

Alexis López Figueroa

Modelagem e otimização por algoritmos genéticos de padrões de marcha quase-estática de robôs bípedes planos

DISSERTAÇÃO DE MESTRADO

DEPARTAMENTO DE ENGENHARIA MECÂNICA

Programa de Pós-Graduação em
Engenharia Mecânica

Rio de Janeiro
Setembro de 2016



Alexis López Figueroa

**Modelagem e otimização por algoritmos genéticos
de padrões de marcha quase-estática de robôs
bípedes planos**

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Engenharia Mecânica do Departamento de Engenharia Mecânica da PUC-Rio.

Orientador: Prof. Marco Antonio Meggiolaro

Rio de Janeiro
Setembro de 2016



Alexis López Figueroa

**Modelagem e otimização por algoritmos genéticos
de padrões de marcha quase-estática de robôs
bípedes planos**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-Graduação em Engenharia Mecânica do Departamento de Engenharia Mecânica da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Marco Antonio Meggiolaro, Ph.D

Orientador

Departamento de Engenharia Mecânica – PUC-Rio

Prof. Mauro Speranza Neto, D.Sc

Departamento de Engenharia Mecânica – PUC-Rio

Prof. Max Suell Dutra, Dr.-Ing.

Programa de Engenharia Mecânica-COPPE-UFRJ

Prof. Márcio da Silveira Carvalho

Coordinator of the Centro Técnico Científico da PUC-Rio

Rio de Janeiro, 12 de setembro de 2016

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Alexis López Figueroa

Estudante de Mestrado em Engenharia Mecânica na Pontifícia Universidade Católica do Rio de Janeiro. Possui graduação em Engenharia Mecatrônica pela Universidade Santo Tomas de Aquino. Sua linha de pesquisa e interesse envolve sistemas robóticos, automação e projeto mecatrônico.

Ficha Catalográfica

<p>López Figueroa, Alexis</p> <p>Modelagem e otimização por algoritmos genéticos de padrões de marcha quase-estática de robôs bípedes planos / Alexis López Figueroa; orientador: Marco Antonio Meggiolaro. – 2016. 126 f. : il. color. ; 30 cm</p> <p>Dissertação (mestrado)–Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Mecânica, 2016.</p> <p>Inclui bibliografia</p> <p>1. Engenharia Mecânica – Teses. 2. Padrão de marcha. 3. Robô bípede. 4. Algoritmos genéticos. 5. Otimização. I. Meggiolaro, Marco Antonio. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Mecânica. III. Título.</p>

CDD: 621

À minha família pelo apoio e amor incondicional.

Agradecimentos

Agradeço à minha mãe que com suas palavras me brinda energia para continuar cada dia, meu pai, exemplo de paciência e persistência.

Agradeço ao Giovanni pela ajuda constante.

Aos meus colegas da PUC-Rio

A todos os professores e funcionários do Departamento.

Capes, pelos auxílios concedidos.

Aos professores Marco Antonio Meggiolaro, Mauro Speranza Neto e Max Suell Dutra.

Resumo

López Figueroa, Alexis; Meggiolaro, Marco Antonio. **Modelagem e otimização por algoritmos genéticos de padrões de marcha quase-estática de robôs bípedes planos.** Rio de Janeiro, 2016. 126p. Dissertação de Mestrado - Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

Este trabalho aborda a modelagem cinemática e estática de um robô bípede plano, visando a otimização de padrões de marcha quase-estática. Os padrões de marcha são avaliados e otimizados utilizando algoritmos genéticos para minimizar a relação entre a energia consumida pelos motores do robô e o deslocamento obtido em cada passo, sempre garantindo equilíbrio estático dentro de uma margem de segurança. A evolução dos diferentes ângulos das juntas do robô para os padrões de marcha resultantes são armazenados indiretamente ajustando os coeficientes de funções polinomiais ou trigonométricas. Além disso, duas abordagens distintas são exploradas para os algoritmos genéticos: uma visando a otimização de um passo completo da marcha, e outra para otimizar de forma independente sub-passos dentro de cada passo. O melhor padrão de marcha obtido diminuiu a relação energia/deslocamento em até 18% em relação a outros padrões de caminhada não-otimizados. O melhor padrão obtido foi programado em um robô Bioloid GP para validação física da estabilidade da marcha sobre uma superfície plana sem obstáculos.

Palavras-Chave

Padrão de marcha; Robô bípede; Algoritmos genéticos; Otimização.

Abstract

López Figueroa, Alexis; Meggiolaro, Marco Antonio (Advisor). **Modeling and optimization with genetic algorithms of quasi-static gait patterns in planar biped robots**. Rio de Janeiro, 2016. 126p. Master Dissertation – Mechanical Engineering Department, Pontifical Catholic University of Rio de Janeiro.

This work deals with the kinematic and static modeling of a planar biped robot, aiming to optimize quasi-static gait patterns. The gait patterns are evaluated and optimized using genetic algorithms to minimize the relationship between the energy consumed by the robot motors and the displacement obtained in each robot step, always guaranteeing static equilibrium within a safety margin. The evolution of the different joint angles for the gait patterns are indirectly stored by adjusting the coefficients of polynomial or trigonometric functions. In addition, two different approaches are explored for the genetic algorithms: one aiming to optimize a complete step from the gait, and other to independently optimize sub-step within each step. The gait pattern obtained decreased the energy/displacement ratio by up to 18% with respect to other non-optimized patterns. The best pattern has been programmed into a Bioloid GP robot to physically validate the gait stability on a planar surface without obstacles.

Key words

Gait pattern; Biped robot; Genetic algorithms; Optimization.

Sumário

1	Introdução	16
1.1.	Motivação	16
1.2.	Objetivo	17
1.3.	Principais robôs humanoides	17
1.4.	Evolução do caminhar em robôs bípedes	23
1.5.	Tipos de caminhada e geradores de padrões	28
1.6.	Estrutura da dissertação	33
2	Robô Bípede Plano	34
2.1.	Representação do robô bípede físico em simulação	34
2.2.	Definição dos ângulos envolvidos	36
2.3.	Cinemática, parâmetros de Denavit-Hartenberg	41
2.4.	Estática	46
2.5.	Controle Cinemático	48
3	Algoritmos Genéticos	56
3.1.	Geração de variáveis utilizando algoritmos genéticos	58
3.2.	Função Objetivo	59
3.3.	Penalizações nos Indivíduos	61
3.4.	Configuração do Cromossomo.	62
3.4.1.	Cromossomo função polinomial	62
3.4.2.	Cromossomo função trigonométrica (cosseno)	63
3.4.3.	Cromossomo passo completo	63
3.4.4.	Cromossomo sub-passo	64
3.4.5.	Cromossomo de controle	66
4	Resultados	67
4.1.	Controle Cinemático	67
4.1.1.	Ajuste do controlador com Algoritmos genéticos.	73
4.2.	Caminhada heurística	76
4.2.1.	Simulações	77

4.2.2. Experimento inicial	79
4.3. Caminhada com algoritmos genéticos	80
4.3.1. Função polinomial	80
4.3.2. Função trigonométrica (cosseno)	83
4.3.3. Passo completo	86
4.3.4. Sub-passos	93
4.3.5. Melhor indivíduo	97
5 Conclusões	100
5.1. Sugestões para trabalhos futuros	101
6 Referências bibliográficas	102
Apêndice A Desenhos SolidWorks	106
Apêndice B. Dados motor Dynamixel AX-18A	109
Apêndice C. Principais linhas de código	111
Apêndice D. Exportação de “SolidWorks” para “Simulink de Matlab”	117
Apêndice E. Desempenho e robustez	123

Lista de Figuras

Figura 1. Competição robótica Darpa 2015	18
Figura 2. Robô Drc-Hubo	20
Figura 3. Robô Nasa's R5	20
Figura 4. Robô Atlas.....	21
Figura 5. Robô Petman	22
Figura 6. Robô Asimo	22
Figura 7. Bípede WAP-1	23
Figura 8. Bípede WAP-2	24
Figura 9. Bípede WAP-3	24
Figura 10. Bípede WL-5.....	25
Figura 11. Robô Wabot 1	26
Figura 12. Robô WL-10RD	26
Figura 13. Robô E0.....	27
Figura 14. Evolução robô Asimo.....	28
Figura 15. Projeção do centro de massa e polígono de apoio.....	29
Figura 16. Ciclo de marcha de uma perna humana	30
Figura 17. Ciclo de marcha humana	30
Figura 18. Caminhada quase-estática	31
Figura 19. Captura de movimento	31
Figura 20. Interface de usuário pin/drag.....	32
Figura 21. Interface de usuário Bioloid GP.....	32
Figura 22. RRT-Connect	33
Figura 23. Representação do robô bípede físico em simulação	35
Figura 24. Desenhos de Solidworks e Matlab	36
Figura 25. Ângulos α fase 1.....	37
Figura 26. Ângulos α fase 2.....	37
Figura 27. Ângulos α , visualização	38
Figura 28. Ângulos de transição	39
Figura 29. Ângulos dos motores físicos	40
Figura 30. Ângulos de programação	41
Figura 31. Sistemas coordenados adjacentes com a notação DH.....	42

Figura 32. Matriz de posição orientação	43
Figura 33. Parâmetros DH na representação do bípede.....	45
Figura 34. Cinemática direta simplificada.....	45
Figura 35. Equilíbrio estático	47
Figura 36. Desenho 3D importado do “SolidWorks” para “Simulink”	48
Figura 37. Sistema no “SimMechanics” primeira geração.....	49
Figura 38. Transição suave entre fotogramas.....	50
Figura 39. Malha de controle com bloco PID auto-ajustável	52
Figura 40. Malha de controle com bloco PID Simulink.....	53
Figura 41. Interface de ajuste bloco Pidcontroller.....	54
Figura 42. Malha de controle fase 2, controle de posição e orientação.....	55
Figura 43. Processo evolutivo do algoritmo genético	56
Figura 44. Objetivo do GA para os movimentos do robô bípede.....	58
Figura 45. Fotogramas sucessivos	58
Figura 46. Variação de parâmetros do controlador	59
Figura 47. Penalizações, movimentos do bípede.....	61
Figura 48. Possíveis trajetórias do pé para diferentes cromossomos	64
Figura 49. Possíveis comportamentos durante as evoluções sub-passo	65
Figura 50. Possíveis trajetórias do pé construídas com sub-passo	65
Figura 51. Transição de entradas angulares desejadas	67
Figura 52. Resposta dos motores ao controle auto ajustado.....	68
Figura 53. Resposta do motor 1 ao controle auto-ajustado	69
Figura 54. Resposta ao degrau motor 1 com bloco auto ajustado	69
Figura 55. Trajetória do extremo, controlador PID.....	70
Figura 56. Orientação, controlador PID	70
Figura 57. Transição de entradas desejadas.....	71
Figura 58. Trajetória do quadril, controlador com parâmetros heurísticos	72
Figura 59. Posição do quadril, controlador com parâmetros heurísticos.....	72
Figura 60. Orientação, controlador com parâmetros heurísticos.....	73
Figura 61. Trajetória do quadril, parâmetros ajustados pelo GA	74
Figura 62. Posição do quadril, parâmetros ajustados pelo GA.....	74
Figura 63. Orientação do quadril, parâmetros ajustados pelo GA.....	75
Figura 64. Comparação entre o esqueleto de um avestruz e de um ser humano ...	76
Figura 65. Fotogramas do caminhar de uma galinha.....	77

Figura 66. Simulação caminhada heurística	78
Figura 67. Fotogramas do robô bípede físico montado	79
Figura 68. Resposta do GA com função polinômio	81
Figura 69. Ângulos dos motores, para função polinômio	81
Figura 70. Energia consumida, para função polinômio	82
Figura 71. Evolução para função polinômio	82
Figura 72. Evolução da trajetória do pé, para função polinômio	83
Figura 73. Resposta AG função cosseno	84
Figura 74. Ângulos dos motores, para função cosseno	84
Figura 75. Energia consumida, para função cosseno.....	85
Figura 76. Evolução para função cosseno	86
Figura 77. Evolução trajetória do pé, para função cosseno	86
Figura 78. Resposta do GA para passo completo	87
Figura 79. Ângulos motores, passo completo	88
Figura 80. Energia consumida, passo completo	88
Figura 81. Evolução para passo completo.	89
Figura 82. Evolução da trajetória do pé, para passo completo	89
Figura 83. Resposta AG, passo completo B	90
Figura 84. Ângulos motores, passo completo B	91
Figura 85. Energia consumida, passo completo B.....	92
Figura 86. Evolução para o passo completo B	92
Figura 87. Evolução trajetória do pé, passo completo B	93
Figura 88. Resposta do GA, sub-passo resposta A.....	94
Figura 89. Ângulos dos motores, sub-passo resposta A	94
Figura 90. Energia consumida, sub-passo resposta A	95
Figura 91. Resposta GA, sub-passo B	96
Figura 92. Ângulos motores, sub-passo B	96
Figura 93. Energia consumida, sub-passo B.....	97
Figura 94. Trajetórias do pé, respostas por GA	98
Figura 95. Robô Bioloid GP	106
Figura 96. Robô bípede planar, desenho Solidworks	106
Figura 97. Motor AX-18A.....	110
Figura 98. Web baixar SimEscape	117
Figura 99. Web SimEscape	118

Figura 100. Link baixar	118
Figura 101. Registro para Simscape	119
Figura 102. Arquivos SimEscape	119
Figura 103. Mensagem instalação	120
Figura 104. Ativação em Solidworks	120
Figura 105. Salvar em solidWorks .xml	121
Figura 106. Interface Simulink.....	122
Figura 107. Modelo 3D SimMechanics.....	122
Figura 108. Arquitetura de controle de dois graus de liberdade.....	123

Lista de tabelas

Tabela 1. Parâmetros DH das origens dos sistemas coordenados	44
Tabela 2. Parâmetros DH dos centros de massa	44
Tabela 3. Cromossomo polinômio.....	62
Tabela 4. Cromossomo cosseno	63
Tabela 5. Cromossomo passo completo	63
Tabela 6. Cromossomo sub-passo	65
Tabela 7. Cromossomo de controle	66
Tabela 8. Resumo da avaliação de sintonizações do controlador.....	75
Tabela 9. Resumo, dados das respostas GA	99
Tabela 10. Elos robô bípede	107
Tabela 11. Dimensões peças robô bípede.....	108
Tabela 12. Motor AX-18A	110

*“Cuando un sueño se te muera
o entre en coma una ilusión,
no lo entierres ni lo llores, resucítalo.*

*Y jamás des por perdida
la partida, cree en ti.
y aunque duelan, las heridas curarán”.*

Mago de oz.

1 Introdução

1.1.Motivação

Projetar um robô que possa duplicar os complexos movimentos humanos e ajudar as pessoas, em diversas situações, seja de assistência, cooperação ou substituição em tarefas de alto risco, é um dos objetivos principais pela qual a robótica humanoide cada dia tem mais investimentos e avanços.

A PUC-Rio e o autor da presente dissertação não estão isolados deste objetivo, a universidade tem dentro de seus laboratórios professores e alunos altamente motivados nesta linha, que interatuam e participam das aulas e diferentes concursos de robótica.

Inicialmente com a geração e otimização de padrões de marcha bípede quase-estática, tem-se a intenção de transportar carga delicada sobre terreno uniforme, na qual se precisa de movimentos suaves e lentos com o maior aproveitamento das baterias em relação ao espaço percorrido; o presente documento só é um passo para continuar com futuros trabalhos e avançar no mundo crescente da robótica humanoide.

As hipóteses básicas da presente dissertação são:

Caminhada quase-estática, ou seja, se é parado o movimento, a estrutura deve se manter equilibrada, pois o centro de massa está dentro do polígono de apoio dos pés do robô em todo o momento.

O modelamento matemático é cinemático, logo não são consideradas condições dinâmicas.

As simulações não impedem colisões entre peças do robô nem o entorno, já que não tem detector de colisões.

A caminhada do robô é sobre terreno uniforme sem obstáculos.

O contato com o chão é fixo e sem deslizamento.

Os pés estão restritos a permanecerem horizontais (o que poderia se converter em uma limitação em futuros trabalhos ao observar a seção 1.5 Tipos de caminhada e geradores de padrões) e o tronco a 90 graus em relação a horizontal, e considera-se que o mesmo não produz momento externo no quadril.

O robô bípede físico é utilizado apenas para reproduzir o padrão de marcha fornecido em malha aberta, ou seja, não tem implementação de controle no hardware.

Da presente dissertação esperam-se os seguintes resultados:

Simulações do robô bípede em “Matlab” utilizando um modelamento cinemático.

Geração de diferentes padrões de marcha quase-estáticos otimizados e seleção do melhor com a menor relação energia consumida sobre distância.

Caminhada sobre superfície uniforme sem obstáculos para um robô bípede plano.

1.2.Objetivo

O objetivo da presente dissertação é determinar, para um robô bípede planar, uma configuração de ângulos, que mantendo o centro de massa dentro do polígono de apoio, permitam um passo com menor consumo de energia em relação ao deslocamento do pé, utilizando algoritmos genéticos e modelagem cinemática.

1.3.Principais robôs humanoides

Competição DRC

A agência de projetos de pesquisa avançada para defesa “Darpa” (“Defense advanced reserch projects agency” em inglês) é uma agência que investe em tecnologias inovadoras para a segurança nacional dos Estados Unidos da América,

que trabalhando em parceria com universidades e corporações empresariais procuram constantemente melhorar os serviços militares do país [1].

Após o acidente nuclear em Fukushima no Japão em 11 de março de 2011, a Darpa decidiu organizar a competição robótica Darpa DRC, (“Darpa Robotics Challenge” em inglês) com a intenção de motivar o avanço no campo da robótica para atender catástrofes. A competição reúne diferentes equipes do mundo para participar com robôs semiautônomos simulando um ambiente de desastre, robôs esses criados por companhias e universidades (alguns com verba da “Darpa”), que apresentam diferentes tamanhos e formas que incluem pernas, rodas e sistemas híbridos. [2]

A competição consiste em um circuito que tem estações para fazer diferentes tarefas e acumular pontos com cada uma desenvolvida. Aquelas que acumulam pontos são: dirigir um carro, descer do carro, abrir uma porta, abrir uma válvula, criar uma abertura em uma parede, tarefa surpresa, atravessar uma seção com obstáculos e subir escadas. Ganha o robô com mais pontos e o menor tempo empregado, sendo uma hora o limite. A Figura 1 apresenta o circuito da competição 2015 [3].

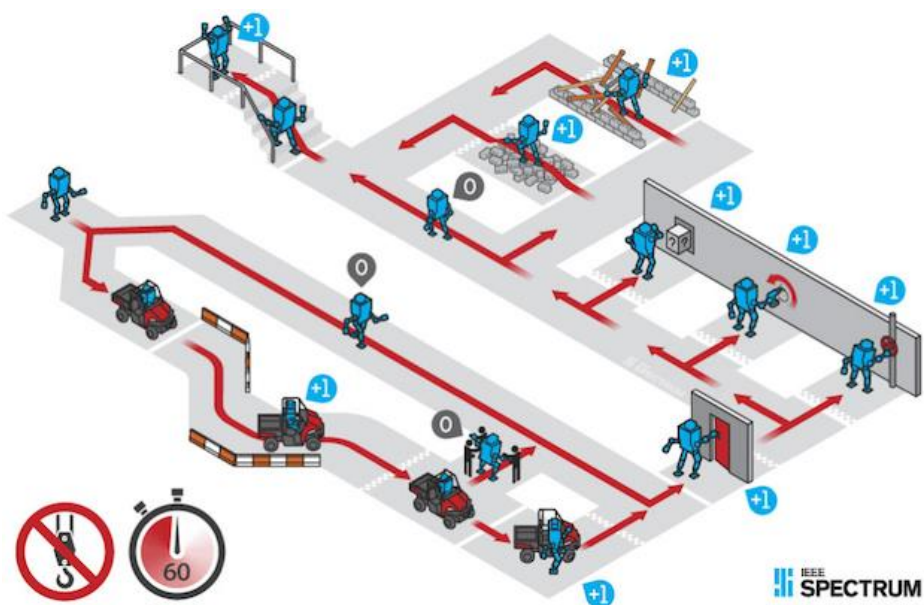


Figura 1. Competição robótica Darpa 2015

Nesta competição podem-se observar alguns dos robôs mais avançados do mundo, os quais são considerados referências, e pontos a serem superados em novos desenvolvimentos de atenção em catástrofes, na qual os mais destacados têm características principais de humanoides, ou seja, apresentam tronco, dois braços e duas pernas, como por exemplo: “Drc-Hubo” [4], “Nasa’s R5” [5], “Atlas” [6].

Além dos robôs participantes na “DRC” anteriormente mencionados, existem outros robôs de igual importância e que também são considerados referências, como por exemplo: “Petman” [7] e “Asimo” [8]. O robô “Asimo” é considerado o humanoide mais avançado no mundo.

Segue uma breve descrição destes robôs.

“Drc-Hubo” [4]

A Figura 2 apresenta o robô campeão da “DRC” 2015, desenvolvido pelo grupo “Kaist” de Coreia do Sul. Suas características principais são: mudar a locomoção entre pernas e rodas, passando desde caminhar em modo bípede a se movimentar em rodas dobrando as pernas, ficando de joelhos; e girar a parte superior do corpo até 180 graus, permitindo trabalhar nos dois sentidos, incluindo marcha para frente e para trás.

“Nasa’s R5” [5]

A Figura 3 mostra o robô humanoide da “NASA” que foi construído para participar na “DARPA Robotics Challenge” 2013. Inicialmente foi concebido para completar manobras de socorro, no entanto o seu principal objetivo é ajudar ou substituir astronautas que trabalham em ambientes extremos no espaço.

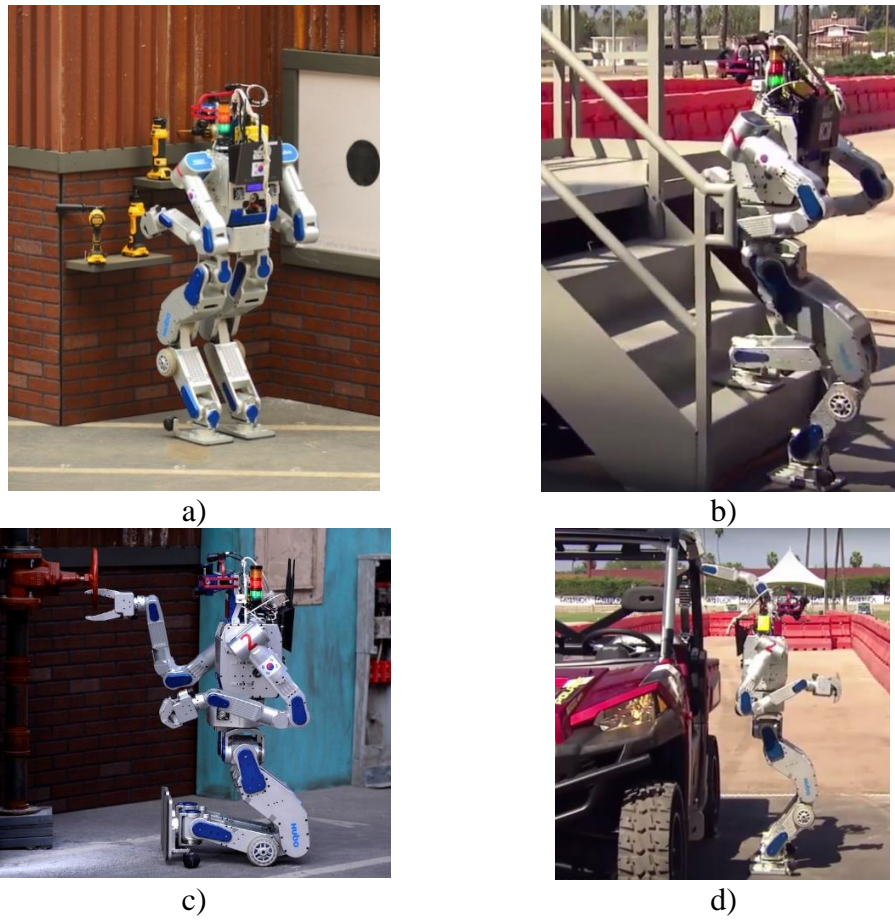


Figura 2. Robô Drc-Hubo



Figura 3. Robô Nasa's R5

“Atlas” - O robô antropomórfico ágil. [6]

O robô apresentado na Figura 4 é um robô de alta mobilidade projetado pela empresa norte-americana “Boston Dynamics” com verba da “Darpa” para se movimentar em terrenos acidentados. Atlas pode andar sobre duas pernas deixando os membros superiores livres para levantar, transportar e manipular o ambiente. Adicionando mãos articuladas com sensores, permite utilizar ferramentas projetadas para humanos. Atlas tem sido um dos robôs mais populares na “DARPA Robotics Challenge”, já que cópias do mostrado na Figura 4a) foram fornecidas como aporte governamental para a competição em 2013. Atualmente a Boston Dynamics é propriedade de “Google”, que comprou em 2013 e em 2016 apresentou uma nova versão, Figura 4b) [9], [10].

