma das funções usadas é redundante. O PCA acabou por reforçar a conclusão de que existe uma forte correlação positiva entre o Deslocamento e a Força e entre esta e o *Ground Clearance*, e que o *Ground Clearance* e o Deslocamento são independentes um do outro.

No futuro seria pertinente passar da plataforma de simulação para testes reais verificando se este algoritmo consegue gerar uma locomoção estável e eficaz em diversos robôs bípedes reais. No caso de se constatar eventuais problemas na locomoção seria relevante fazer um estudo de modo a corrigir imperfeições da nossa proposta de otimização e adaptá-la a ambientes físicos reais.

Por outro lado sugere-se ainda a possibilidade de manipular os parâmetros que evidenciaram maior influência na concretização dos objetivos inerentes à locomoção pretendida de modo a conseguir maximizar a qualidade da marcha do bípede.

Por fim, seria ainda pertinente testar outras funções de custo tais como o consumo mínimo de energia assim como a minimização do torque, entre outras, de modo a constatar se as funções utilizadas na nossa proposta podem ser complementadas com outras funções ou se, na realidade, existem outras funções capazes de substituir as nossas e conseguir obter melhores soluções.

Bibliografia

- [1] D. Gong, J. Yan e G. Zuo, "A Review of Gait Optimization Based on Evolutionary Computation," *Applied Computational Intelligence and Soft Computing*, 2010.
- [2] S. Pouya, E. Aydin, R. Mockel e A. J. Ijspeert, "Locomotion Gait Optimization For Modular Robots; Coevolving Morphology and Control," *The European Future Technologies Conference and Exhibition*, pp. 320-322, 2011.
- [3] J. Belda-Lois, S. Horno, I. Bermejo-Bosch, J. Moreno, J. Pons, D. Farina, M. Losa, F. Tamburella, A. Ramos, A. Carla, T. Solis-Escalante, C. Brunner e M. Rea, "Rehabilitation of gait after stroke: a review towards a top-down approach," *Journal of NeuroEngineering and Rehabilitation*, 2011.
- [4] J. Hu, A. Edsinger, Y. Lim, N. Donaldson, M. Solano, A. Solochek e R. Marchessault, "An Advanced Medical Robotic System Augmenting Healthcare Capabilities – Robotic Nursing Assistant," *International Conference on Robotics and Automation*, 2011.
- [5] K. Kaneko, S. Kajita, K. Yokoi, V. Hugel, P. Blazevic e P. Coiffet, "Design of LRP Humanoid Robot and its Control Method," *IEEE International Worksop on Robot and Human Interactive Communication*, 2001.
- [6] M. Oliveira, V. Matos, C. Santos, L. Costa e P. Barbosa, "Multi-objective Parameter CPG Optimization for Gait Generation of a Biped Robot," *2013 IEEE International Conference on Robotics and Automation*, 2012 (submetido).
- [7] G. Capi, Y. Nasub, L. Barolli e K. Mitobe, "Real time gait generation for autonomous humanoid robots: A case study for walking," *Robotics and Autonomous Systems*, p. 107–116, 2003.
- [8] Q. Huang, k. Yokoi, S. Kajita, K. Kaneko, H. Arai, N. Koyachi e K. Tanie, "Planning Walking Patterns for a Biped Robot," *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION*, vol. 17, 2001.
- [9] H. Inada e K. Ishii, "Bipedal walk using a Central Pattern Generator,"

International Congress Series, p. 185– 188, 2004.