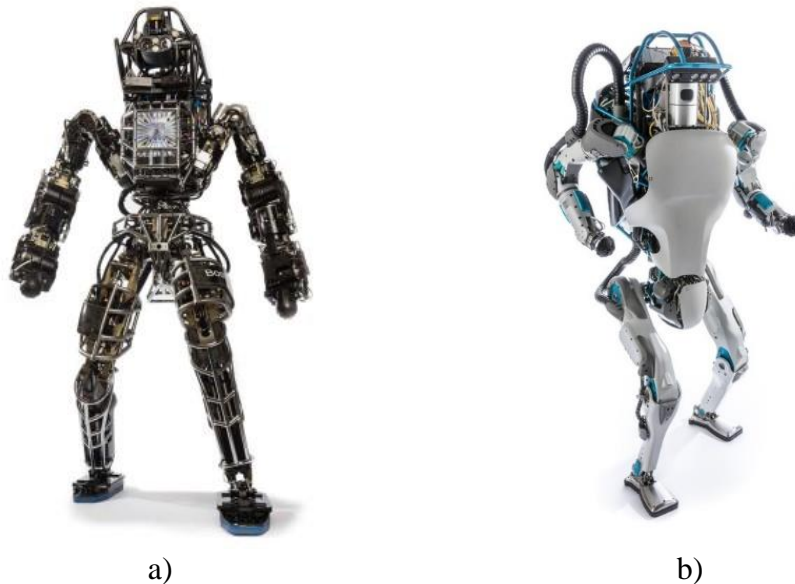


Figura 4. Robô Atlas

“Petman” [7]

O robô apresentado na Figura 5 é um robô antropomórfico projetado pela “Boston Dynamics” com verba da “Darpa” para testar roupa de proteção química. Ágil e com os movimentos de um soldado vestido em condições reais, pode simular as condições fisiológicas humanas com a proteção de sua vestimenta controlando temperatura, umidade e transpiração. A Figura 5a) mostra uma das primeiras etapas de desenvolvimento, apenas como bípede; Figura 5b) e Figura 5c) são etapas humanoide.

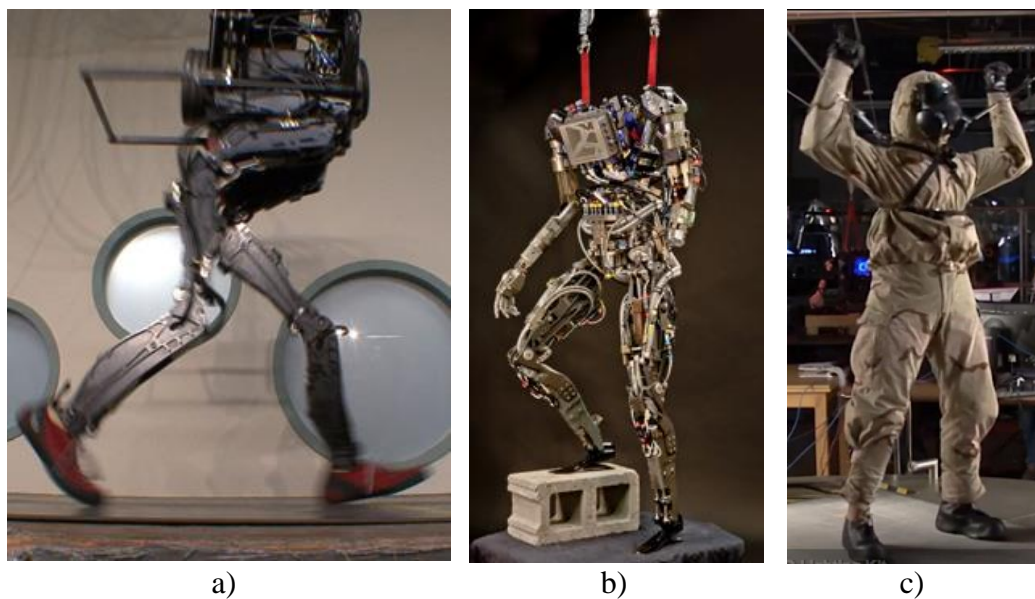


Figura 5. Robô Petman

“Asimo” (“Advanced Step in Innovative Mobility”), 2000 [8]

A Figura 6 apresenta o robô “Asimo” (“Passo Avançado em Mobilidade Inovadora”) criado pela “Honda”, projetado para ajudar as pessoas com dificuldade de mobilidade, além de motivar a juventude a estudar ciências e matemática. Este robô pode caminhar para frente e para trás, correr, abrir portas, subir escadas, evitar obstáculos, empurrar um carrinho de serviço, levar uma bandeja, pegar e levar pequenos objetos, e abrir garrafas. É ideal para ajudar pessoas deitadas em uma cama ou em uma cadeira de rodas, seja em casa ou em escritório. "Pense o potencial que daria a essa pessoa uma vida melhor", disse Jeffrey Smith, porta-voz da Honda.

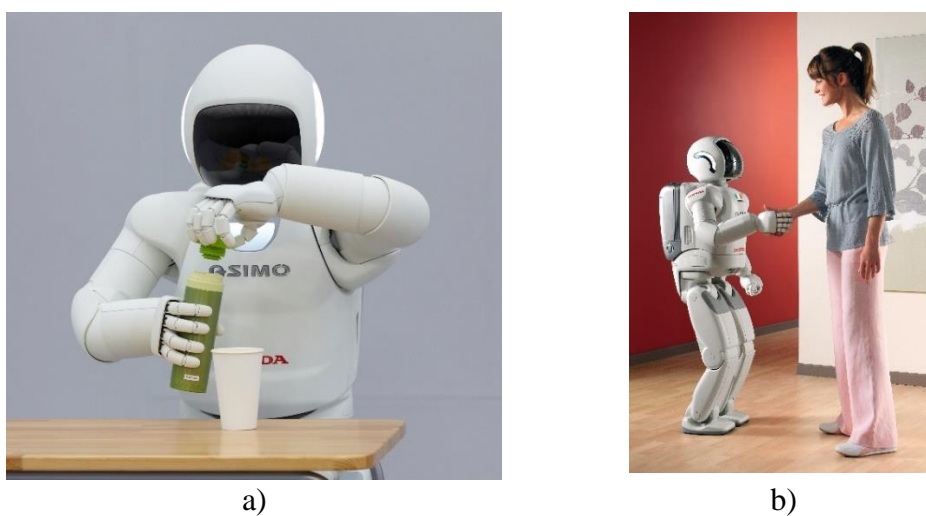


Figura 6. Robô Asimo

1.4.Evolução do caminhar em robôs bípedes

O doutor Ichiro Kato (1925-1994) da universidade “Waseda” [11] no Japão foi o líder da primeira pesquisa de robótica humanoide, a qual tinha por objetivo projetar duas pernas com capacidade de suportar o peso do tronco e andar. Em 1969 foi apresentado o mecanismo antropomórfico pneumático, “WAP-1”, (“Waseda’s anthropomorphic pneumatically-activated pedipulators” em inglês), capaz de andar em um plano. Este mecanismo composto de músculos artificiais de borracha como atuadores e de uma extensão em cada pé que dá suporte no plano frontal é mostrado na Figura 7.

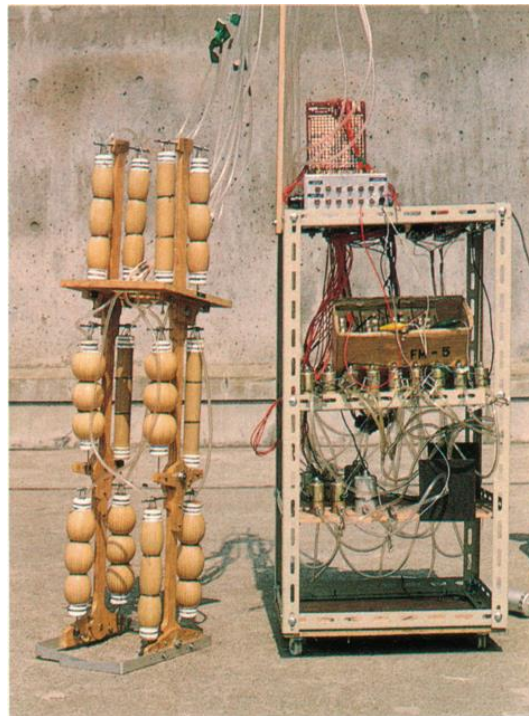


Figura 7. Bípede WAP-1

Em 1970 foi apresentado o “WAP-2”, mostrado na Figura 8, com músculos artificiais pneumáticos tipo bolsa como atuadores que apresentava movimentos em um plano.

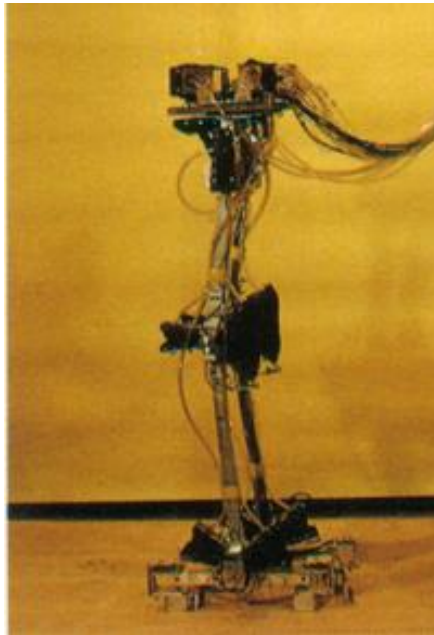


Figura 8. Bípede WAP-2

Em 1971 apresentou-se o “WAP-3”, modelo refinado do WAP-2, capaz de movimentar o centro de massa no plano frontal, subir e descer escadas. O mesmo é mostrado na Figura 9.

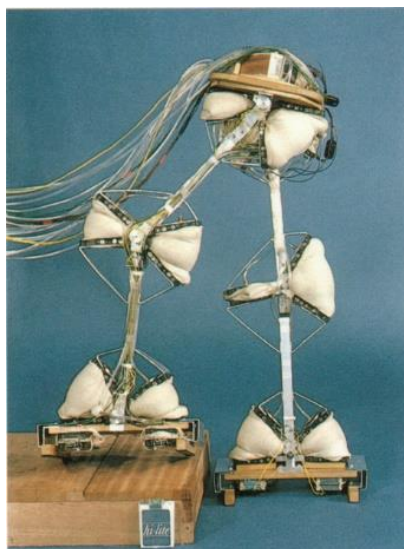


Figura 9. Bípede WAP-3

A Figura 10 apresenta o “WL-5”, desenvolvido em 1972, que tem o corpo lateralmente flexível, o que permite movimentar o centro de massa no plano frontal

para facilitar a caminhada quase-estática, pode mudar a direção, e gasta 45s para cada passo.

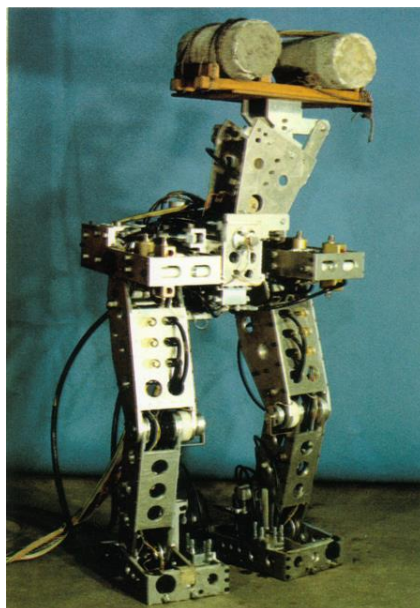


Figura 10. Bípede WL-5

A Figura 11 apresenta o robô antropomórfico inteligente “Wabot 1” (“Waseda robot” em inglês) projetado pela universidade “Waseda” no Japão em 1973 [11], considerado o primeiro robô humanoide no mundo. Ele tem um sistema de visão e de conversa habilitado para se comunicar com uma pessoa, estimado com a inteligência mental de uma criança de um ano e meio de idade; tem mãos com sensores táteis que permitem levar objetos com delicadeza e utiliza as pernas do modelo “WL-5”.

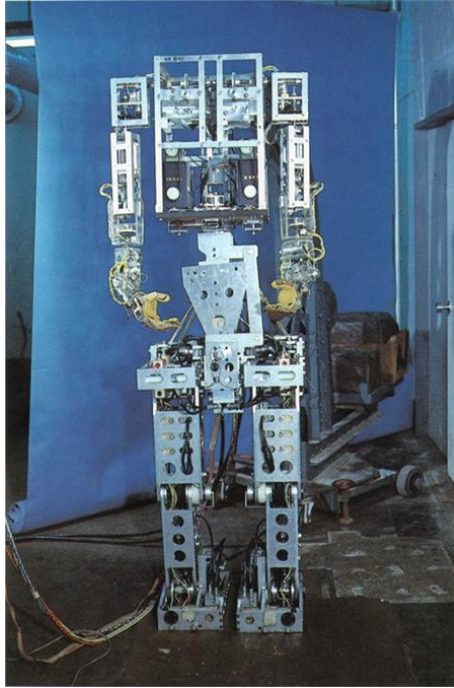


Figura 11. Robô Wabot 1

Em 1984 foi considerado o primeiro sucesso no mundo de caminhar dinâmico completo feito pelo “WL-10RD”. Os sensores de torque nos tornozelos e quadril permitiram um controle flexível de mudança de fase (troca do pé de apoio), vide Figura 12, que apresenta o bípede, que leva 1,3s por passo.

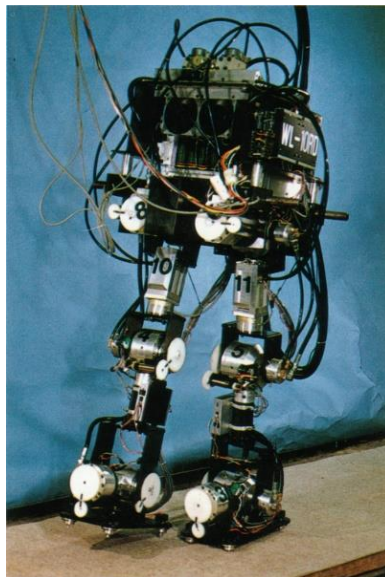


Figura 12. Robô WL-10RD

Em 1986 a “Honda” começou, em alto segredo e silêncio, um projeto para criar um robô humanoide. O primeiro passo da Honda foi o bípede sem tronco “modelo experimental 0” “E0” - (“Experimental model 0” em inglês) [8], antecessor do robô “Asimo”. “E0” levou 20 segundos para completar cada passo quase-estático, só pode andar em linha reta, e com movimentos de duas dimensões. O modelo com altura de 1m, peso de 16,5kg e com 3 motores por perna é mostrado na Figura 13.

A “Honda” continuou em segredo durante 10 anos desenvolvendo os modelos serie “E”, “E0” a “E6”, incluindo o modelo “P1”; até que em 1996 apresentou para o mundo os avanços da pesquisa com o modelo “P2”, que se converteu em uma nova referência e motivação da robótica humanoide. Em 1997 apresentou o “P3”, em 2000 a primeira versão do robô “Asimo”, e nos anos 2002, 2005, 2007e 2011 melhorias foram apresentadas, como mostra a Figura 14. “Asimo” é considerado o robô humanoide mais avançado no mundo. Na seção 1.3, “Principais robôs humanoides”, pode-se ver uma breve descrição do robô.

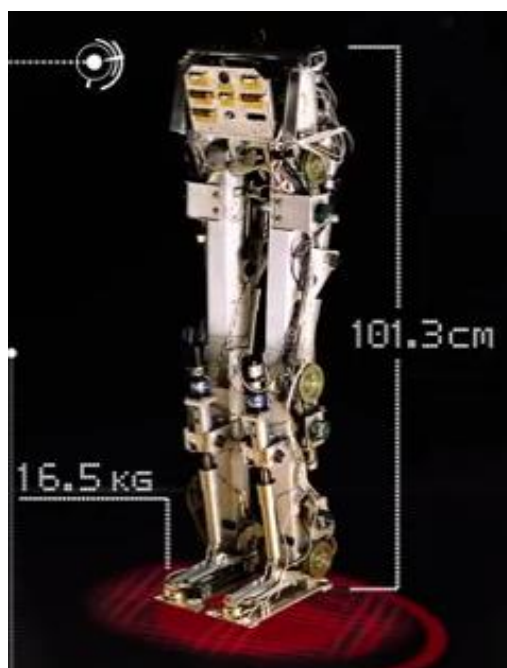


Figura 13. Robô E0

HONDA



ASIMO
<http://www.honda.co.jp/ASIMO/>
<http://www.asimoblog.com/>

© Honda Motor Co., Ltd. and its subsidiaries and affiliates. All Rights Reserved.

Figura 14. Evolução robô Asimo

1.5. Tipos de caminhada e geradores de padrões

Existem dois tipos de caminhada ou marcha, chamados de quase-estática (ou estática) e dinâmica. Na caminhada quase-estática, a projeção do centro de massa nunca abandona o polígono de apoio (definido pela envoltória da projeção vertical dos pontos de apoio) durante o movimento, enquanto na dinâmica existem períodos nos quais a projeção do centro de massa abandona esse polígono. A Figura 15 representa as situações [12].

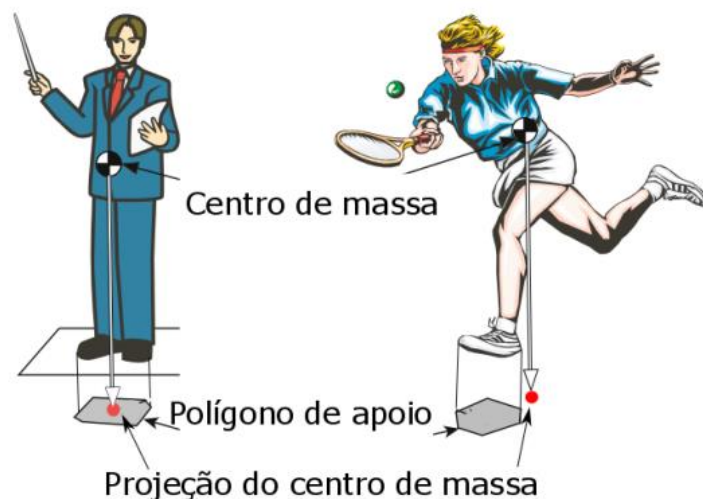


Figura 15. Projeção do centro de massa e polígono de apoio

O conjunto de séries temporais de ângulos das juntas para uma caminhada desejada é chamado *padrão de marcha*, e para criá-lo é utilizado um *gerador de padrões de marcha*, através de, por exemplo: técnicas de captura de movimento [13], interface de usuário gráfica [14], métodos de busca multivariada de alta velocidade [15], aprendizado e computação evolucionária [16], [17] [18] [19] [20] que são descritas brevemente a seguir. O presente documento utiliza algoritmos genéticos dentro da área de computação evolucionária como gerador de padrões de marcha quase-estática, desenvolvida no capítulo 3.

Fases da caminhada dinâmica

A Figura 16 amostra a representação de uma perna humana executando um ciclo completo (100%) de marcha normal. O padrão de marcha é dividido em duas posturas básicas: apoio, que é quando o pé está em contato com o chão; e balanço quando o pé está no ar. Contabilizam aproximadamente 60% e 40% do ciclo respectivamente [21]. A Figura 17 apresenta a interação das posturas dos dois pés da marcha humana.

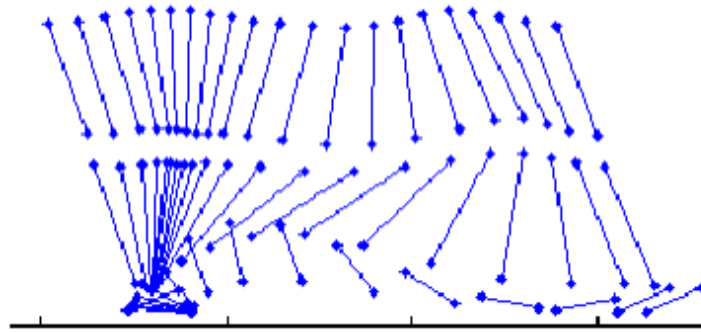


Figura 16. Ciclo de marcha de uma perna humana

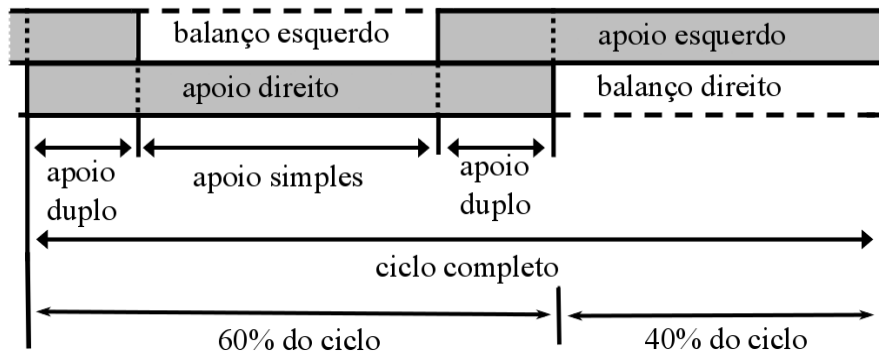


Figura 17. Ciclo de marcha humana

Uma postura intermediária é a de apoio duplo, que é quando os dois pés estão em contato com o chão, e contabiliza aproximadamente 20% do ciclo.

Além disso, cada uma das posturas básicas está sub-dividida da seguinte maneira: Apoio é dividida em cinco (5) intervalos, contato do calcanhar, apoio da planta do pé, meio do apoio, elevação do calcanhar e decolar do pé; e Balanço é dividido em três (3): aceleração, meio do balanço e desaceleração [22]

Fases da caminhada quase-estática

O caminhar é dividido em duas fases consecutivas para completar um passo [23], e repetidas para n passos.

A fase 1 ou de suporte simples é a etapa na qual o robô está apoiado no chão com um pé, e movimenta o outro pelo ar para avançar um passo, devendo-se manter o centro de massa dentro do espaço definido pelo pé ou polígono de apoio.

Na fase 2 ou de suporte duplo, o robô está com ambos os pés no chão, devendo-se transladar o centro de massa no espaço abrangido pelo pé que anteriormente estava no ar. A Figura 18 mostra a projeção vertical do centro de massa no chão das fases de caminhada quase-estática, representando o suporte simples Figura 18a), suporte duplo Figura 18b) e uma queda Figura 18c) [23].

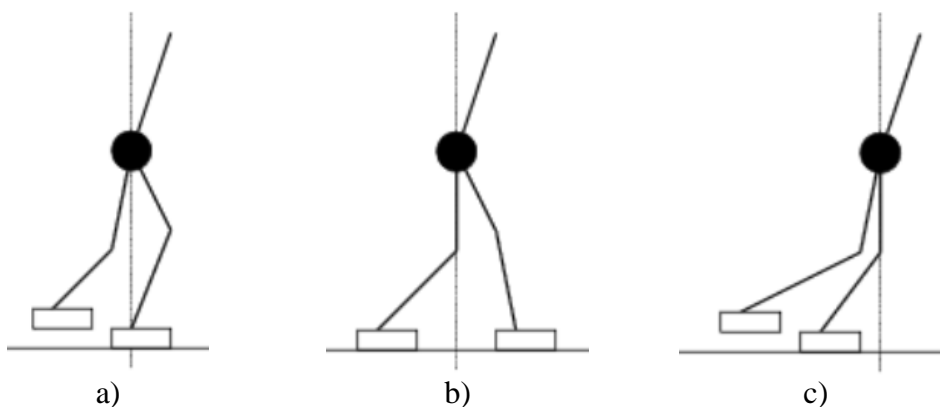


Figura 18. Caminhada quase-estática

Geradores de padrões

Captura de movimento [13]

Utiliza um sistema que pode gravar a trajetória no espaço de pontos específicos marcados no corpo de um ser vivo, como mostra a Figura 19. A informação capturada é digitalizada e armazenada no computador.



Figura 19. Captura de movimento

Interface de usuário gráfica [14]

Um método como o chamado de interface “pin/drag” permite fixar alguns pontos e movimentar outros dentro de uma interface gráfica, como apresenta a Figura 20. O kit “Bioid GP” utilizado neste trabalho também permite programar os movimentos do robô, utilizando o ambiente gráfico mostrado na Figura 21 [24].

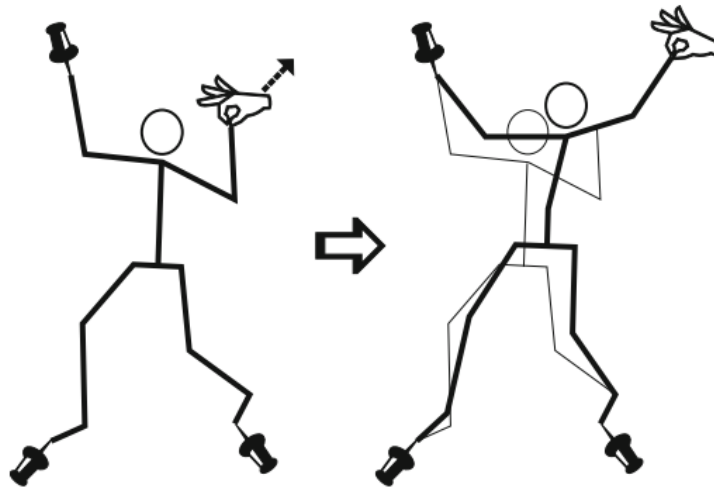


Figura 20. Interface de usuário pin/drag

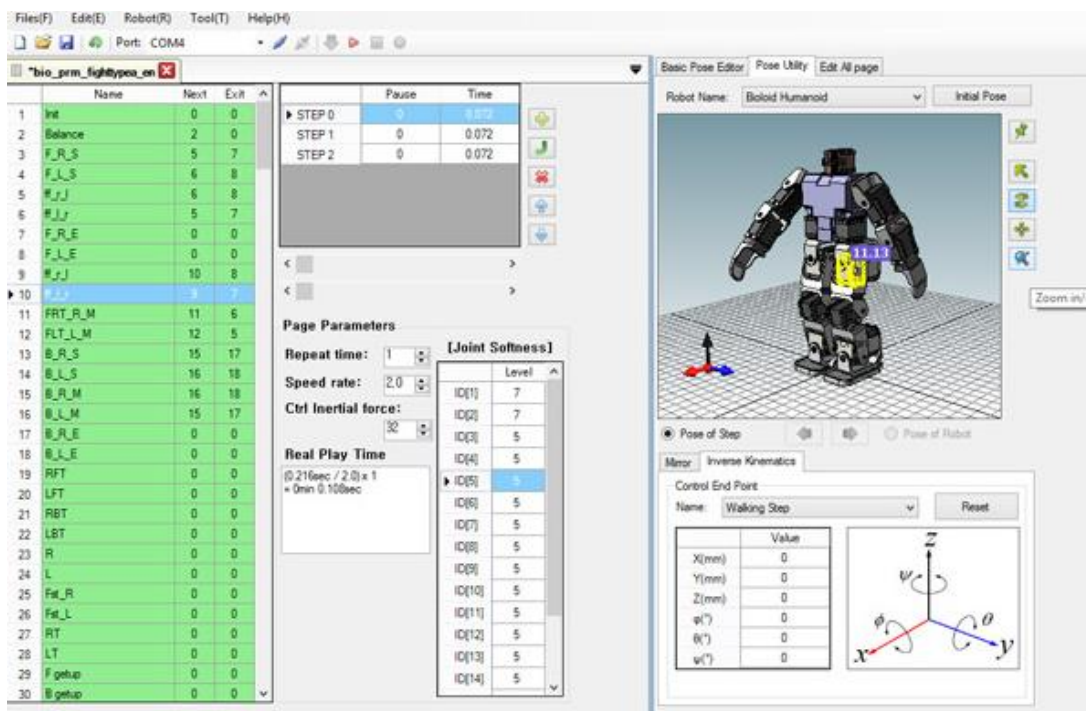


Figura 21. Interface de usuário Bioid GP

Métodos de busca multivariada de alta velocidade [15]

É um método como por exemplo a busca rápida de conexão de árvores exploratórias aleatórias, ou “RRT-Connect, Rapidly-exploring Random Trees – Connect, em inglês”. Utilizado normalmente na cinemática inversa quando se tem obstáculos. No espaço de configuração, cria pontos que representam o tronco de árvores, e suas raízes ao serem conectadas com as de outra árvore formam um caminho que os comunica. Este caminho é uma resposta de configurações que satisfazem os requerimentos, como apresenta a Figura 22, com o crescimento das raízes de duas árvores em direção à outra evitando colisão com os obstáculos.

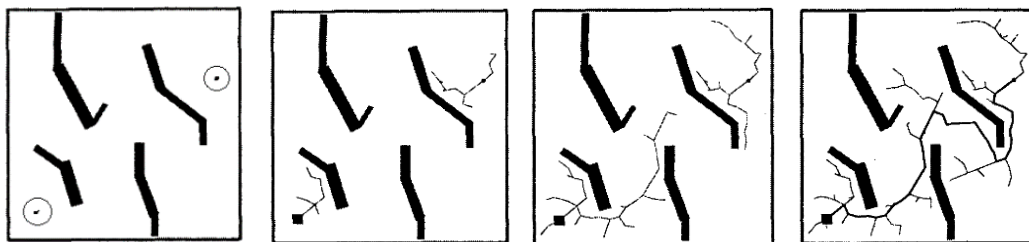


Figura 22. RRT-Connect

Aprendizado e computação evolucionária [16], [17], [18], [19], [20].

Utiliza algoritmos genéticos dentro da computação evolucionária como gerador de padrões de marcha quase-estática, foco principal do presente documento, desenvolvida no Capítulo 3.

1.6. Estrutura da dissertação

Esta dissertação está organizada em 6 capítulos. No Capítulo 2 faz-se o desenvolvimento da modelagem cinemática do robô bípede estudado. O Capítulo 3 descreve e apresenta o uso dos algoritmos genéticos na geração dos padrões de marcha. No Capítulo 4 relatam-se os resultados obtidos. O Capítulo 5 apresenta as conclusões e as propostas de trabalhos futuros. E, finalmente, o Capítulo 6 apresenta as referências bibliográficas.

2 Robô Bípede Plano

Robô bípede plano é aquele que se desloca sobre duas pernas com movimentos em um só plano. O sistema tratado na presente dissertação não é humanoide, ou seja, não tem braços nem cabeça, consistindo apenas de pernas e quadris.

2.1. Representação do robô bípede físico em simulação

A Figura 23 apresenta o robô bípede físico montado e o desenho feito no programa “Matlab” que o representa nas simulações. As Figura 23a) e Figura 23b) relacionam cada elo (*link*) do robô com o nome das partes equivalentes do corpo humano, para maior familiaridade; e as Figura 23c) e Figura 23d) relacionam os motores e os centros de massa dos elos. O robô tem seis (6) motores, três (3) para cada perna, localizados no tornozelo, joelho e quadril. Cada elo tem a posição de seu centro de massa (cm) calculada segundo o software “Solidworks”, como mostrado no Apêndice A.

O centro de massa geral do robô bípede é calculado com a equação (1), na qual o vetor de posição do centro de massa é igual ao somatório do produto da massa do *i*-ésimo elo com seu vetor posição, dividido pela massa total.

$$r_{cm} = \frac{\sum_i m_i r_i}{\sum_i m_i} \quad (1)$$

onde:

r_{cm} : centro de massa total do robô bípede.

m_i : massa do *i*-ésimo elo.

r_i : vetor posição *i*-ésimo elo.

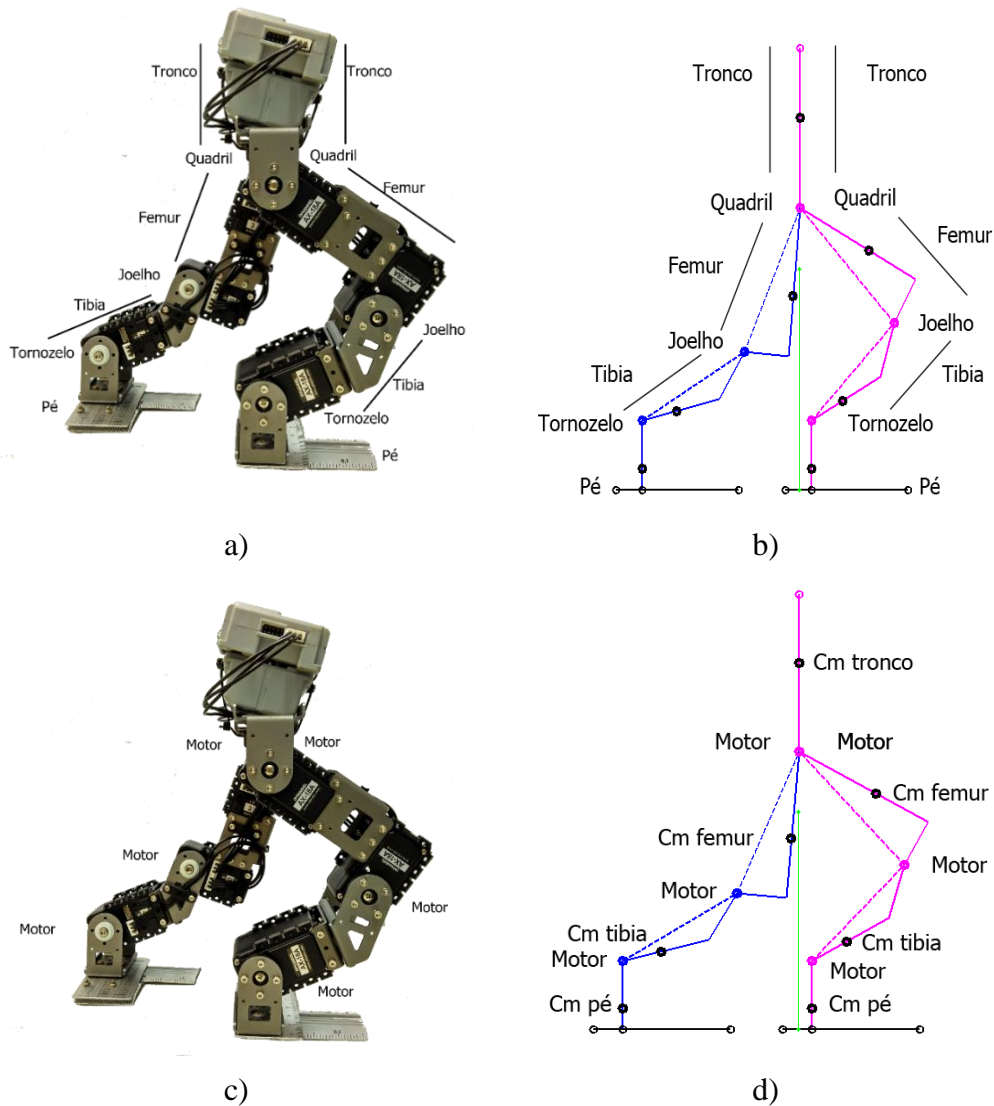


Figura 23. Representação do robô bípede físico em simulação

O ângulo do tronco em relação à horizontal é independente do ângulo entre os fêmures das pernas, e é uma combinação entre os dois motores no quadril. Os pés estão restritos neste trabalho a se conservarem de forma horizontal (o que poderia se converter em uma limitação em futuros trabalhos ao observar a seção 1.5 Tipos de caminhada e geradores de padrões), uma vez que o ambiente considerado é plano.

A Figura 24 apresenta as duas etapas com desenhos de Solidworks e Matlab, na parte esquerda e direita, respectivamente. As Figura 24a) e Figura 24b) se encontram na fase 1 enquanto as Figura 24c) e Figura 24d) estão na fase 2.

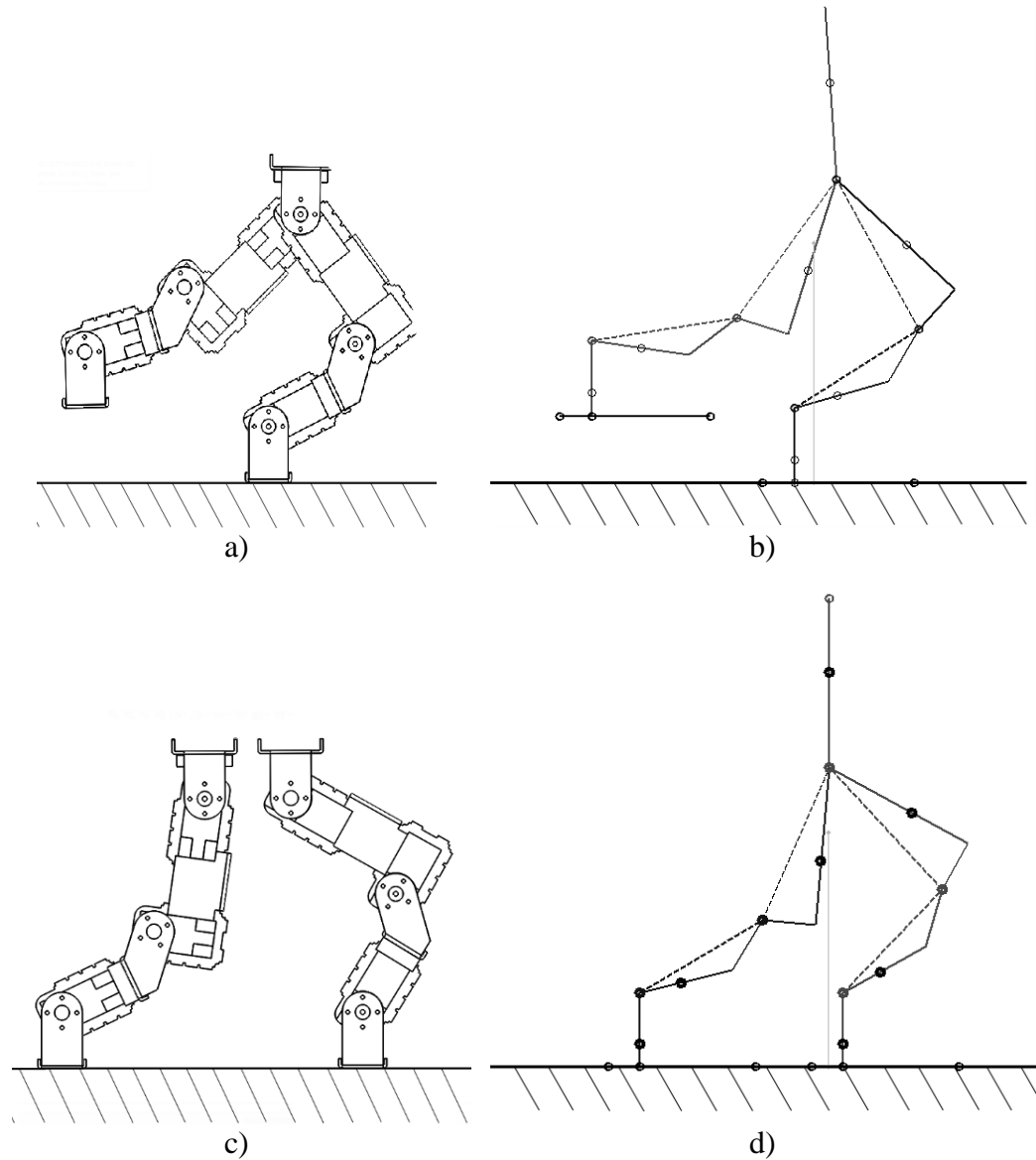


Figura 24. Desenhos de Solidworks e Matlab

2.2. Definição dos ângulos envolvidos

Ângulos simplificados α

Por facilidade de cálculo computacional, a ser mencionado na Seção 3.4, é importante diminuir o número de variáveis envolvidas na representação das soluções de marcha. Por esse motivo, tendo em conta que o ângulo do tronco do robô é independente do ângulo entre os fêmures das pernas, e que os pés estão restritos a serem horizontais, aproveitou-se que em duas dimensões tem-se cinco

pontos chaves, os quais ao serem unidos serialmente com linhas retas permitem a representação do robô como um manipulador serial de quatro (4) graus de liberdade, como é apresentado na Figura 25. Os pontos são: tornozelo do pé que está no chão; joelho da perna de apoio; quadril; joelho da perna no ar; e tornozelo do pé no ar, tendo motores em todos os pontos com exceção do último. Os ângulos dos atuadores são α_1 , α_2 , α_3 e α_4 , respectivamente.

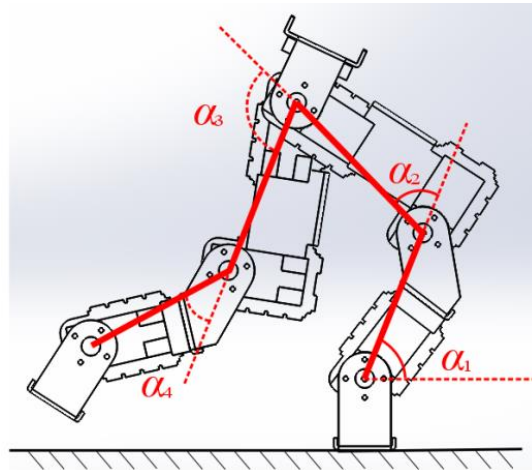


Figura 25. Ângulos α fase 1

A Figura 26 para a fase 2 do passo apresenta os pontos no tornozelo do pé que está no chão, no joelho da perna de apoio, e no quadril, com ângulos dos motores iguais α_1 e α_2 .

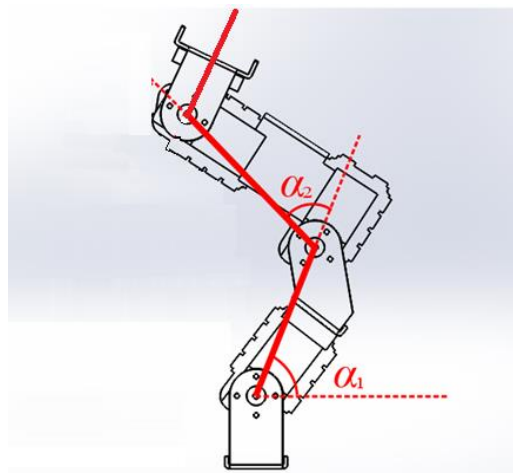


Figura 26. Ângulos α fase 2

O uso destes ângulos α é uma forma simplificada de representar a configuração do robô. Em configurações de perna quase esticada, eles dificultam a interpretação da simulação e não permitem uma fácil exibição da situação física, como mostra a Figura 27a). Nessa figura, segundo os movimentos naturais de uma perna humana, o robô estaria se movimentado da direita para esquerda, quando na verdade, como apresenta a Figura 27b), o deslocamento é reverso, ou com ângulos não permitidos para uma pessoa. Com isso decidiu-se parametrizar o modelo das simulações com mais variáveis, para obter melhores visualizações e interpretações.

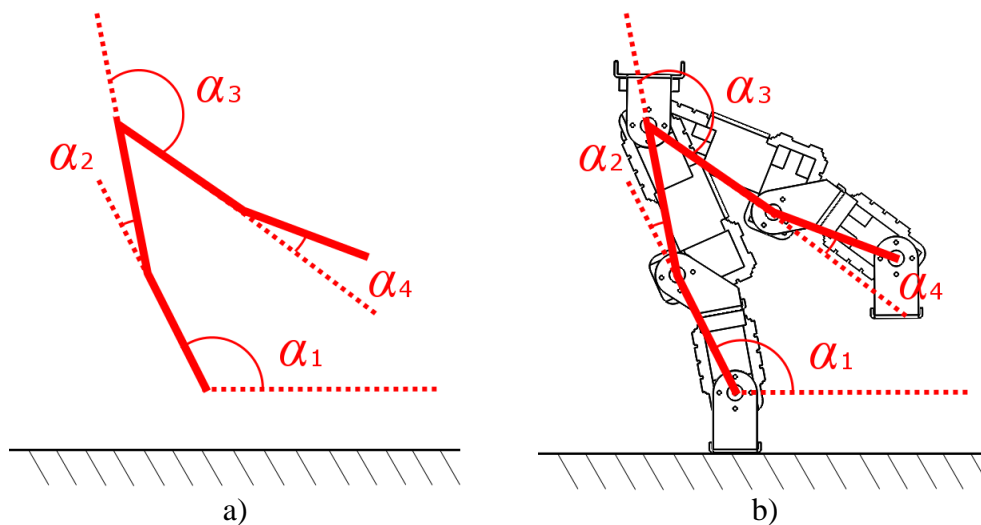


Figura 27. Ângulos α , visualização

Este cálculo computacional forneceu de modo simplificado as quatro variáveis de ângulo, mas as mesmas têm que ser transformadas em seis variáveis novamente que representam os motores físicos. Por esta razão determinaram-se ângulos de transição entre ângulos α e os ângulos reais dos motores.

Ângulos de transição θ_{qm}

Os ângulos de transição θ_{iqm} (θ dos quatro motores) mostrados na Figura 28 são ângulos intermediários que permitem transformar ângulos α em ângulos reais dos motores físicos e vice-versa. A equação (2) relaciona α com θ_{qm} .

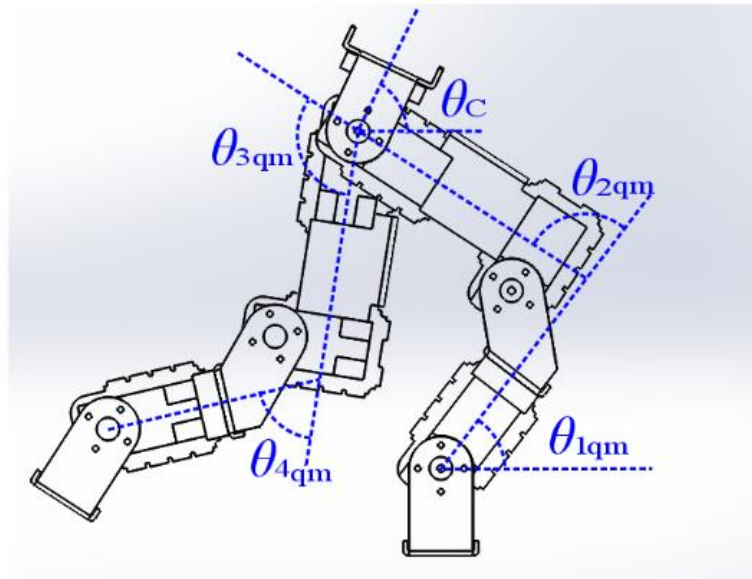


Figura 28. Ângulos de transição

$$\begin{aligned}
 \theta_{1qm} &= \alpha_1 - 17^\circ \\
 \theta_{2qm} &= \alpha_2 + 35.42^\circ \\
 \theta_{3qm} &= \alpha_3 \\
 \theta_{4qm} &= \alpha_4 - 35.42^\circ
 \end{aligned}
 \tag{2}$$

onde:

θ_c : ângulo do tronco em relação à horizontal, fixado em 90° .

θ_{qm} : ângulo de transição (θ quatro motores).

Ângulos motores físicos θ_{rf}

A Figura 29 apresenta os ângulos reais dos motores, que serão usados para seus controles. A equação (3) expressa a relação entre esses ângulos e os ângulos de transição (ou intermediários).