- [10] C. Vaughan, “Theories of bipedal walking: an odyssey,” *Journal of Biomechanics*, p. 513–523, 2002.
- [11] Ijspeert, “Central pattern generators for locomotion control in animals and robots: A review,” *Neural Networks*, p. 642–653, 2008.
- [12] B. Mulloney e C. Smarandache, “Fifty years of CPGs: two neuroethological papers that shaped the course of neuroscience,” *Frontiers in Behavioral Neuroscience*, 2010.
- [13] J. G. Jones, E. Tansey e D. Stuart, “Thomas Graham Brown (1882–1965): Behind the Scenes at the Cardiff Institute of Physiology,” *Journal of the History of the Neurosciences*, p. 188–209, 2011.
- [14] G. Dip, V. Prahlad e P. Kien, “Genetic algorithm-based optimal bipedal walking gait synthesis considering tradeoff between stability margin and speed,” *Robotica*, vol. 27, pp. 355-365, 2009.
- [15] S. Rutishauser, A. Sprowitz, L. Righetti e A. Ijspeert, “Passive compliant quadruped robot using central pattern generators for locomotion control,” *Biomedical Robotics and Biomechatronics, 2nd IEEE RAS & EMBS International Conference* , pp. 710-715, 2008.
- [16] H. Kimura, Y. Fukuoka e A. Cohen, “Adaptive Dynamic Walking of a Quadruped Robot on Irregular Terrain Based on Biological Concepts,” *The International Journal of Robotic Research*, 2003.
- [17] k. Matsuoka, “Sustained Oscillations Generated by Mutually Inhibiting Neurons with Adaptation,” *Biological Cybernetics*, pp. 367-376, 1985.
- [18] G. Taga, “A model of the neuro-musculo-skeletal system for anticipatory adjustment of human locomotion during obstacle avoidance,” *Biological Cybernetics*, p. 9–17, 1998.
- [19] S. Miyakoshi, G. Taga, Y. Kuniyoshi e A. Nagakubo, “Three dimensional bipedal stepping motion using neural oscillators-towards humanoid motion in the real world,” *Intelligent Robots and Systems*, vol. 1, pp. 84-89, 1998.
- [20] H. Cruse, T. Kindermann, M. Schumm, J. Dean e J. Schmitz, “Walknet—a

biologically inspired network to control six-legged walking,” *Neural Networks*, vol. 11, p. 1435–1447, 1998.

- [21] T. Zilic, D. Pavkovic e D. Zorc, “Modeling and control of a pneumatically actuated inverted pendulum,” *ISA Transactions*, pp. 327-335, 2009.
- [22] A. Kuo, “The six determinants of gait and the inverted pendulum analogy: A dynamic walking perspective,” *Human Movement science pendulum analogy: A dynamic walking perspective*, pp. 617-656, 2007.
- [23] A. Kuo, M. Donelan e A. Ruina, “Energetic Consequences of Walking Like an Inverted Pendulum: Step-to-Step Transitions,” *Exercise and Sport Sciences Reviews*, vol. 33, 2005.
- [24] T. Kwon e J. Hodgins, “Control Systems for Human Running using an Inverted Pendulum Model and a Reference Motion Capture Sequence,” *Symposium on Computer Animation*, 2010.
- [25] J. Park e H. Cho, “An On-Line Trajectory Modifier for the Base Link of Biped Robots To Enhance Locomotion Stability,” *International Conference on Robotics & Automation*, 2000.
- [26] B. Vanderborght, B. Verrelst, R. Ham, M. Damme e D. Lefeber, “Objective locomotion parameters based inverted pendulum trajectory generator,” *Robotics and Autonomous Systems*, p. 738–750, 2008.
- [27] A. Albert e W. Gerth, “Analytic Path Planning Algorithms for Bipedal Robots without a Trunk,” *Journal of Intelligent and Robotic Systems*, p. 109–127, 2002.
- [28] E. Garcia, M. Jimenez, P. Santos e M. Armada, “The Evolution of Robotics Research,” *IEEE Robotics & Automation Magazine*, 2007.
- [29] M. Armand, J. Huisson e A. Patla, “Stepping Over Obstacles During Locomotion: Insights from Multiobjective Optimization on Set of Input Parameters,” *TRANSACTIONS ON REHABILITATION ENGINEERING*, vol. 6, 1998.
- [30] T. Jason, “Darwin + Robots = Evolutionary Robotics: Challenges in Automatic Robot Synthesis,” *2nd International Conference on Artificial Intelligence in Engineering and Technology*, pp. 7-13, 2004.
- [31] H. Lim e A. Takanishi, “Biped walking robots created at Waseda University: WL

and WABIAN family,” *Philosophical Transactions of The Royal Society*, pp. 49-64, 2012.