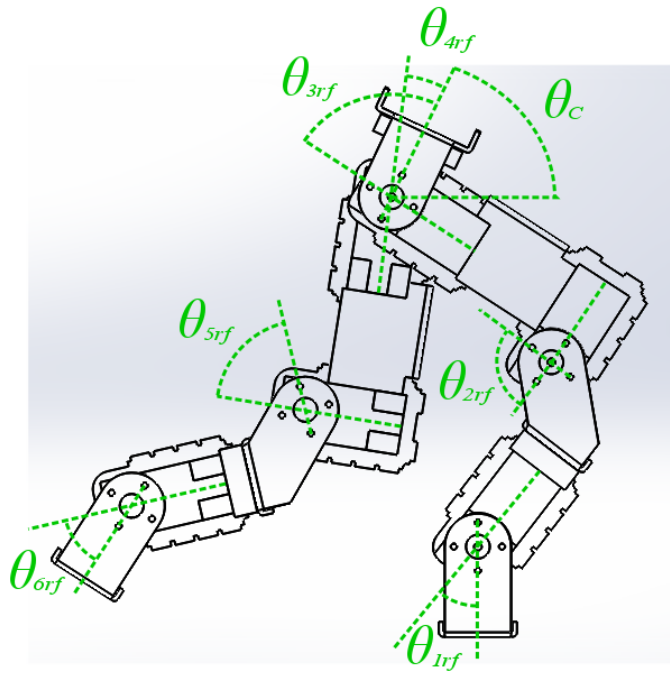


Figura 29. Ângulos dos motores físicos

$$\begin{aligned}
 \theta_{1rf} &= 90 - \theta_{1qm} \\
 \theta_{2rf} &= \theta_{2qm} \\
 \theta_{3rf} &= \theta_C - \theta_{1qm} - \theta_{2qm} \\
 \theta_{4rf} &= -\theta_C - 180 + \theta_{1qm} + \theta_{2qm} + \theta_{3qm} \\
 \theta_{5rf} &= -\theta_{4qm} \\
 \theta_{6rf} &= -270 + \theta_{1qm} + \theta_{2qm} + \theta_{3qm} + \theta_{4qm}
 \end{aligned} \tag{3}$$

Utilizando os ângulos θ_{irf} na seção 2.3, é possível representar uma simulação bem mais aproximada da situação real.

Ângulos de programação

Os motores utilizados têm uma configuração pré-definida de ângulos e posições equivalentes para serem programados no sistema decimal ou hexadecimal, como apresenta a Figura 30a) [25]. Estes ângulos foram modificados, para continuar a nomenclatura normalmente utilizada na literatura como mostra a Figura 30b). O menor grau (0°) passa a ser -150° , equivalente a zero (0) em decimal ou hexadecimal, como valor de entrada na programação; e o máximo grau (300°) passa a ser $+150^\circ$, equivalente a 1024 em decimal ou 3FF hexadecimal para o programa.

A relação do motor posição decimal e ângulo adaptado é uma relação linear que está expressa na equação (4).

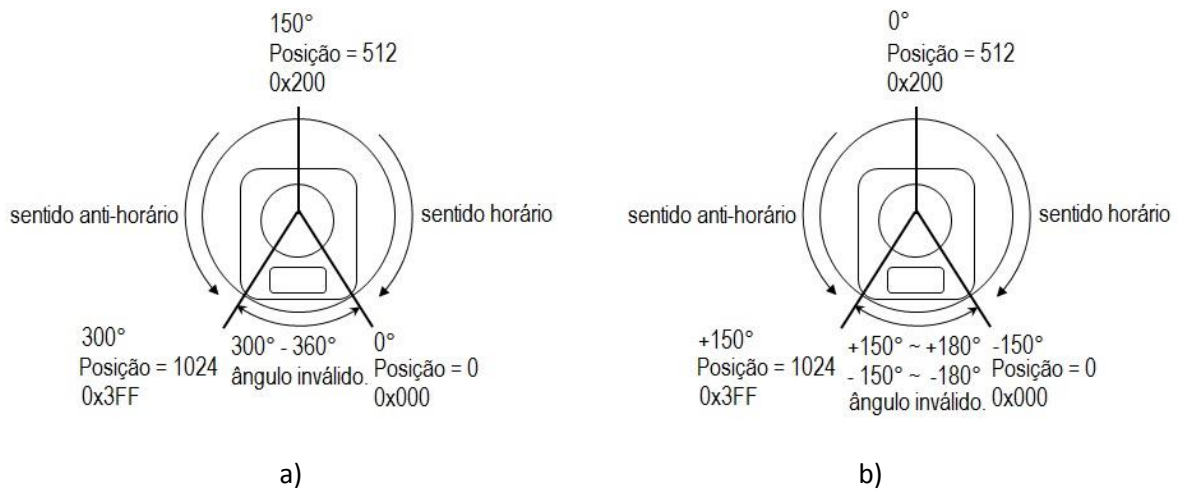


Figura 30. Ângulos de programação

$$P_D = 3.41\theta_{rf} + 511.5$$

$$\theta_{rf} = \frac{P_D - 511.5}{3.41} \quad (4)$$

onde:

P_D : posição representada em notação decimal

θ_{rf} : ângulo θ robô físico

2.3. Cinemática, parâmetros de Denavit-Hartenberg

A Figura 31 apresenta a relação entre dois sistemas coordenados adjacentes, usando a notação de Denavit-Hartenberg (DH) [26]. Note que os ângulos α_i dos parâmetros DH não têm relação com os “ângulos simplificados α ” utilizados para a representação do robô bípede plano, nem no cálculo dos algoritmos genéticos nos Capítulos 2 e 3.

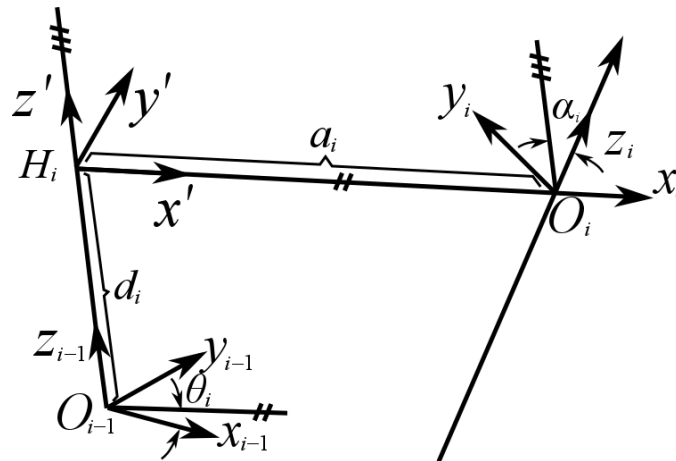


Figura 31. Sistemas coordenados adjacentes com a notação DH

onde:

d_i é a distância no eixo Z entre o sistema O_{i-1} e O_i

θ_i é o ângulo entre os eixos X_{i-1} e X_i

a_i é a distância no eixo X entre o sistema O_{i-1} e O_i

α_i é o ângulo entre os eixos Z_{i-1} e Z_i

A matriz de relação de orientação e posição entre os dois sistemas coordenados (equação (5)) é obtida após aplicar uma rotação no eixo Z de um ângulo θ_i , uma translação no eixo Z de uma distância d_i , uma rotação no eixo X de um ângulo α_i e uma translação no eixo X de uma distância a_i :

$$A_i^{i-1} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

Os três primeiros vetores coluna (3x1) contêm a direção dos cossenos diretores do sistema coordenado O_i , e o último vetor coluna (3x1) representa a posição da origem O_i .

A posição e orientação do sistema coordenado do elo i relativo ao elo $i-1$ é descrito como função de q_i (configuração dos ângulos dos elos ou diferentes fotogramas). Considerando as n transformações consecutivas ao longo dos elos seriais, a posição e orientação do último sistema coordenado em relação ao sistema

coordenado da base (sistema zero) se representa na Figura 32 e é calculado com a equação (6):

$$T = A_1^0(q_1)A_2^1(q_2)\cdots A_n^{n-1}(q_n) \quad (6)$$

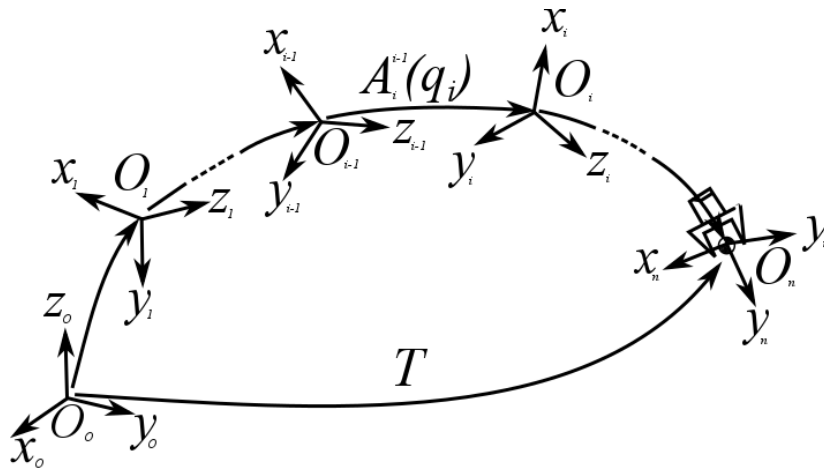


Figura 32. Matriz de posição orientação

O robô bípede foi desenhado como uma série de 12 elos retos. Os parâmetros DH para a cinemática direta do desenho dos elos são mostrados na Tabela 1.

Para achar os pontos que representam os centros de massa, toma-se o sistema de referência imediatamente anterior e aplicam-se as respectivas rotações e translações para obter o novo sistema T para o centro de massa com a equação (7). Os parâmetros para cada centro de massa são mostrados na Tabela 2.

$$T_{cmj}^0 = A_i^0(q_{n-1})A_{cmj}^i(q_n) \quad (7)$$

	d_i	θ_i	a_i	α_i	A_i^{i-1}
1	0	90°	L_p	0	A_1^0
2	0	$-\theta_{1rf}$	L_{1a}	0	A_2^1
3	0	45°	L_{1b}	0	A_3^2
4	0	$-135^\circ - \theta_{2rf}$	L_{2a}	0	A_4^3
5	0	90°	L_{2b}	0	A_5^4
6	0	θ_{3rf}	L_t	0	A_6^5
7	0	180°	L_t	0	A_7^6
8	0	θ_{4rf}	L_{3a}	0	A_8^7
9	0	-90°	L_{3b}	0	A_9^8
10	0	$135^\circ - \theta_{5rf}$	L_{4a}	0	A_{10}^9
11	0	-45°	L_{4b}	0	A_{11}^{10}
12	0	$-\theta_{6rf}$	L_p	0	A_{12}^{11}

Tabela 1. Parâmetros DH das origens dos sistemas coordenados

	d_i	θ_i	a_i	α_i	A_{cmj}^i
cm1	0	90°	L_{cm1}	0	A_{cm1}^0
cm2	0	$-\theta_{1rf}$	L_{cm2}	0	A_{cm2}^1
cm3	0	90°	L_{cm3}	0	A_{cm3}^4
cm4	0	θ_{3rf}	L_{cm4}	0	A_{cm4}^5
cm5	0	θ_{4rf}	L_{cm5}	0	A_{cm5}^7
cm6	0	-45°	L_{cm6}	0	A_{cm6}^{10}
cm7	0	$-\theta_{6rf}$	L_{cm7}	0	A_{cm7}^{11}

Tabela 2. Parâmetros DH dos centros de massa

Os pontos origens dos sistemas coordenados das duas tabelas anteriores são mostrados na Figura 33a), as distâncias a_i estão representadas na Figura 33b) e Figura 33c), e os ângulos θ_{irf} se mostram na Figura 33d).

A forma simplificada inicial de visualização da simulação foi utilizando quatro (4) elos e os quatro (4) ângulos α . Os elos são as linhas pontilhadas mostradas na Figura 33a). A cinemática direta simplificada apresenta a posição da coordenada

extrema (X_{ext} , Y_{ext}) mostrada na Figura 34 pela equação (8), a posição do quadril é mostrada pela equação (9), e a matriz “Jacobiana” que relaciona as velocidades dos elos do pé no chão com a velocidade do quadril está expressa na equação (10).

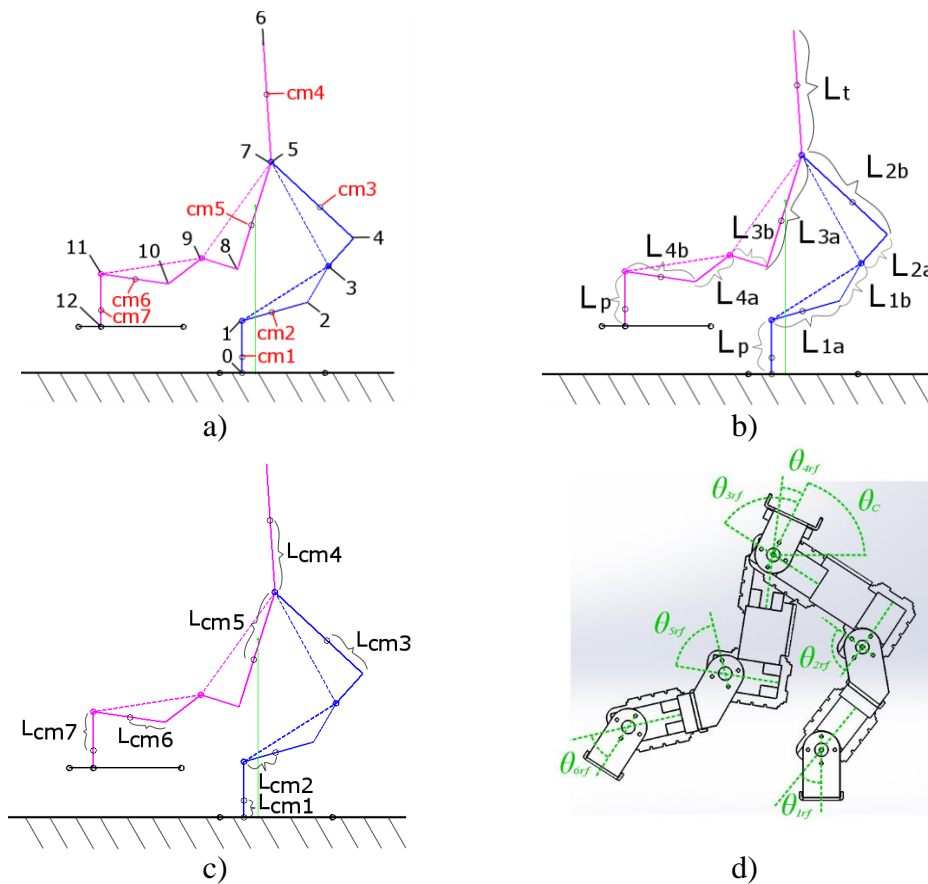


Figura 33. Parâmetros DH na representação do bípede

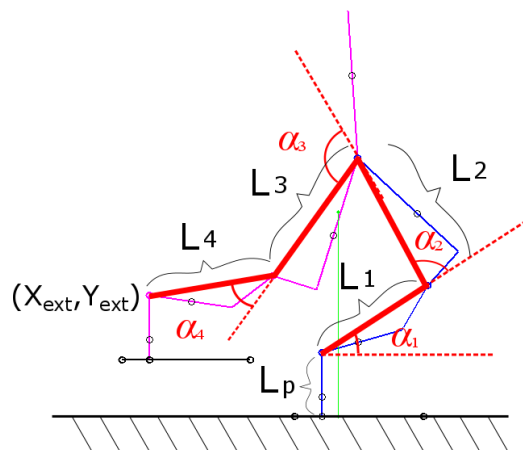


Figura 34. Cinemática direta simplificada

$$\begin{aligned} X_{ext} &= L_1 \cos \alpha_1 + L_2 \cos(\alpha_1 + \alpha_2) + L_3 \cos(\alpha_1 + \alpha_2 + \alpha_3) + L_4 \cos(\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4) \\ Y_{ext} &= L_1 \sin \alpha_1 + L_2 \sin(\alpha_1 + \alpha_2) + L_3 \sin(\alpha_1 + \alpha_2 + \alpha_3) + L_4 \sin(\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4) \end{aligned} \quad (8)$$

$$\begin{aligned} X_{quadril} &= L_1 \cos \alpha_1 + L_2 \cos(\alpha_1 + \alpha_2) \\ Y_{quadril} &= L_1 \sin \alpha_1 + L_2 \sin(\alpha_1 + \alpha_2) \end{aligned} \quad (9)$$

$$J_{quadril} = \begin{bmatrix} -L_1 \sin \alpha_1 - L_2 \sin(\alpha_1 + \alpha_2) & -L_2 \sin(\alpha_1 + \alpha_2) \\ L_1 \cos \alpha_1 + L_2 \cos(\alpha_1 + \alpha_2) & L_2 \cos(\alpha_1 + \alpha_2) \end{bmatrix} \quad (10)$$

O cálculo da matriz Jacobiana do sistema corresponde à equação (11)

$$J_{quadril} = \begin{bmatrix} \frac{\delta X_{quadril}}{\delta \alpha_1} & \frac{\delta X_{quadril}}{\delta \alpha_2} \\ \frac{\delta Y_{quadril}}{\delta \alpha_1} & \frac{\delta Y_{quadril}}{\delta \alpha_2} \end{bmatrix} \quad (11)$$

2.4. Estática

A modelagem do robô bípede tratado nesta dissertação é feita considerando uma situação quase-estática, ou seja, se é parado o movimento, a estrutura deve-se manter equilibrada para a configuração de ângulos fornecidos pelo gerador de padrões de marcha (vide Seção 3.4 “ Configuração do Cromossomo.”), pois o centro de massa deverá estar dentro do polígono de apoio (vide Seção 1.5 “Tipos de caminhada e geradores de padrões”) em todo o momento.

Os torques dos motores calculados são os produzidos pelo próprio peso de cada elo devido à ação da gravidade, chamados de termos gravitacionais, mais os momentos causados por uma força externa que representa o peso da parte superior do corpo no robô bípede. Ela é aplicada no quadril em direção vertical, já que se

fixou em 90° o ângulo do corpo, e considera-se que o mesmo não produz momento externo no quadril.

As forças são mostradas na Figura 35, e os termos gravitacionais e os torques da força no quadril são calculados com as equações (12) e (13). Para a fase de apoio duplo, o peso é dividido igualmente entre as duas pernas, representadas como dois manipuladores seriais independentes “conectados” no extremo.

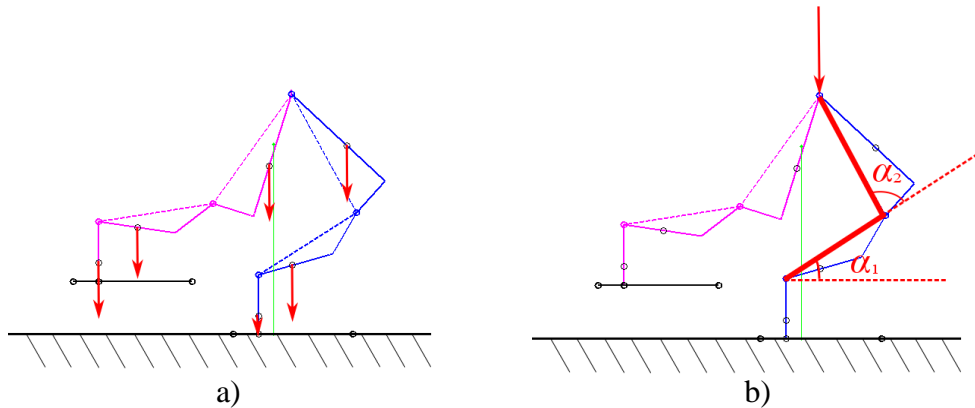


Figura 35. Equilíbrio estático

$$G_i = \sum_{j=1}^n m_j * g^T * J_{Li}^{(j)} \quad (12)$$

onde:

G: torque do motor produzido pela gravidade

m: massa do elo

g: gravidade

J: matriz Jacobiana do centro de massa dos elos, dada por

$$\tau = J^T * F \quad (13)$$

onde:

τ : torque produzido pela força externa

J: matriz jacobiana do ponto onde é aplicada a força (equação (10))

F: força externa aplicada

2.5. Controle Cinemático

Na presente dissertação utilizou-se o robô bípede físico apenas para reproduzir o padrão de marcha fornecido, em malha aberta. Portanto, durante o desenvolvimento não é implementado controle no *hardware* com *feedback* (malha fechada), por não ser um dos objetivos deste trabalho. De qualquer forma, cada um dos motores tem controlador PID de posição e velocidade próprios integrados, confiando-se na atuação de cada um deles nos comandos em malha aberta.

Para as simulações do controle cinemático, importaram-se os desenhos tridimensionais do robô bípede físico do “Solidworks” até o “Simulink” do ambiente Matlab, utilizando a biblioteca “SimMechanics” primeira geração, como mostra a Figura 36 para a fase 1. A representação em diagrama de blocos gerada é mostrada na Figura 37, na qual em cada bloco de junta adicionou-se um atuador ideal que permite o movimento, e um sensor, que têm como entrada e saída a posição. Ainda na Figura 37, AMB é o ambiente de simulação; origem é a origem do sistema de coordenadas inercial localizado na base do pé (para a fase 1 se localiza no pé em contato com o chão, e para a fase 2, que se trata como dois manipuladores independentes, está na base de cada perna); união é o contato que fixa uma peça com outra; peça é a representação da peça, neste caso em 3D; e finalmente, junta é a articulação entre duas peças, neste caso rotativa.

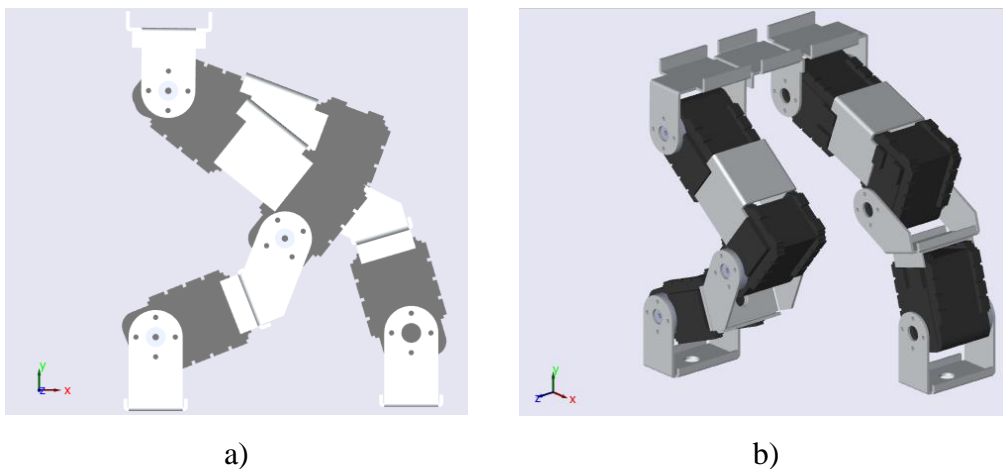


Figura 36. Desenho 3D importado do “SolidWorks” para “Simulink”

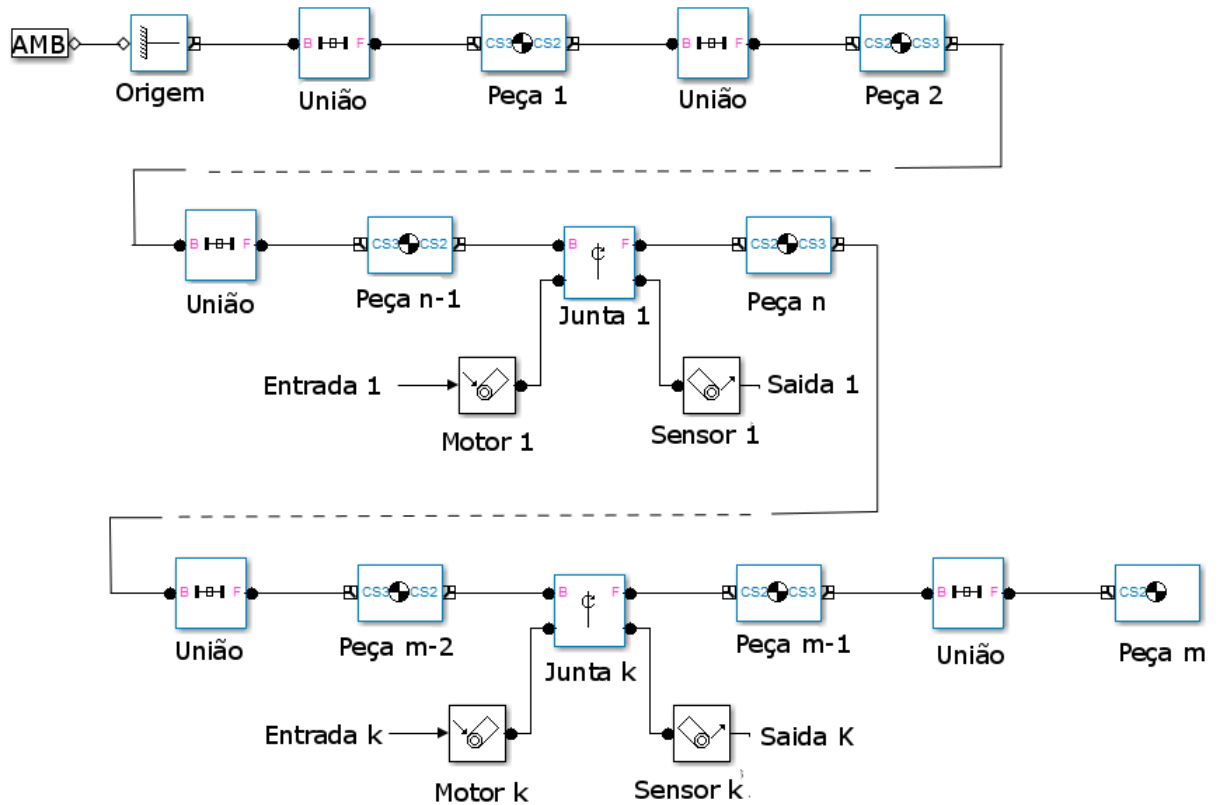


Figura 37. Sistema no “SimMechanics” primeira geração

Transição suave entre fotografamas

Os valores de referência de entradas desejadas foram tratados como uma transição suave da posição entre cada fotografama. Cada metade do intervalo foi traçada segundo equações gerais de movimento uniformemente acelerado, como apresenta a Figura 38 e as equações (14) a (17).

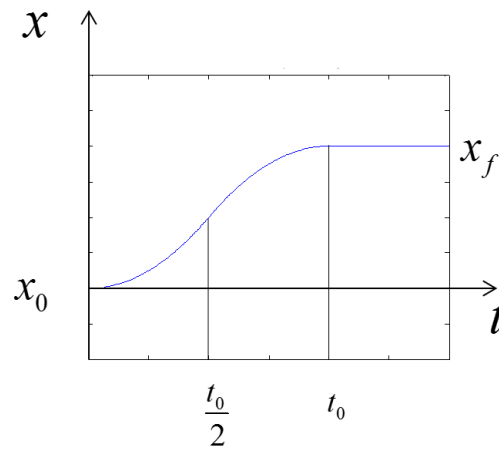


Figura 38. Transição suave entre fotogramas

Do gráfico anterior

$$\Delta x = x_f - x_0 \quad (14)$$

para $x_0 \leq x \leq x_0 + \Delta x/2$

Têm-se que $v_0 = 0$ e $a = a_0 = \text{constante}$, logo em $t = t_0/2$

$$x_0 + \frac{\Delta x}{2} = x_0 + \frac{1}{2} a_0 \left(\frac{t_0}{2}\right)^2 \quad (15)$$

Isolando a_0

$$a_0 = \frac{4\Delta x}{t_0^2} \quad (16)$$

A velocidade é dada por

$$v_{\frac{t_0}{2}} = 0 + a_0 \frac{t_0}{2} \quad (17)$$

no intervalo $x_0 + \Delta x/2 \leq x \leq x_0 + \Delta x$. Desenvolve-se de igual maneira que $v_0 = a_0(t_0/2)$ e $a = -a_0$

As saídas são avaliadas com o somatório do erro quadrático, como mostra a equação (28)

$$Aval = \sum^n (Desejado - Obtido)^2 \quad (18)$$

Apresenta-se o modelamento do controle cinemático desde dois pontos de vista: no primeiro, onde é utilizado o bloco “Pidcontroller”, o qual faz uma identificação da planta e auto ajuste dos parâmetros do controlador PID¹ para cada um dos seis motores, enquanto que a cinemática está envolvida diretamente com a representação das peças em 3D; e para o segundo, os parâmetros do controlador são estimados com algoritmos genéticos, e a cinemática calculada analiticamente para controle da posição do quadril e orientação do tronco utilizando três atuadores.

Função de transferência simplificada do motor DC

Determinou-se a planta do motor DC como uma versão simplificada de primeira ordem para saída de velocidade angular (29). Para obter a posição θ , adiciona-se um integrador, ou seja, multiplica o denominador por S , passando a ter segunda ordem [27], [28], [29]. Note que não está dentro dos objetivos do presente trabalho a identificação da planta dos motores.

$$G(S) = \frac{\omega(S)}{V(S)} = \frac{K_\tau}{RJS + K_\tau^2} \quad (19)$$

$$G(S) = \frac{\theta(S)}{V(S)} = \frac{K_\tau}{S(RJS + K_\tau^2)} \quad (20)$$

onde:

S: variável de Laplace

K_τ : constante de torque

R: resistência elétrica

J: inércia incluindo a redução

ω : velocidade angular

θ : posição angular

¹ A identificação da planta e feita em segundo plano e não é apresentado o resultado.

A inércia J estabeleceu-se em $1,35 \times 10^{-4} \text{ kg.m}^2$, resistência elétrica foi definida em $6,3 \Omega$, e a constante de torque K_τ definida em $0,8182 \text{ Nm/A}$ utilizando a corrente de “*stall torque*” fornecida como mostra a equação (25). Não foram feitos experimentos para obter os valores, utilizou-se o documento de dados do fabricante e parte de uma caracterização feita por Mensink do motor “Dynamixel AX-12” [30], [25].

$$K_\tau = \frac{\tau_{ST}}{i_{ST}} \quad (21)$$

onde:

τ_{ST} : torque de “stall torque”

i_{ST} : corrente elétrica de “stall torque”

Controlador PID “autotuning”

A malha de controle geral utilizada, que simula o controle individual de cada motor do robô bípede físico, é apresentada na Figura 39. A cinemática está envolvida diretamente com a representação das peças em 3D, como mostra a Figura 40 no ambiente “Simulink”

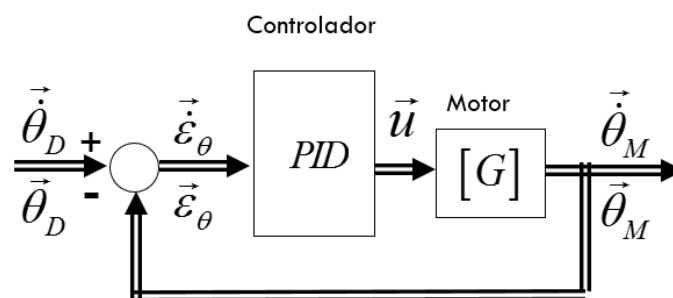


Figura 39. Malha de controle com bloco PID auto-ajustável

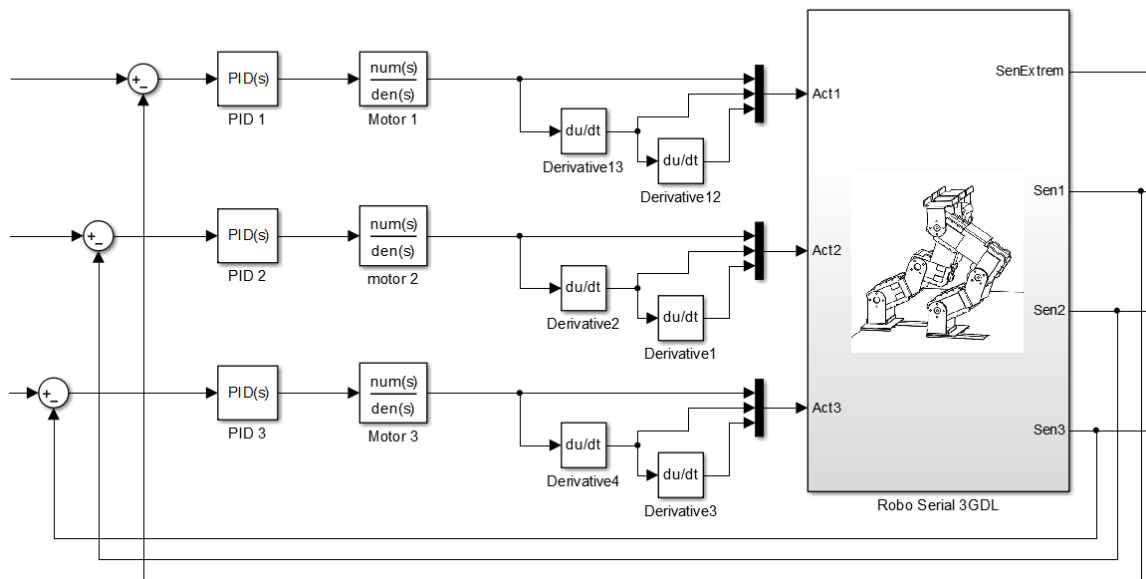


Figura 40. Malha de controle com bloco PID Simulink

O bloco de controle utilizado é o “Pid controller” com a opção de auto-ajustar os valores do controlador. O ajuste do PID é o processo de encontrar os valores dos ganhos proporcional, integral e derivativo para atingir requerimentos de projeto desejados. O ajuste automático do bloco “Pid controller” envolve: a identificação e linearização do modelo da planta com testes de entrada saída, e teste de controladores em modo “batch” ou segundo plano. O controlador PID inicial computado pelo bloco de “Matlab” possui desempenho e robustez² equilibrados, que então permitem ajustes do usuário [31].

A equação (22) é utilizada para a sintonização automática, e inclui um filtro derivativo, o qual tem a função de diminuir os ruídos de alta frequência, para que não sejam amplificados pelo termo derivativo [32].

$$PID = P + I \frac{1}{S} + D \frac{N}{1 + N \frac{1}{S}} \quad (22)$$

² Vide apêndice E

onde:

P: constante proporcional

I: constante integral

D: constante derivativa

N: constante de filtro derivativo (valor por *default* igual a 100)

A Figura 41 mostra a interface de ajuste que permite ao usuário modificar os ganhos, após serem fornecidos os valores automaticamente.

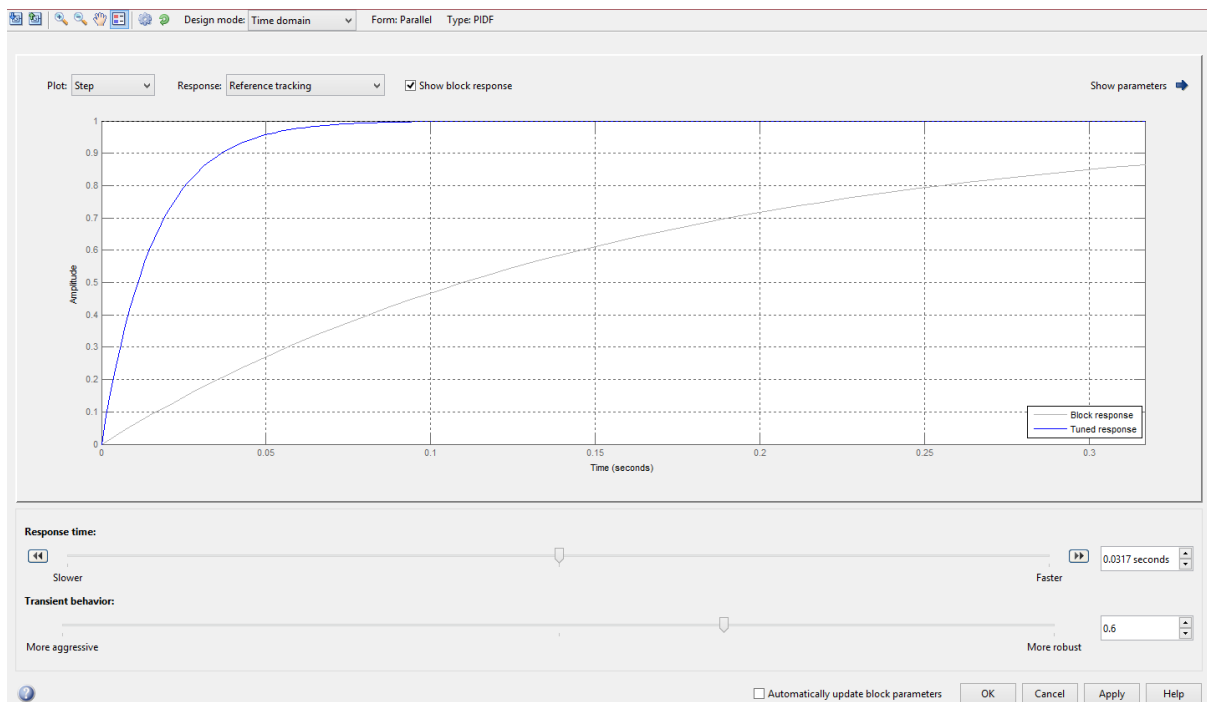


Figura 41. Interface de ajuste bloco Pidcontroller

Controlador por Algoritmos Genéticos

A Figura 42 apresenta, para a fase 2, a malha de controle cinemático, na qual parâmetros do controlador são estimados com algoritmos genéticos. Pretende-se controlar a posição do quadril e orientação do tronco do corpo do robô bípede. A cinemática envolvida é calculada analiticamente.

As posições e velocidades lineares desejadas, já conhecidas no extremo, são transformadas pela cinemática inversa em ângulos dos motores, das quais são subtraídas as magnitudes medidas para então obter o erro, que é inserido no

controlador. A ação de controle é aplicada no modelo do motor para obter posições e velocidades angulares, as quais são retroalimentadas e transformadas pela cinemática direta em posições e velocidades lineares do extremo.

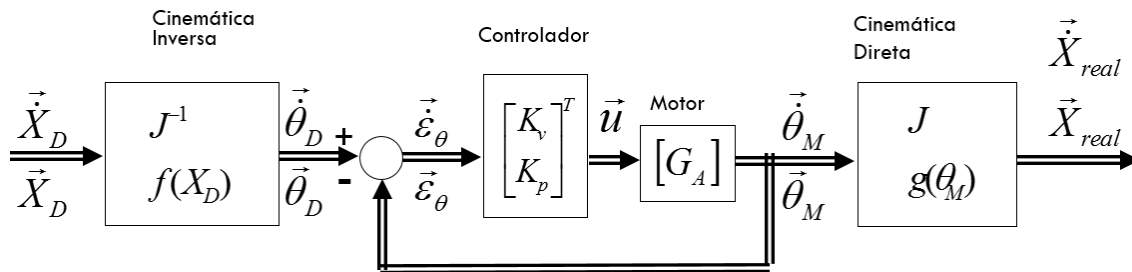


Figura 42. Malha de controle fase 2, controle de posição e orientação

O controlador é então representado por uma matriz de ganhos para posição e velocidade de cada motor, na forma

$$C = \begin{bmatrix} Kp1 & Kv1 & 0 & 0 & 0 & 0 \\ 0 & 0 & Kp2 & Kv2 & 0 & 0 \\ 0 & 0 & 0 & 0 & Kp3 & Kv3 \end{bmatrix} \quad (23)$$

No próximo capítulo, os algoritmos genéticos usados nesse trabalho e seus princípios são apresentados.

3 Algoritmos Genéticos

Algoritmos genéticos (*genetic algorithms*, GA) são técnicas de busca e otimização altamente paralela inspirados no princípio de seleção natural e reprodução genética de Charles Darwin [33], [34], onde os indivíduos mais aptos e com maior longevidade da população têm maior probabilidade de reprodução e de perpetuar seus códigos genéticos nas próximas gerações. Tais códigos genéticos constituem a identidade de cada sujeito, e estão representados nos cromossomos [35].

O cromossomo é a estrutura de dados que caracteriza o indivíduo ou possível solução do espaço de busca do problema; é submetido a um processo evolucionário que envolve avaliação, seleção, recombinação, mutação e substituição. Após vários ciclos ou gerações de evolução, a população deverá conter indivíduos mais aptos. A Figura 43 apresenta de maneira geral o processo evolucionário do algoritmo genético [36].

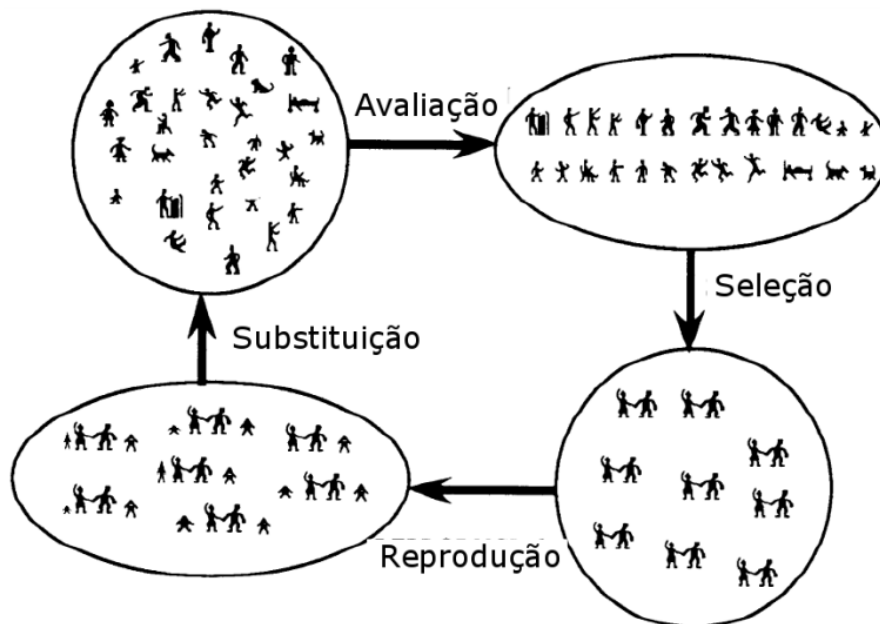


Figura 43. Processo evolutivo do algoritmo genético

A avaliação é a ligação entre o algoritmo genético e o problema, que fornece um valor quantitativo ao sujeito chamado de aptidão. A aptidão representa o quão adaptado está ou o quão bom é como resposta ao problema este sujeito, caracterizado por uma função específica ou função objetivo.

A seleção é o processo pelo qual procuram-se indivíduos para a reprodução, e aqueles com maior aptidão têm maior probabilidade de serem escolhidos. O operador de seleção tipicamente utilizado em GAs é a roleta, onde cada indivíduo é representado com uma faixa proporcional à sua aptidão relativa em um disco que gira. Os principais mecanismos de seleção são: proporcional, por torneios, com truncamento, e por normalização linear e exponencial [37].

O processo de recombinação ou reprodução utiliza o operador de “*crossover*”, considerado a característica fundamental nos GAs: pares de indivíduos genitores escolhidos da população chamados de pais aportam material genético ou informação para criar novos sujeitos diferentes chamados de filhos, que compartilham características de ambos os genitores. Os principais operadores de *crossover* são: de um ponto, de dois pontos e uniforme.

A mutação é um operador exploratório, já que dispersa a população pelo espaço de busca, modificando um valor no cromossomo se este for escolhido para mudar.

Substituição é o processo onde são selecionados os indivíduos que vão continuar na próxima geração. Os métodos mais conhecidos são: troca de toda a população, na qual novos indivíduos substituem toda a população anterior; troca de toda a população com elitismo, onde todos os cromossomos são substituídos, sendo o cromossomo mais apto da população corrente o único copiado na seguinte geração; troca parcial da população, na qual só são trocados os piores indivíduos permitindo manter os melhores; e finalmente troca parcial da população sem duplicados, onde os cromossomos duplicados também são trocados, além dos piores.

3.1. Geração de variáveis utilizando algoritmos genéticos

Obtenção dos ângulos dos motores

Procura-se descobrir a configuração de ângulos que produzem um passo com menor consumo de energia em relação ao deslocamento do pé, mantendo o centro de massa dentro do espaço abrangido pelo apoio, como mostra a Figura 44, mantendo assim seu equilíbrio quase-estático.

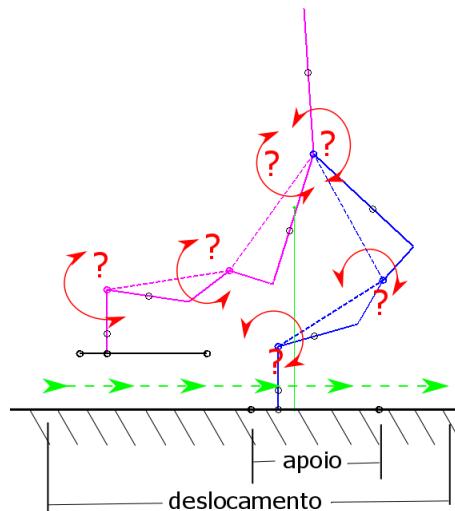


Figura 44. Objetivo do GA para os movimentos do robô bípede

O movimento do robô é construído por fotogramas ou quadros de um filme, onde cada fotograma é uma configuração de ângulos, como mostra a Figura 45.

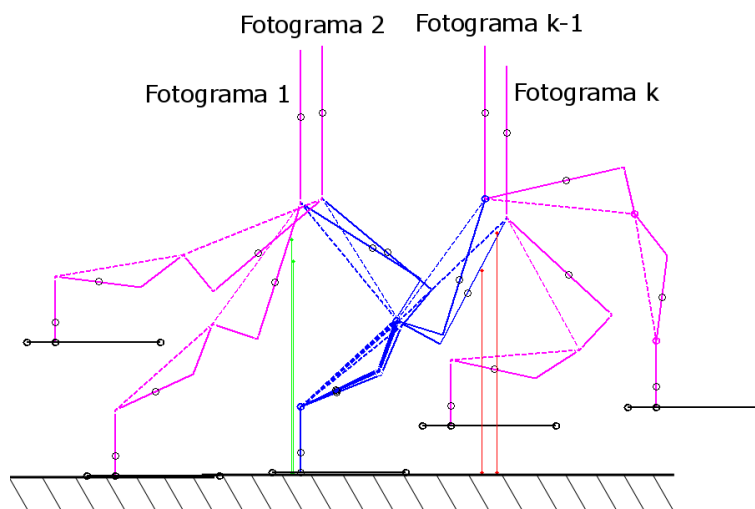


Figura 45. Fotogramas sucessivos

Obtenção dos parâmetros do controlador

Procuram-se os parâmetros do controlador que produzam a resposta mais próxima aos valores desejados, ou seja, que minimizem a função objetivo, mantendo a ação de controle dentro dos limites físicos permitidos. A Figura 46 exemplifica esta situação.

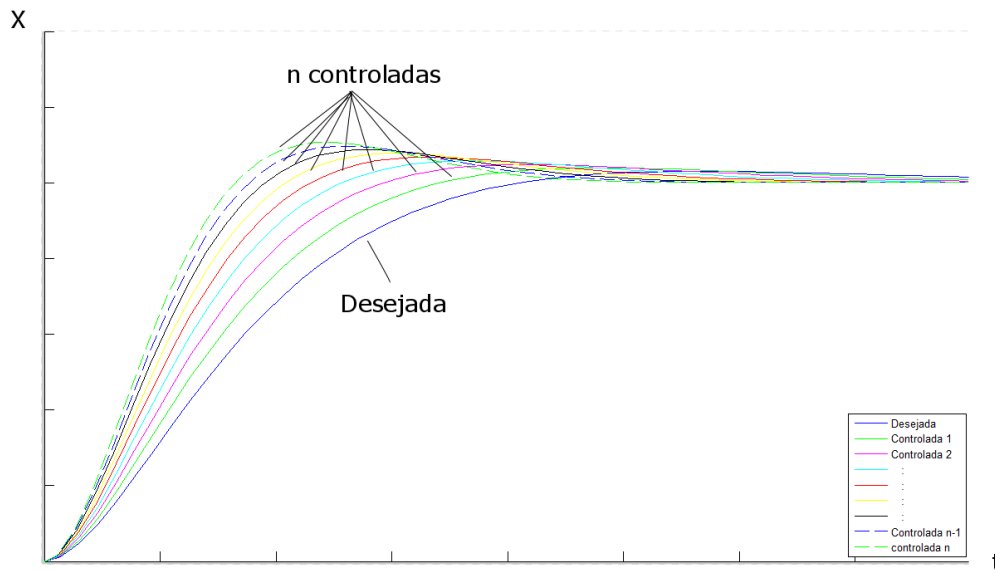


Figura 46. Variação de parâmetros do controlador

3.2. Função Objetivo

Caminhada

A função objetivo f_1 a ser minimizada para o caminhar está indicada na equação (24), como uma relação entre a energia mecânica e elétrica consumidas pelos motores dividida pelo deslocamento (E/D), mais as penalidades se houver. A energia mecânica é o produto entre o torque no motor e a velocidade angular pelo tempo, enquanto a elétrica é o produto entre a resistência interna do motor e a corrente elétrica ao quadrado pelo tempo.

$$f_1 = \frac{\sum_1^k \left(\sum_1^6 \tau * \omega + R * i^2 \right) * \Delta t}{D} + Penalidades \quad (24)$$

onde:

τ : torque

ω : velocidade angular

R: resistência elétrica

i: corrente elétrica

Δt : tempo entre fotogramas

D: deslocamento

Penalidades: punições ante configurações não permitidas

k: fotograma

A resistência elétrica foi definida em $6,3\Omega$ e a corrente elétrica foi interpretada como o torque no motor dividido pela constante de torque K_τ definida em $0,8182\text{Nm/A}$ utilizando a corrente de “stall torque”, como mostra a equação (25). Como explicado anteriormente, não foram feitos experimentos para obter estes valores, utilizou-se o documento de dados do fabricante e a caracterização feita por Mensink do motor “Dynamixel AX-12” [30], [25].

$$K_\tau = \frac{\tau_{ST}}{i_{ST}} \quad (25)$$

onde:

τ_{ST} : torque de “stall”

i_{ST} : corrente elétrica do “stall torque”

Controlador

A função objetivo f_2 a ser minimizada está indicada na equação (26), como o somatório dos erros quadráticos.

$$f_2 = \sum^n (Desejado - Obtido)^2 \quad (26)$$

onde:

Desejado: são os valores de entrada esperados

Obtido: são os valores de saída medidos

3.3. Penalizações nos Indivíduos

Consideraram-se situações que não podem ser permitidas ao indivíduo na configuração do robô. Caso estas aconteçam, as mesmas devem ser punidas para estabelecer que não são ângulos bons. As penalizações são aplicadas quando:

- A posição Y do quadril sai fora da faixa de altura considerada.
- O pé não avança na direção X positiva, ou seja, se ele não vai para frente.
- A posição da coordenada Y do pé é negativa, o que quer dizer que o pé está abaixo do chão, o que é impossível.
- O centro de massa está fora do espaço abrangido pelo pé de apoio, significando uma queda.
- O pé não avança na direção Y, seja positiva se é para subir, seja negativa se é para descer.