- [32] M. Akhtaruzzaman e A. Shafie, “Evolution of Humanoid Robot and Contribution of Various Countries in Advancing the Research and Development of the Platform,” *International Conference on Control, Automation and Systems*, 2010.
- [33] C. Niehaus, T. Rofer e T. Laue, “Gait Optimization on a Humanoid Robot using Particle Swarm Optimization”.
- [34] Y. Hurmuzlu, F. Génot e B. Brogliato, “Modeling, stability and control of biped robots—a general framework,” *Automatica*, p. 1647 – 1664, 2004.
- [35] R. Ambrose, H. Aldridge, R. Askew, R. Burridge, W. Bluethmann, M. Diftler, C. Lovchik, D. Magruder e F. Rehnmark, “Robonaut: NASA’s Space Humanoid,” *Humanoid robotics*, 2000.
- [36] D. Gouaillier, V. Hugel e P. Blazevic, “The NAO humanoid: a combination of performance and affordability,” 2008.
- [37] L. R. D. Honda Motor Co., *Honda ASIMO Technical Information*, 2007.
- [38] S. Ushida, K. Yoshimi, T. Okatani e K. Deguchi, “The Importance of Gaze Control Mechanism on Vision-based Motion Control of a Biped Robot,” *International Conference on Intelligent Robots and Systems*, 2006.
- [39] K. Muecke e D. Hong, “DARwIn’s Evolution: Development of a Humanoid Robot,” *Conference on Intelligent Robots and Systems*, 2007.
- [40] I. Ha, Y. Tamura, H. Asama, J. Han e D. Hong, “Development of Open Humanoid Platform DARwIn-OP,” *SICE Annual Conference* , 2011.
- [41] A. Nelson, G. Barlow e L. Doitsidis, “Fitness functions in evolutionary robotics: A survey and analysis,” *Robotics and Autonomous Systems*, pp. 345-370, 2009.
- [42] P. Sardain e G. Bessonet, “Forces Acting on a Biped Robot. Center of Pressure—Zero Moment Point,” *IEEE Transactions on Systems and Cybernetics - Part A: Systems and Humans*, vol. 34, 2004.
- [43] M. Vukobratovic e B. Borovac, “Zero-Moment-Point - Thirty Five Years of its Life,” *International Journal of Humanoid Robotics*, vol. 1, p. 157–173, 2004.

- [44] G. Capi e Y. Mitobe, "Optimal Multi-Criteria Humanoid Robot Gait Synthesis - an Evolutionary Approach," *International Journal of Innovative Computing, Information and Control*, vol. 2, 2006.
- [45] S. Ha, Y. Han e H. Hahn, "Adaptive Gait Pattern Generation of Biped Robot based on Human's Gait Pattern Analysis," *Proceedings of World Academy of Science, Engineering and Technology*, vol. 23, 2007.
- [46] Y. Fujimoto, "Minimum Energy Biped Running Gait and Development of Energy Regeneration Leg," 2004.
- [47] M. Hardt, K. Kreutz-Delgado e J. Helton, "Optimal Biped Walking with a Complete Dynamical Model," *Proceedings of the 38 Conference on Decision & Control*, 1999.
- [48] S. Fan, M. Sun e M. Shi, "Real-time Gait Generation for Humanoid Robot Based on Fuzzy Neural Networks," *Third International Conference on Natural Computation*, 2007.
- [49] J. Zhao, L. Qi, J. Yan e Y. Zhu, "Dynamic Stability Gait Planning of kid Humanoid Robot," *International Conference on Robotics and Biomimetics*, 2009.
- [50] G. Capi, Y. Nasu, M. Yamano e K. Mitobe, "Multicriteria Optimal Humanoid Robot Motion Generation," *Humanoid Robots, New Developments*, 2007.
- [51] C. Shih, S. Chung e W. Gruver, "Trajectory Synthesis and Physical Admissibility for a Biped Robot During the Single-Support Phase," 1990.
- [52] Y. Nakamura e Q. Huang, "Sensory Reflex Control for Humanoid Walking," *IEEE Transactions on Robotics*, vol. 21, 2005.
- [53] T. Arakawa e T. Fukuda, "Natural motion generation of biped locomotion robot using hierarchical trajectory generation method consisting of GA, EP layers," *Robotics and Automation*, vol. 1, pp. 211-216, 1997.
- [54] A. Goswami, "Foot rotation indicator point: A new gait planning tool to evaluate postural stability of biped robots," *Robotics and Automation*, vol. 1, pp. 47-52, 1999.
- [55] F. Takemori, A. Kitamura e D. Kushida, "Constraint of Center of Gravity in a Biped Walking Robot via Sliding Mode Approach," *Proceedings of 2004 IEEE/RSJ*

International Conference on intelligent Robots and Systems, 2004.