Estas penalizações são mostradas na Figura 47.

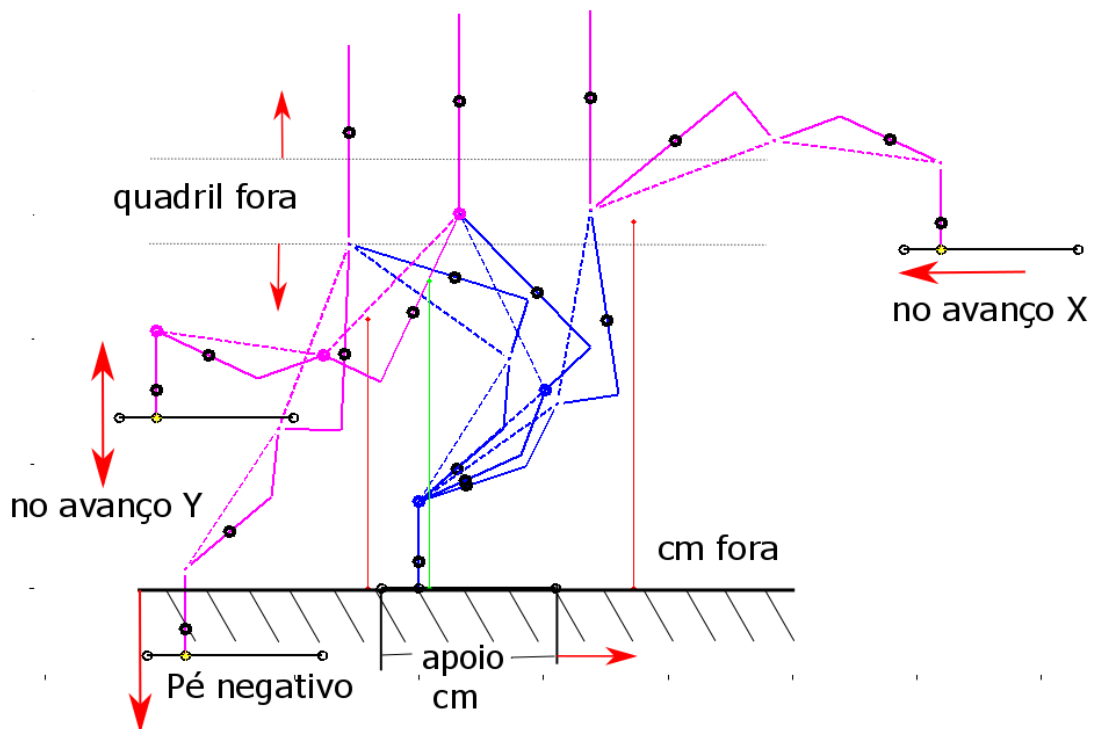


Figura 47. Penalizações, movimentos do bípede

3.4. Configuração do Cromossomo.

Para diminuir o custo computacional dos programas da caminhada, decidiu-se aproveitar a representação de seis motores com quatro variáveis, para diminuir o número de variáveis no espaço de busca (ver Seção 2.2 “ Definição dos ângulos envolvidos”). Foram utilizadas quatro (4) representações de cromossomo para gerar o valor de ângulos dos motores para cada fotograma, chamados assim: função polinomial, função cosseno, passo completo, e sub-passo, explicados adiante.

Para o programa dos parâmetros do controlador, utilizou-se uma única representação de cromossomo com as variáveis do controlador chamada de cromossomo de controle.

3.4.1. Cromossomo função polinomial

Nesta representação, os ângulos são fornecidos por uma função polinomial de grau seis (6), logo cada polinômio tem sete (7) coeficientes, para um cromossomo total de 28 variáveis ajustadas pelo GA. A Tabela 3 apresenta a estrutura, e a equação (27) os ângulos nesse caso.

Cromossomo															
θ1					θ2					...		θn			
a0	a1	a2	...	am	a0	a1	a2	...	am	...	a0	a1	a2	...	am

Tabela 3. Cromossomo polinômio

$$\theta_n = a_0 + a_1 * k + a_2 * k^2 + a_3 * k^3 + \dots + a_m * k^m \quad (27)$$

onde:

a: coeficiente

k: número do fotograma.

3.4.2. Cromossomo função trigonométrica (cosseno)

Nesta representação, os ângulos são fornecidos por uma função cosseno com 4 coeficientes para um cromossomo, num total de 16 variáveis ajustados pelo GA. A Tabela 4 apresenta a estrutura, e a equação (28) apresenta os ângulos nesse caso.

Cromossomo															
θ ₁				θ ₂				...				θ _n			
a	b	c	d	a	b	c	d	...				a	b	c	d

Tabela 4. Cromossomo cosseno

$$\theta_n = a + b * \cos(c * k + d) \quad (28)$$

onde:

- a: deslocamento vertical
- b: amplitude
- c: frequência
- d: fase ou deslocamento vertical
- k: número do fotograma

3.4.3. Cromossomo passo completo

Nesta representação, os ângulos são fornecidos diretamente pelo algoritmo genético. O cromossomo é configurado com 120 variáveis correspondentes a 30 posições consecutivas por quatro motores. A Tabela 5 apresenta a estrutura, e a equação (29) apresenta os ângulos nesse caso.

Cromossomo															
Fotograma 1				Fotograma 2				...				Fotograma k			
θ ₁	θ ₂	θ ₃	θ ₄	θ ₁	θ ₂	θ ₃	θ ₄	...				θ ₁	θ ₂	θ ₃	θ ₄

Tabela 5. Cromossomo passo completo

$$\begin{aligned}
 \theta_1(k) &= (4 * k) - 3 \\
 \theta_2(k) &= (4 * k) - 2 \\
 \theta_3(k) &= (4 * k) - 1 \\
 \theta_4(k) &= (4 * k)
 \end{aligned}
 \tag{29}$$

Esta configuração é a que tem maior abrangência no espaço de busca, já que admite que as variáveis dos indivíduos possam ter todos os possíveis valores permitidos. Ou seja, permite que o extremo do pé possa estar em qualquer ponto do plano X - Y pois a função de ângulo do motor pode ser qualquer. A Figura 48 apresenta possíveis trajetórias do pé para diferentes cromossomos.

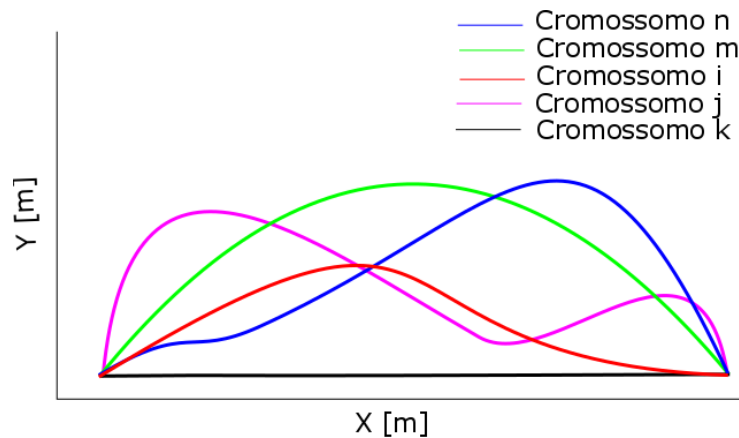


Figura 48. Possíveis trajetórias do pé para diferentes cromossomos

3.4.4. Cromossomo sub-passo

Nesta representação, os ângulos são fornecidos diretamente pelo algoritmo genético. Porém o cromossomo é configurado aqui com apenas 4 variáveis, em uma evolução por cada fotograma, em sub-passos. Esta é a construção de uma caminhada completa partindo de evoluções (sub-passos) em relação à posição anterior. Cada elo tem uma faixa de movimento entre $[-\theta : +\theta]$ onde o GA determina qual configuração de ângulos tem a menor relação de energia sobre deslocamento desde a posição anterior; ou seja, cada configuração é uma evolução incremental. Os fotogramas são calculados até que a posição do pé seja menor que um percentual arbitrário da distância geral do objetivo, que no robô adotado resultou em 5mm. A Tabela 6 apresenta a estrutura do cromossomo para cada fotograma, e a Figura 49

mostra uma representação dos possíveis comportamentos durante as evoluções sucessivas.

Cromossomo 1			Cromossomo 2					Cromossomo k				
Fotograma 1			Fotograma 2				...	Fotograma k				
θ_1	θ_2	θ_3	θ_4	θ_1	θ_2	θ_3	θ_4	...	θ_1	θ_2	θ_3	θ_4

Tabela 6. Cromossomo sub-passo

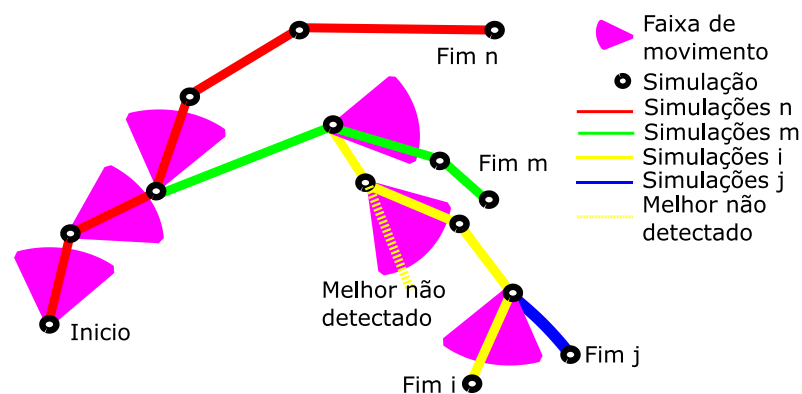


Figura 49. Possíveis comportamentos durante as evoluções sub-passo

Na Figura 49, desde o ponto de início, pode-se observar que na faixa de movimentos durante uma simulação de sub-passo é escolhida uma resposta e que, partindo dessa, outra é procurada. A Figura 50 apresenta possíveis trajetórias do pé construídas com evoluções consecutivas sub-passo.

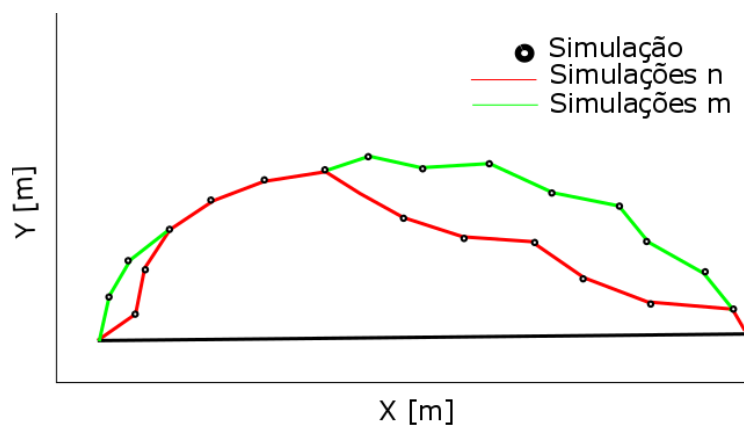


Figura 50. Possíveis trajetórias do pé construídas com sub-passo

3.4.5. Cromossomo de controle

Cromossomo com parâmetros fornecidos diretamente pelo algoritmo genético, com o objetivo de controlar o sistema. O cromossomo é configurado com 6 variáveis de ganhos. A Tabela 7 apresenta a estrutura.

Cromossomo					
Kp1	Kp2	Kp3	Kv1	Kv2	Kv3

Tabela 7. Cromossomo de controle

No próximo capítulo, são apresentados os resultados da incorporação dos GA na otimização de caminhada do robô bípede plano considerado.

4 Resultados

4.1. Controle Cinemático

PID auto tuning

Neste caso, as entradas desejadas são expressas como transições suaves entre fotogramas, como mostra-se na Seção 2.5. A Figura 51 apresenta uma transição entre dois fotogramas do motor 1 para ajuste da condição inicial da caminhada, passando da posição 0 a 64 graus em 5 segundos, tempo de segurança pensado para não danificar os motores, caso aconteça colisão e seja preciso desligá-lo³.

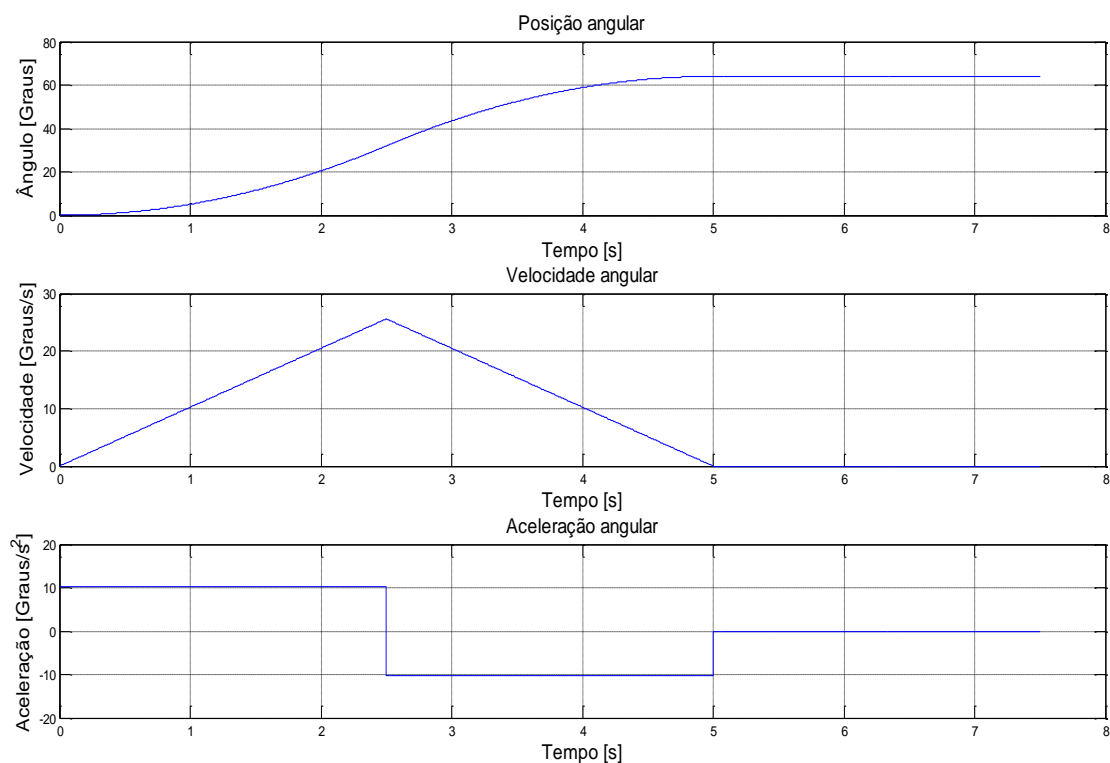


Figura 51. Transição de entradas angulares desejadas

³ O motor tem sensor de torque, mas em caso de mudança abrupta pode danificá-lo

Os parâmetros fornecidos pelo auto ajuste do bloco “Pidcontroller” são:

$P=136,59$

$I= 168,17$

$D=0,40$

$N=200$

A Figura 52 apresenta a resposta dos motores controlados com os parâmetros anteriores. Pode-se apreciar que os sinais foram controlados dando seguimento à trajetória desejada, cuja avaliação pelo somatório do erro quadrático foi (em graus ao quadrado):

Motor 1: 2,9771. Motor 2: 1,3087. Motor 3: 0,2319. Motor 4: 0,9886. Motor 5: 1,9243. Motor 6: 3,2850. Somando tudo, tem-se: 10,7158. O melhor comportamento foi o motor 3 com 0,2319, e o menos comportado o 6, com 3,2850.

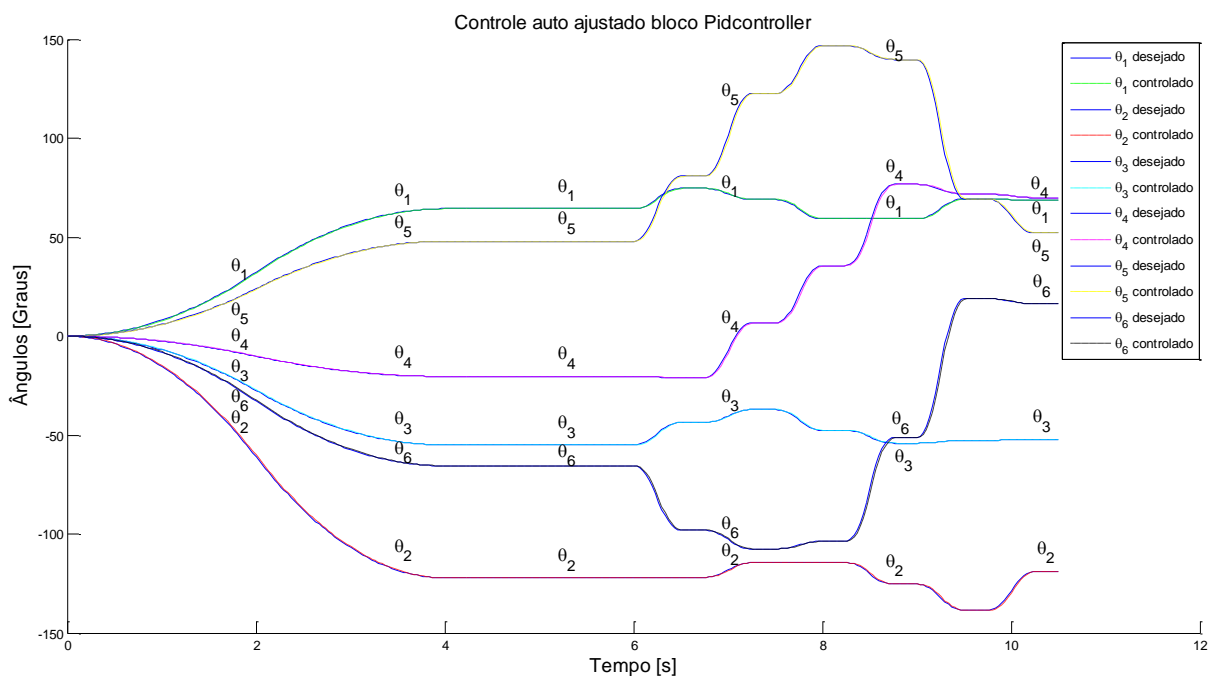


Figura 52. Resposta dos motores ao controle auto ajustado

A Figura 53 mostra um *zoom* da resposta do motor 1 entre 6s e 11s.

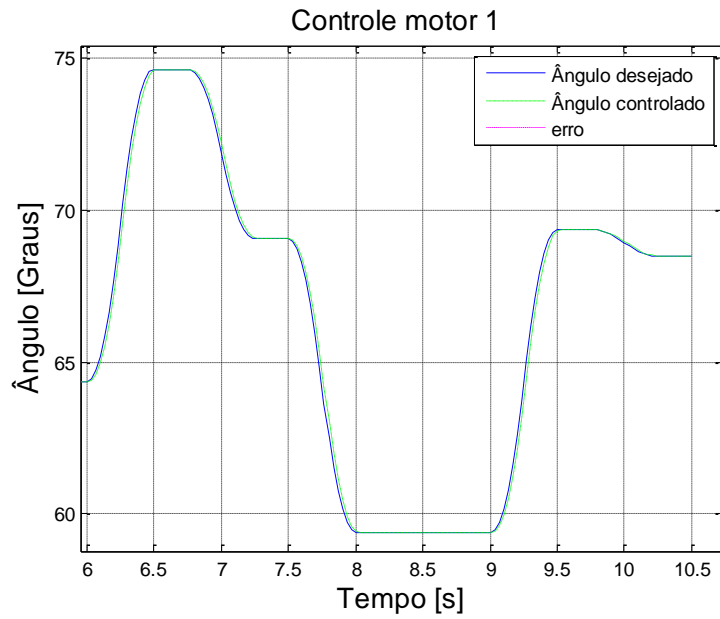


Figura 53. Resposta do motor 1 ao controle auto-ajustado

A Figura 54 mostra a resposta ao degrau do motor 1 com valores fornecidos pelo bloco auto-ajustado, onde não se apresenta sobrepasso e chega a 98% em 0,067s.

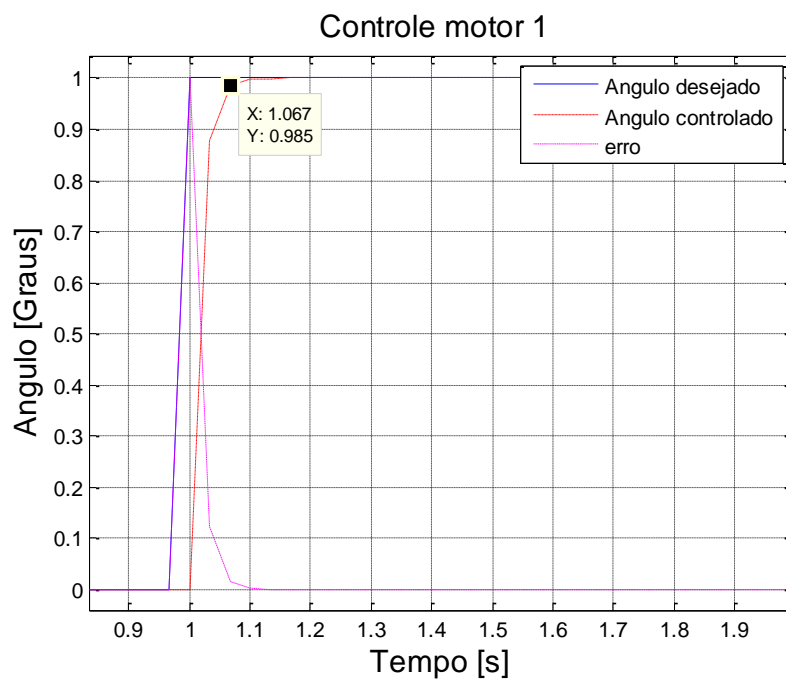


Figura 54. Resposta ao degrau motor 1 com bloco auto ajustado

Na Figura 55 e Figura 56 se apresentam a resposta ao controle da posição do quadril e orientação do corpo. A avaliação pelo somatório dos erros quadráticos para X foi $2,51e-6 \text{ m}^2$ e para Y foi $1,74e-7 \text{ m}^2$.

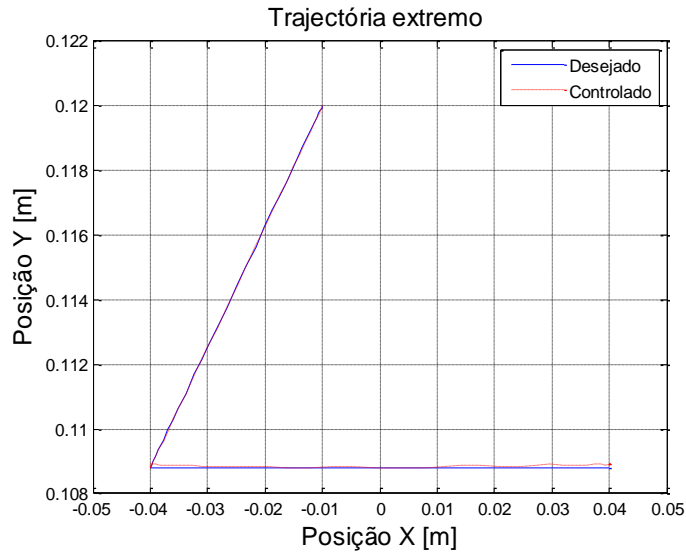


Figura 55. Trajetória do extremo, controlador PID

Da Figura 56 observa-se a orientação do corpo, o qual deve se manter vertical, mas teve amplitude de oscilação considerável perto dos 0,16 graus. A avaliação pelo somatório dos erros quadráticos foi de $0,04 \text{ graus}^2$.

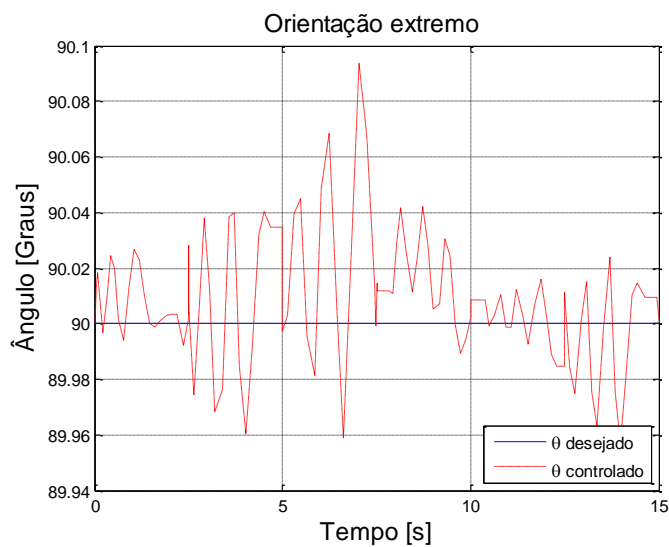


Figura 56. Orientação, controlador PID

Resultado do controlador heurístico

As entradas desejadas são expressas como transições suaves entre fotogramas, como mostra-se na seção 2.5. A Figura 57 apresenta uma transição entre dois fotogramas da posição do quadril no eixo X, passando de -10mm para -40mm.

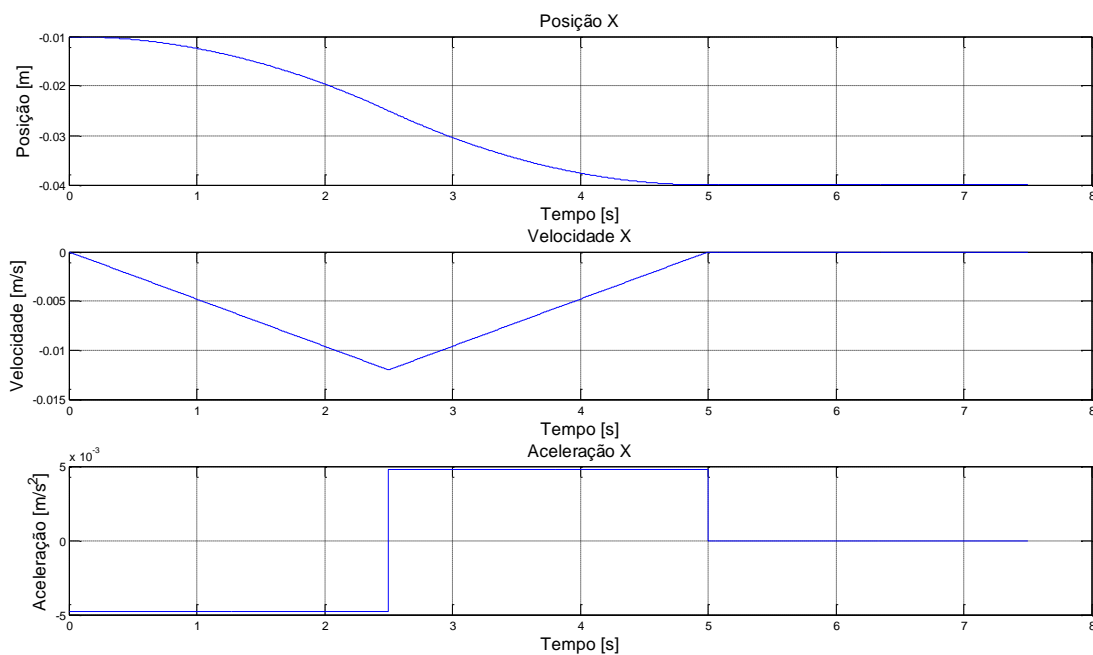


Figura 57. Transição de entradas desejadas

Inicialmente foram feitos testes heurísticos onde observou-se um melhor comportamento para valores de k_p altos em relação aos de k_v . As Figura 58 a Figura 60 mostram a resposta a um grupo de parâmetros heurísticos: $k_{p1}=4$, $k_{v1}=4$, $k_{p2}=0,5$, $k_{v2}=0,5$, $k_{p3}=0,5$, $k_{v3}=0,5$.

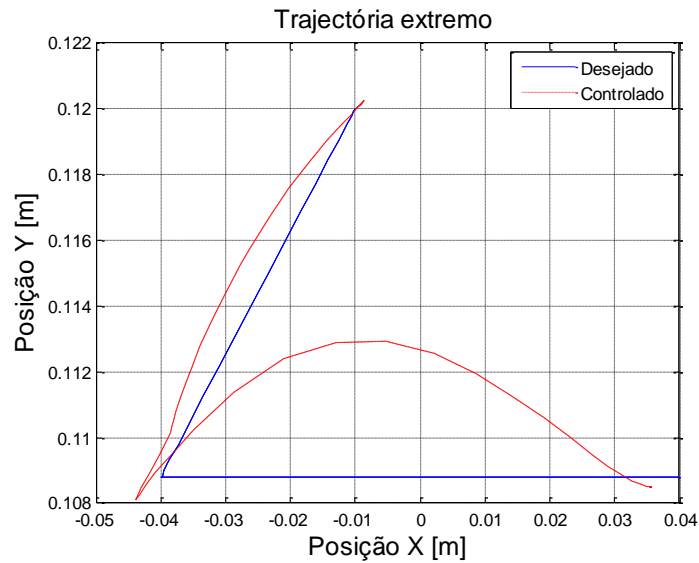


Figura 58. Trajetória do quadril, controlador com parâmetros heurísticos

Pôde-se apreciar da figura anterior que a trajetória controlada do quadril não acompanhou corretamente os valores desejados, ao se desviar na parte de descida inclinada, e ampla oscilação próxima de 20mm na parte horizontal.

Na Figura 59 observa-se a posição das coordenadas X e Y do extremo no tempo. Nota-se que a coordenada Y foi melhor comportada, com uma pequena oscilação em relação a X que tardou em acompanhar o sinal desejado. A avaliação pelo somatório dos erros quadráticos foi para X 0,0316 e para Y 0,0011, ambos em m^2 .

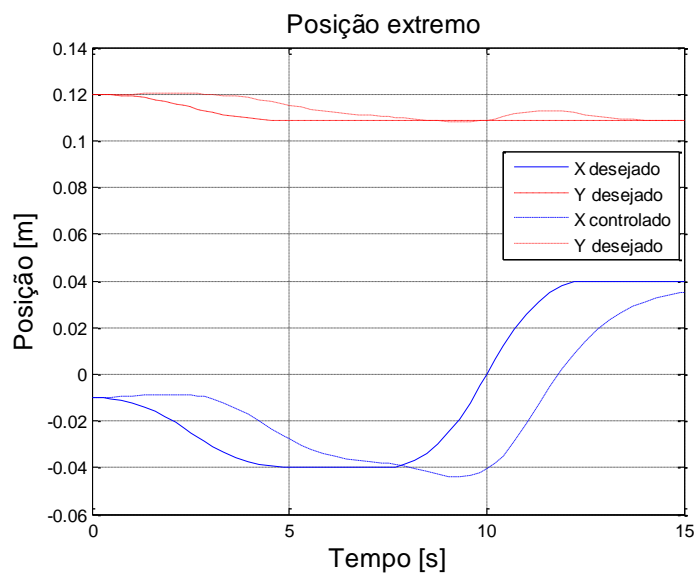


Figura 59. Posição do quadril, controlador com parâmetros heurísticos

Na Figura 60 observa-se a orientação do corpo, o qual deve-se manter vertical, mas teve amplitude de oscilação considerável perto dos 10 graus. A avaliação pelo somatório dos erros quadráticos foi de 549,9 graus².

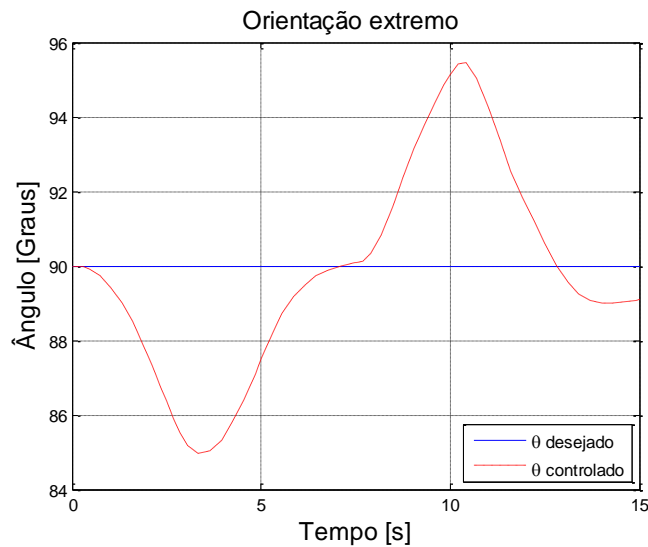


Figura 60. Orientação, controlador com parâmetros heurísticos

4.1.1. Ajuste do controlador com Algoritmos genéticos.

Controle da posição do quadril e orientação do corpo

As Figura 61 a Figura 63 apresentam a resposta aos parâmetros ajustados pelo GA. As características do computador no qual os programas foram rodados são: processador “i7-5950HQ” com frequência 2.9GHz, memória “RAM” 16Gb, sistema operacional “Windows 10”. Os parâmetros do GA são: gerações 600, população 60, seleção roleta, mutação um ponto 65% e crossover 5%. O tempo de cálculo computacional foi de 1003s, aproximadamente 17min. Parâmetros ajustados pelo GA: $k_p=99,91$, $k_v1=0,25$, $k_p2=92,80$, $k_v2=0,01$, $k_p3=77,36$, $k_v3=3,06$.

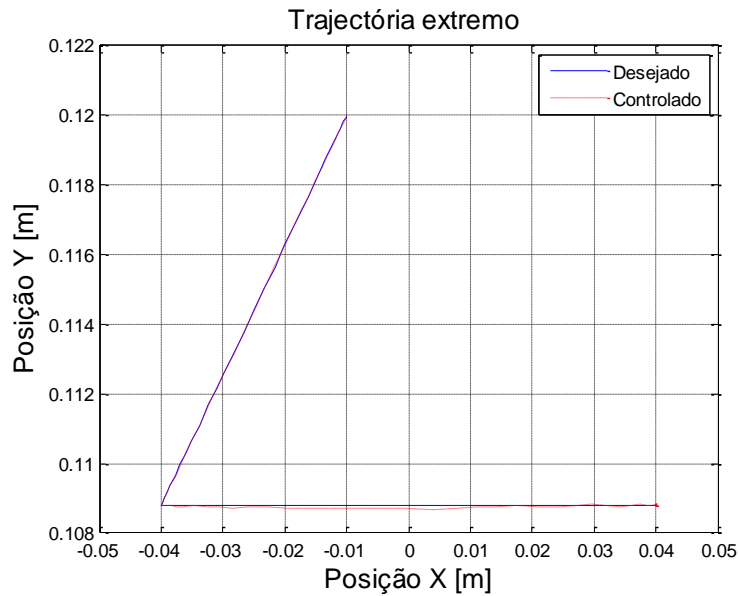


Figura 61. Trajetória do quadril, parâmetros ajustados pelo GA

Pôde-se apreciar da figura anterior que a trajetória controlada do quadril acompanhou corretamente os valores desejados, ao não se desviar na parte de descida inclinada, e uma leve oscilação próxima de 0,2mm na parte horizontal.

Na Figura 62 observa-se a posição das coordenadas X e Y do extremo no tempo. Nota-se que as duas coordenadas foram muito bem-comportadas com a avaliação pelo somatório dos erros quadráticos para X $5,0946e-06$ e para Y $1,7417e-07$, ambas em m^2 .

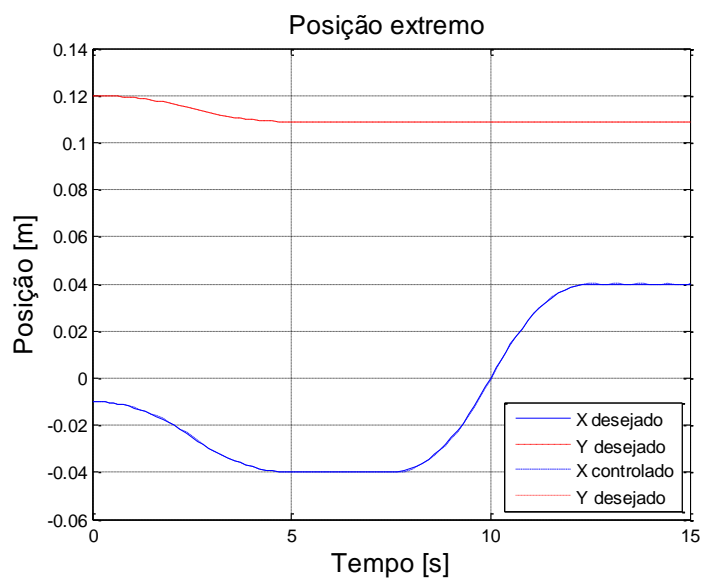


Figura 62. Posição do quadril, parâmetros ajustados pelo GA

Na Figura 63 observa-se a orientação do corpo, o qual deve-se manter vertical. Houve amplitude de oscilação relativamente pequena perto dos 0,1 graus. A avaliação pelo somatório dos erros quadráticos foi de 0.0594 graus².

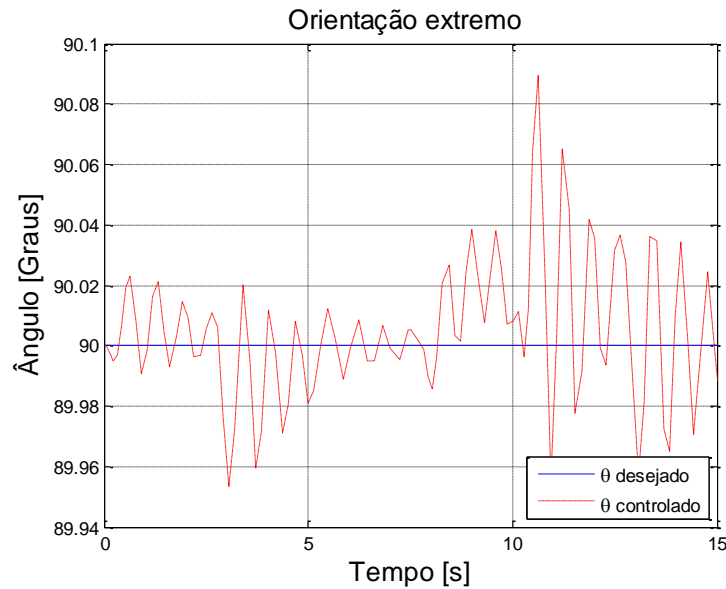


Figura 63. Orientação do quadril, parâmetros ajustados pelo GA

A Tabela 8 apresenta o resumo das avaliações para as diferentes sintonizações de controlador, na qual pôde-se apreciar que o melhor comportamento no ajuste dos parâmetros foi o Pid auto ajustado com o bloco “Pidcontroller” do “Simulink”.

Sintonização	Somatório do erro quadrático (m ² ou grau ²)		
	Posição X	Posição Y	Orientação tronco
Heurística	0,0316	0,0011	549,9292
Algoritmo genético	5,09E-06	1,74E-07	0.0594
PID auto ajustado	2,51E-06	1,13E-06	0,04

Tabela 8. Resumo da avaliação de sintonizações do controlador

4.2. Caminhada heurística

O robô bípede físico montado só tem objetivo de reproduzir os ângulos fornecidos pelo *gerador de padrão de marcha*. Neste experimento, incrementou-se a sua velocidade de movimento durante um passo até o sistema tombar pelo efeito das forças centrífugas e de Coriolis [26], já que o pé não está colado no chão. Observou-se que para um passo em um tempo inferior a 4s a condição quase-estática é altamente afetada.

A primeira configuração para as pernas do robô bípede foi pensada de maneira similar à de uma ave, como é mostrado na Figura 64 [38], [39], a qual faz uma comparação entre o esqueleto de um avestruz e de um ser humano. Observa-se que o fêmur do animal é muito mais curto do que a tíbia, e que o joelho está muito mais perto do quadril quando comparado com o quadril de uma pessoa, guardando as mesmas proporções. Na mesma figura também se mostra a localização dos motores.

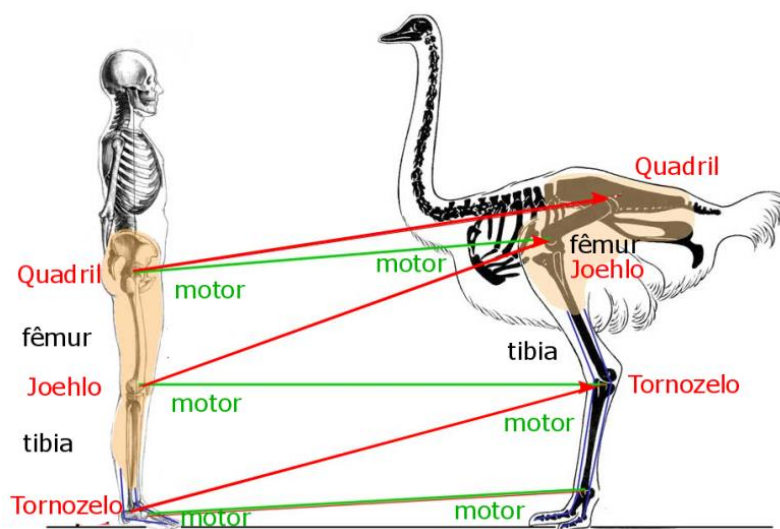


Figura 64. Comparação entre o esqueleto de um avestruz e de um ser humano

4.2.1. Simulações

O robô físico foi representado no software “Matlab” nas primeiras simulações com quatro elos retos rígidos e quatro motores com ângulos chamados de α , como mencionado na Seção 2.2. A posição de cada elo foi determinada de forma empírica observando o caminhar de uma galinha em vídeo. A Figura 65 apresenta nove fotogramas com 0,222 segundos cada. O vídeo só teve a função de guia de referência para os ângulos das pernas, já que o bípede tratado neste documento tem caminhada quase-estática.

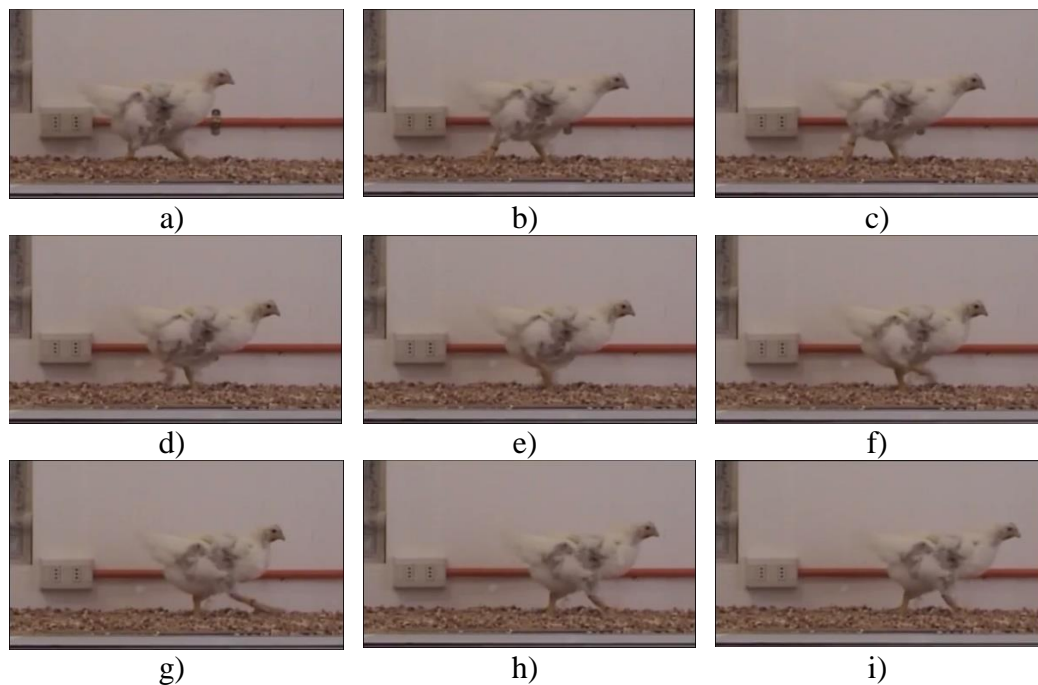
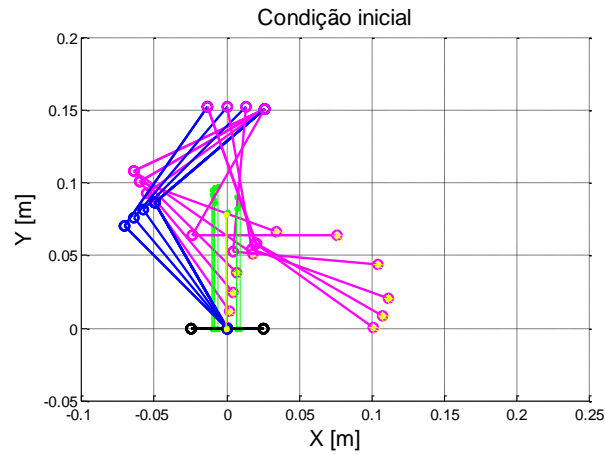
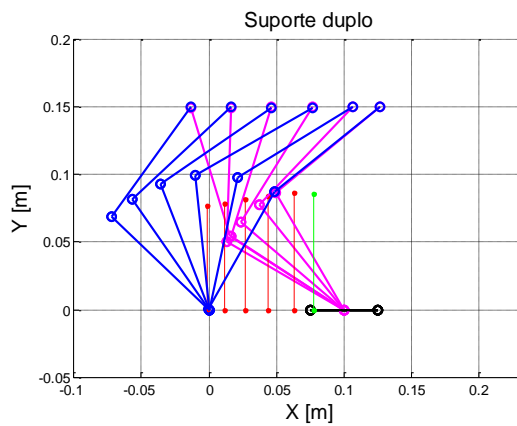


Figura 65. Fotogramas do caminhar de uma galinha

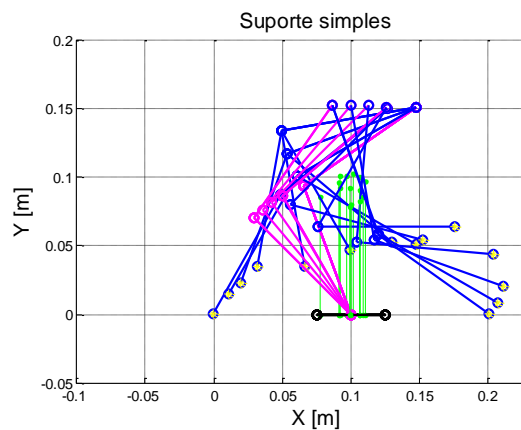
A Figura 66a) apresenta as posições dos elos para a condição inicial; a Figura 66b) a fase de suporte duplo; e a Figura 66c) a fase de suporte simples, como mencionado na Seção 1.5.



a)



b)



c)

Figura 66. Simulação caminhada heurística

As dimensões iniciais fornecidas para o programa foram adquiridas a partir do software “SolidWorks”, no qual realizou-se a montagem das peças desenhadas em 3D fornecidas pelo fabricante do kit “Bioid Premium”, de dimensões muito similares, mas que não correspondiam exatamente as peças do robô real finalmente montado com peças do kit “Bioid GP”, como mostrado na Figura 67 [24]. No apêndice são apresentados os desenhos em “Solidworks”.