- [56] C. Liu e Q. Chen, "Walking Control Strategy for Biped Robots Based on Central Pattern Generator," *IEEE International Conference on Robotics and Automation*, 2012.
- [57] S. Kudoh e T. Komura, "Continuous Gait-Pattern Generation for Biped Robots," *Conference on Intelligent Robots and Systems*, 2003.
- [58] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal e M. Kawato, "Learning from demonstration and adaptation of biped locomotion," *Robotics and Autonomous Systems*, p. 79–91, 2004.
- [59] K. Deb, "Introduction to Evolutionary Multiobjective Optimization," *Springer-Verlag Berlin Heidelberg*, p. 59–96, 2008.
- [60] E. Nakano, H. Imamizu, R. Osu, Y. Uno, H. Gomi, T. Yoshioka e M. Kawato, "Quantitative Examinations of Internal Representations for Arm Trajectory Planning: Minimum Commanded Torque Change Model," *The American Physiological Society*, p. 2140–2155, 1999.
- [61] P. Channon, S. Hopkins e D. Pham, "Simulation and optimization of Gait for a Bipedal robot," *Proc. 7th Int. Conf. on Mathematical and Computer Modelling*, vol. 14, pp. 463-467, 1990.
- [62] R. Becerra e C. Coello, "Optimization with Constraints using a Cultured Differential Evolution Approach," 2005.
- [63] A. Coello e A. Jaimes, "Multi-Objective Evolutionary Algorithms: A Review of the State-of-the-Art and some of their Applications in Chemical Engineering," *World Scientific Book*, 2007.
- [64] I. Sbalzarini, S. Muller e P. Koumoutsakos, "Multiobjective optimization using evolutionary algorithms," *Center for Turbulence Research - Proceedings of the Summer Program*, 2000.
- [65] E. Zitzler, K. Deb e L. Thiele, "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," *Evolutionary Computation*, pp. 173-195, 2000.
- [66] E. Zitzler e L. Thiele, "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach," *Transactions on Evolutionary*

Computation, vol. 3, 1999.

- [67] G. Capi, M. Yokota e K. Mitobe, "A New Humanoid Robot Gait Generation based on Multiobjective Optimization," *International Conference on Advanced Intelligent Mechatronics*, 2005.
- [68] M. Emmerich, N. Beume e B. Naujoks, "An EMO Algorithm Using the Hypervolume Measure as Selection Criterion," *Springer-Verlag Berlin Heidelberg*, p. 62–76, 2005.
- [69] M. Fleischer, "The Measure of Pareto Optima: Applications to Multi-objective Metaheuristics," *Springer-Verlag Berlin Heidelberg*, p. 519–533, 2003.
- [70] A. Auger, J. Bader, D. Brockhoff e E. Zitzler, "Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications," *Theoretical Computer Science*, pp. 75-103, 2012.
- [71] M. Mitchell, *An Introduction to Genetic Algorithms*, Massachusetts Institute of Technology: MIT press, 1998.
- [72] R. Saravanan, S. Ramabalan e P. Babu, "Optimum static balancing of an industrial robot mechanism," *Engineering Applications of Artificial Intelligence*, p. 824–834, 2008.
- [73] D. Goldberg e K. Deb, "A Comparative Analysis of Selection Schemes Used in Genetic Algorithms," *Foundation of Genetic Algorithms*, 1991.
- [74] B. Miller e D. Goldberg, "Genetic Algorithms," 1995.
- [75] K. Seo e S. Hyun, "A Comparative Study between Genetic Algorithm and Genetic Programming Based Gait Generation Methods for Quadruped Robots," em *Applications of Evolutionary Computation*, University, Seoul, Korea, Springer Berlin Heidelberg, 2010, pp. 352-360.
- [76] M. Lewis e A. Fagg, "Genetic Programming Approach to the Construction of a Neural Network for Control of a Walking Robot," *International Conference on Robotics and Automation*, 1992.
- [77] C. Fonlupt, D. Robilliard e V. Marion-Poty, "Continuous Schemes for Program Evolution," em *Genetic Programming – New Approaches and Successful Applications*, 2012.

- [78] L. Hu, C. Zhou e Z. Sun, "Biped Gait Optimization Using Estimation of Distribution Algorithm," *Proceedings of 2005 5th IEEE-RAS International Conference on Humanoid Robots*, 2005.
- [79] K. Wolff e P. Nordin, "Learning Biped Locomotion from First Principles on a Simulated Humanoid Robot Using Linear Genetic Programming," *Springer-Verlag Berlin Heidelberg*, p. 495–506, 2003.
- [80] J. Eperješi, "Gait Optimization of AIBO Robot Based on Interactive Evolutionary Computation," *6th International Symposium on Applied Machine Intelligence and Informatics*, 2008.
- [81] K. Deb, S. Agarwal e T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *Transactions on Evolutionary Computation*, vol. 6, 2002.
- [82] A. Konak, D. Coit e A. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," *Reliability Engineering and System Safety*, 2006.
- [83] J. Zheng, R. Shen e J. Zou, "Enhancing Diversity for NSGA-II in Evolutionary Multi-Objective Optimization," *8th International Conference on Natural Computation (ICNC 2012)*, 2012.
- [84] K. Deb, "Multi-Objective Optimization Using Evolutionary Algorithms: An Introduction," *KanGAL Report*, 2011.
- [85] K. Tran, "Elitist Non-Dominated Sorting GA-II (NSGA-II) as a Parameter-less Multi-Objective Genetic Algorithm," *SoutheastCon Proceedings*, pp. 359-367 , 2005.
- [86] Y. Yusoff, M. Ngadiman e A. Zain, "Overview of NSGA-II for Optimizing Machining Process Parameters," *Procedia Engineering*, 2011.
- [87] C. Santos, M. Oliveira, L. Costa, V. Matos e M. Ferreira, "Multi-objective Parameter CPG Optimization for Gait Generation of a Quadruped Robot Considering Behavioral Diversity," *International Conference on Intelligent Robots and Systems*, 2011.
- [88] K. Song e T. Chu, "Reinforcement learning and its application to force control of an industrial robot," *Control Engineering Practice*, pp. 37-44, 1998.
- [89] G. Yen e T. Hickey, "Reinforcement learning algorithms for robotic navigation in dynamic environments," *ISA Transactions*, p. 217–230, 2003.

- [90] N. Kohl e P. Stone, "Policy Gradient Reinforcement Learning for Fast Quadrupedal Locomotion," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2619-2624, 2004.
- [91] M. Erden e K. Leblebicioglu, "Free gait generation with reinforcement learning for a six-legged robot," *Robotics and Autonomous Systems*, p. 199–212, 2008.
- [92] I. Vincent e Q. Sun, "A combined reactive and reinforcement learning controller for an autonomous tracked vehicle," *Robotics and Autonomous Systems*, pp. 599-698, 2012.
- [93] M. Svinin, K. Yamada e K. Ueda, "Emergent Synthesis of motion patterns for locomotion robots," *artificial Intelligence in Engineering*, pp. 353-363, 2001.
- [94] Y. Nakamura, T. Mori, M. Sato e S. Ishii, "Reinforcement learning for a biped robot based on a CPG-actor-critic method," *Neural Networks*, p. 723–735, 2007.
- [95] N. Rokbani, E. Benbousaada, B. Ammar e A. Alimi, "Biped Robot Control Using Particle Swarm Optimization," *Systems Man and Cybernetics (SMC)*, pp. 506-512, 2010.
- [96] J. Kim, S. Park e J. Lee, "Experience Repository based Particle Swarm Optimization for Evolutionary Robotics," *ICROS-SICE International Joint Conference*, 2009.
- [97] J. Shan, C. Junshi e C. Jiapin, "Design of central pattern generator for humanoid robot walking based on multi-objective GA," *Intelligent Robots and Systems*, 2000.
- [98] V. Matos e C. Santos, "Humanoids: Central pattern generators with phase regulation for the control of humanoid locomotion," *12th IEEE-RAS International Conference on Humanoid Robots*, 2012.
- [99] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, pp. 311-338, 2000.
- [100] "<http://asbg.dei.uminho.pt/>," [Online].
- [101] J. Lee, M. Kim e J. J. Lee, "Multi-objective walking trajectories generation for a biped robot," *In Intelligent Robots and Systems*, 2004.
- [102] J. v. d. Kieboom, *Biped Locomotion and Stability - A Practical Approach*, 2009.