4.2.2. Experimento inicial

Os ângulos obtidos empiricamente foram programados, ajustados e reproduzidos no sentido inverso, pois decidiu-se utilizar o robô bípede em configuração de movimentos naturais das pernas de um ser humano (joelhos flexionando para frente, ao invés de para trás), como apresenta a Figura 67 com nove fotogramas sequenciais atemporais. Um trabalho futuro poderia retomar a configuração “tipo ave” e dar a possibilidade ao bípede de utilizar as duas, como faz o robô “Drc-Hubo” [4].

Os pés, construídos com cortes em uma régua metálica, têm suporte lateral para permitir os movimentos em um único plano.

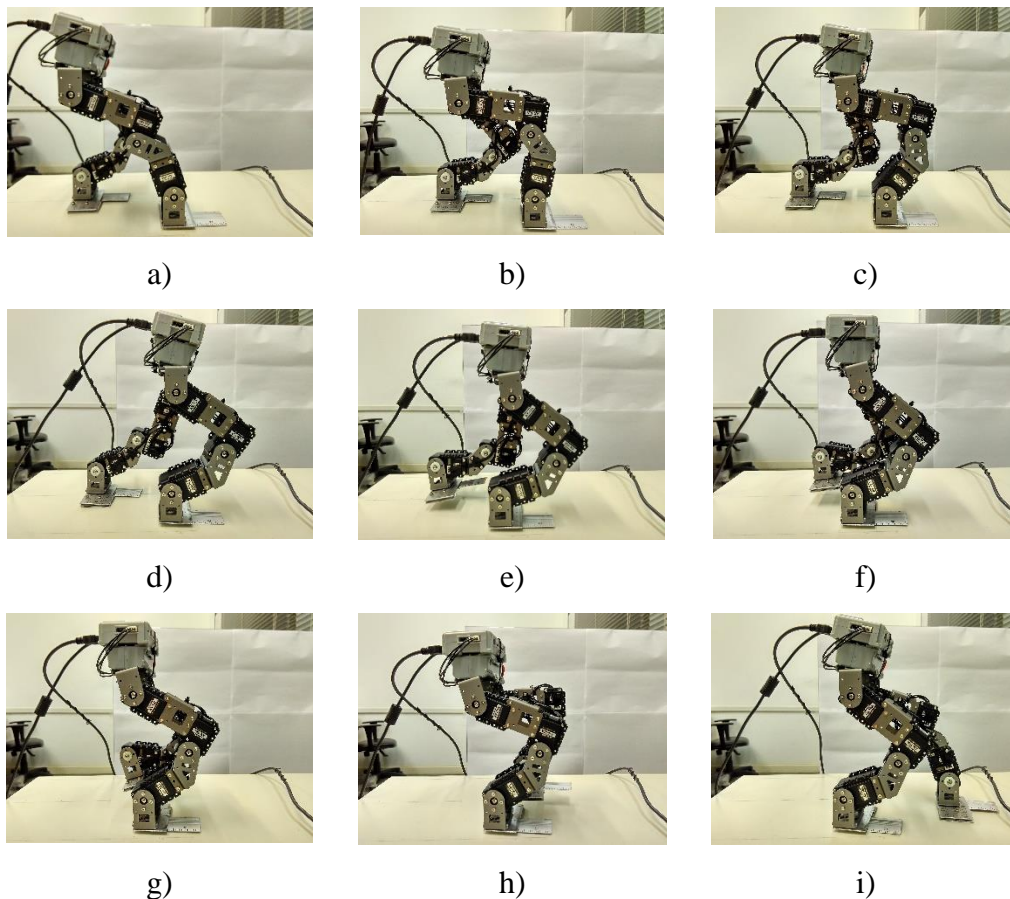


Figura 67. Fotogramas do robô bípede físico montado

As simulações na primeira fase são construídas com intervalo de 0,1s entre cada fotograma para um total de 30. A partir dos resultados, determinaram-se: a faixa de altura do quadril entre 13cm e 15cm do chão; a distância horizontal máxima de avanço entre os pés em 10cm; o período do *padrão de marcha* em 8s, o que compreende dois passos; cada passo de 3s em fase 1 ou de suporte simples e 1s em fase 2 ou de suporte duplo.

4.3. Caminhada com algoritmos genéticos

Para as simulações individuais determinou-se que o tempo entre os fotogramas seria de 0,1s e o tempo para um passo de 4s, já que durante os testes iniciais no robô físico observou-se que um tempo menor afeta consideravelmente a condição quase-estática, ou seja, movimentos mais rápidos fazem o robô cair já que o pé não está fixo no chão (ver Seção 4.2.2). Denomina-se 1 passo a execução da fase 1 e fase 2, e 2 passos como o período total do gráfico dos ângulos de cada motor, equivalente a 8 segundos. As características do computador no qual os programas foram rodados são: processador “i7-5950HQ” frequência 2.9GHz, memória “RAM” 16Gb, sistema operacional “Windows 10”.

4.3.1. Função polinomial

Neste caso, usam-se funções polinomiais cujos coeficientes são ajustados pelo GA. As configurações penalizadas foram: pé negativo (abaixo do solo), centro de massa fora (da projeção de estabilidade), e nível do quadril (a ser controlado). Os parâmetros do GA são: gerações 8000, população 3000, seleção roleta, mutação dois pontos 65% e crossover 5%. O tempo de cálculo computacional foi de 93.573s, aproximadamente 26 horas. A Figura 68 apresenta a resposta da trajetória do pé em um passo para os ângulos calculados com a equação polinomial (27) de coeficientes ajustados pelo GA (ver Seção 3.4.1).

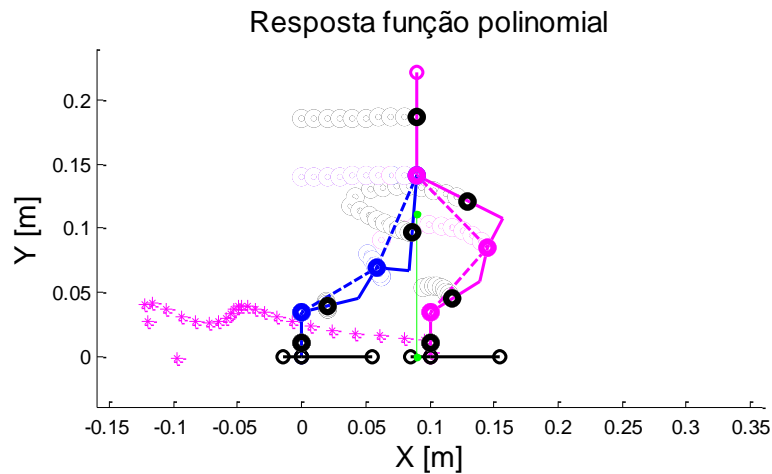


Figura 68. Resposta do GA com função polinômio

A Figura 69 apresenta os ângulos dos seis motores durante 8s, dois passos equivalentes a um período do padrão de marcha. Vide a Seção 2.2, “ Definição dos ângulos envolvidos”, na qual se relacionam estes ângulos com os ângulos intermediários fornecidos pelo GA.

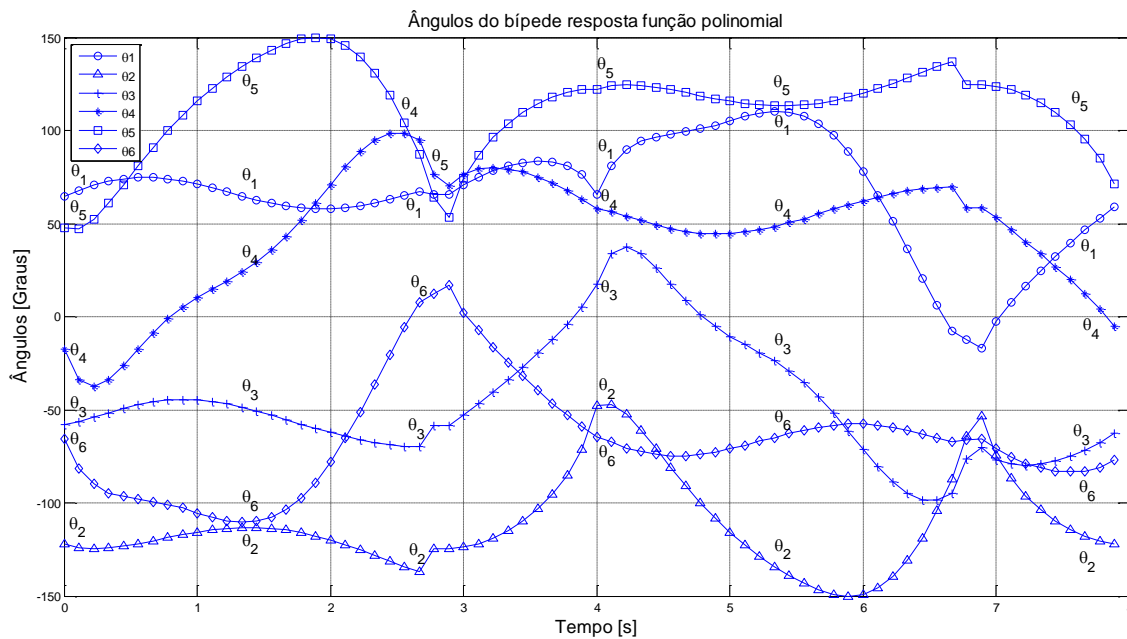


Figura 69. Ângulos dos motores, para função polinômio

A Figura 70 apresenta o incremento da energia consumida durante dois passos e a configuração do bípode em instantes de tempo.

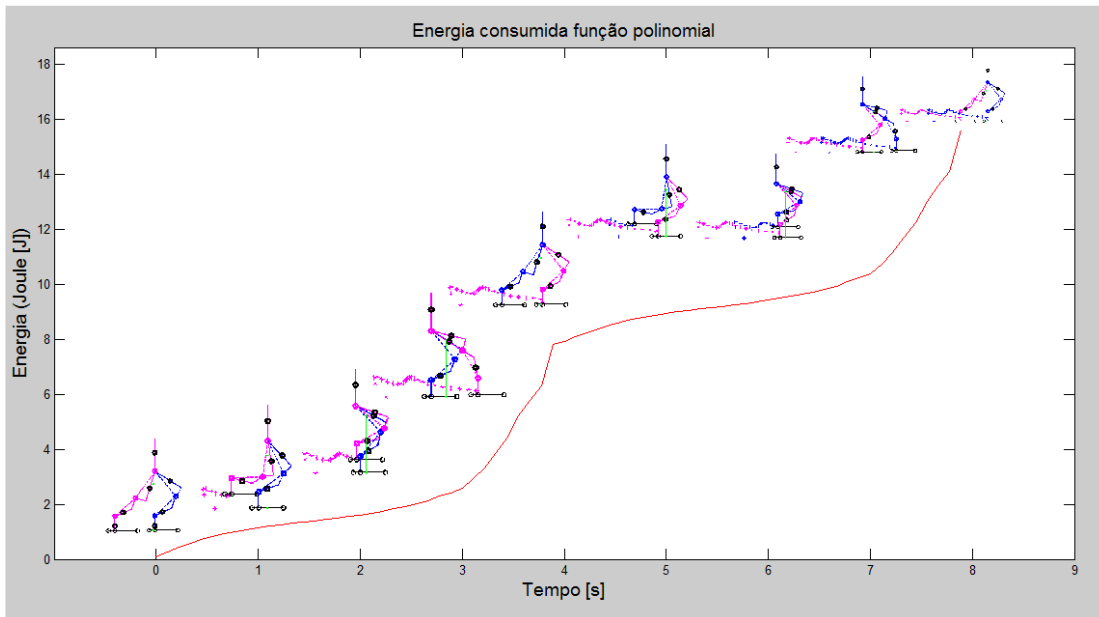


Figura 70. Energia consumida, para função polinômio

A energia consumida durante o período do padrão de marcha foi de 15,61J e a relação de energia sobre deslocamento foi de 39,5J/m.

A Figura 71a) apresenta a evolução da aptidão do melhor indivíduo de cada geração, sendo a Figura 71b) um zoom entre os valores das gerações na faixa de 2000 e 7000. Selecionaram-se as gerações número 100, 1000, 3000, 4000, 6500 e 7500 para observar a trajetória do pé, já que representam mudanças representativas na evolução, as quais são devidas às penalizações. Por cada punição do indivíduo, a unidade de aptidão foi incrementada em 10 vezes para indicar que não são ângulos bons.

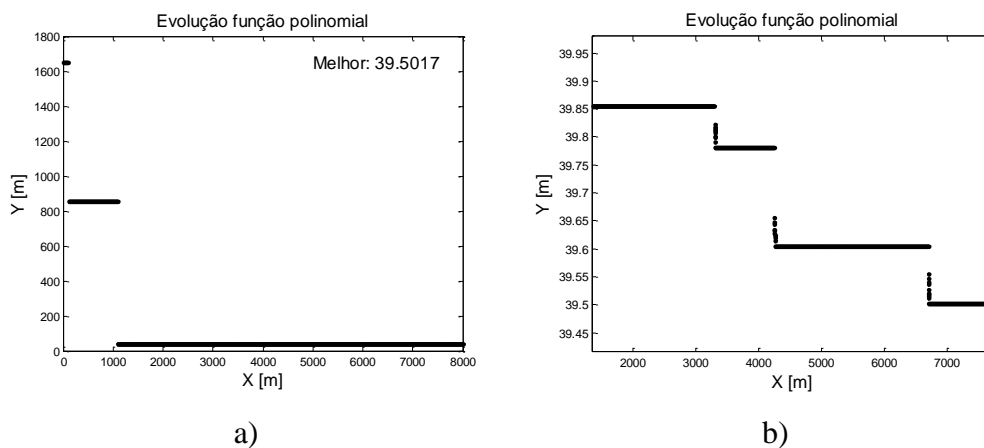


Figura 71. Evolução para função polinômio

A Figura 72 apresenta a evolução da trajetória do pé.

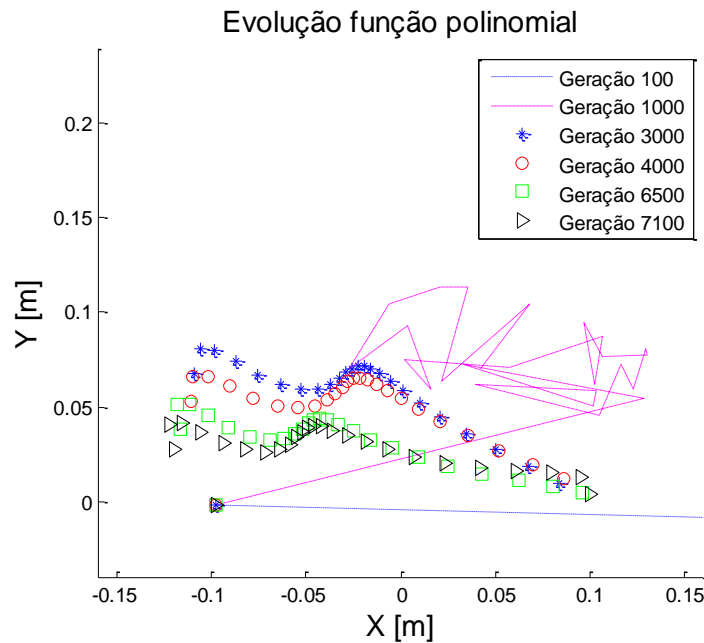


Figura 72. Evolução da trajetória do pé, para função polinômio

4.3.2. Função trigonométrica (cosseno)

Neste caso, usam-se funções trigonométricas (cossenos) cujos coeficientes são ajustados pelo GA. As configurações penalizadas foram as mesmas que as anteriores: pé negativo, centro de massa fora, e nível quadril. Os parâmetros do GA também foram os mesmos. O tempo de cálculo computacional foi de 33.796s, aproximadamente 9 horas e 15 minutos.

A Figura 73 apresenta a resposta da trajetória do pé em um passo para os ângulos calculados com a função cosseno (28) de coeficientes ajustados pelo GA (ver Seção 3.4.2).

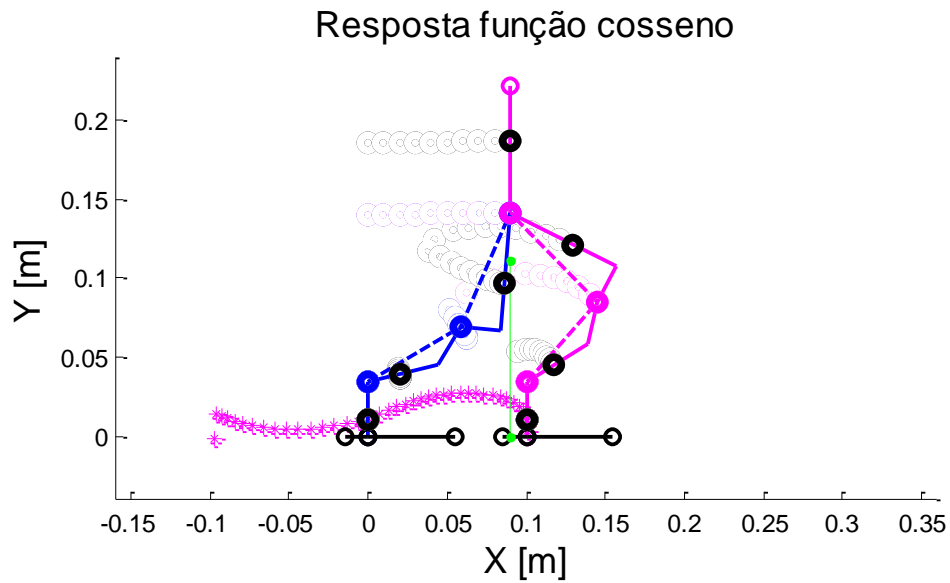


Figura 73. Resposta AG função cosseno

A Figura 74 apresenta os ângulos dos seis motores durante 8s, dois passos equivalentes a um período do padrão de marcha.

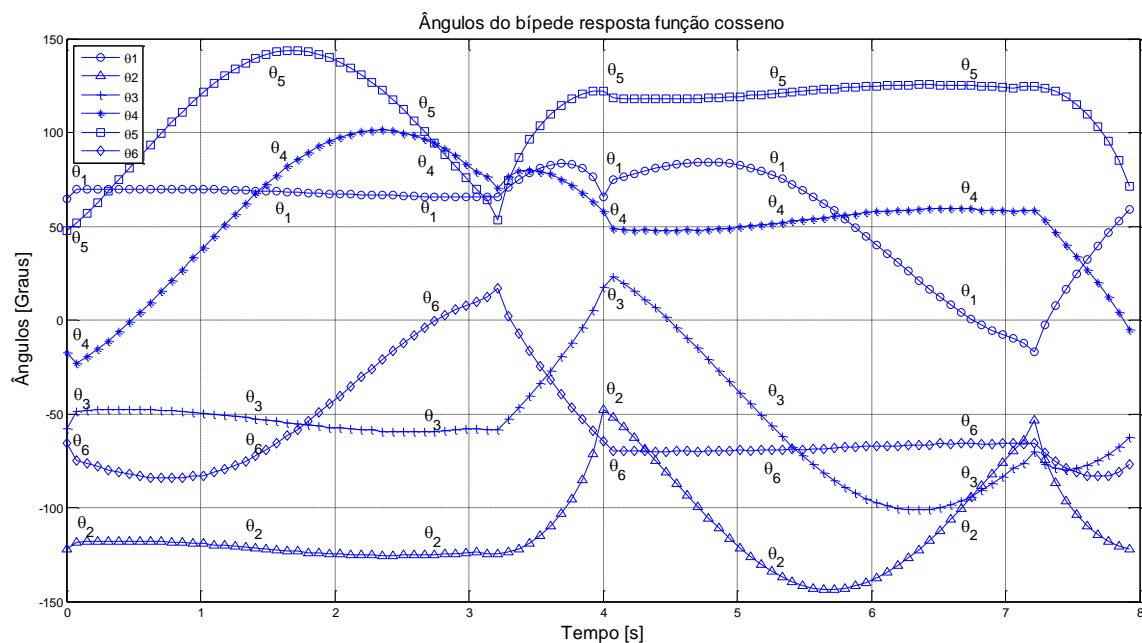


Figura 74. Ângulos dos motores, para função cosseno

A Figura 75 apresenta o incremento da energia consumida durante dois passos e a configuração do robô bípede em diversos instantes.

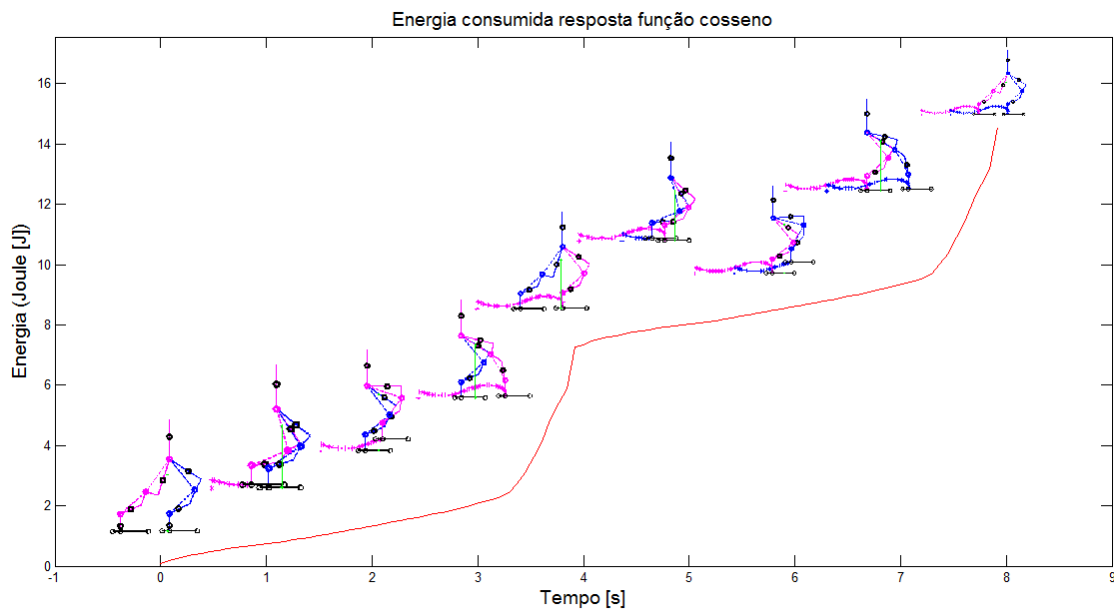


Figura 75. Energia consumida, para função cosseno

A energia consumida durante o período do padrão de marcha foi de 14,55J e a relação de energia sobre deslocamento foi de 36,8J/m.

A Figura 76 apresenta a evolução da aptidão do melhor indivíduo de cada geração. Selecionaram-se as gerações número 100, 200, 800, 1000, 1600 e 6669 para observar a trajetória do pé, já que representam mudanças representativas na evolução. Pode-se observar que a simulação parou antes das 8000 gerações, já que não tinha mudança na avaliação durante um terço das gerações, ou seja 2666 gerações.

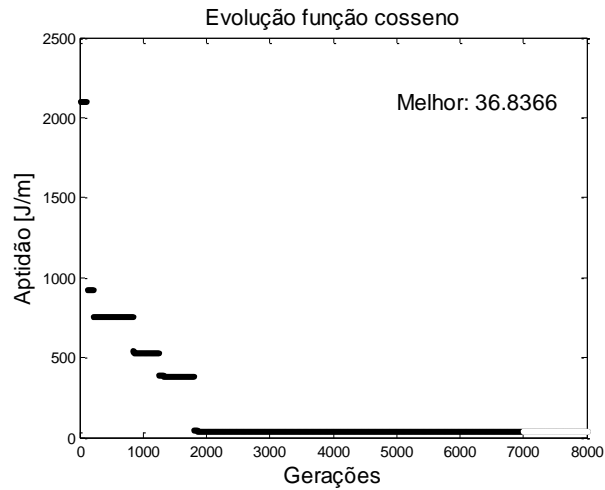


Figura 76. Evolução para função cosseno

A Figura 77 apresenta a evolução da trajetória do pé.

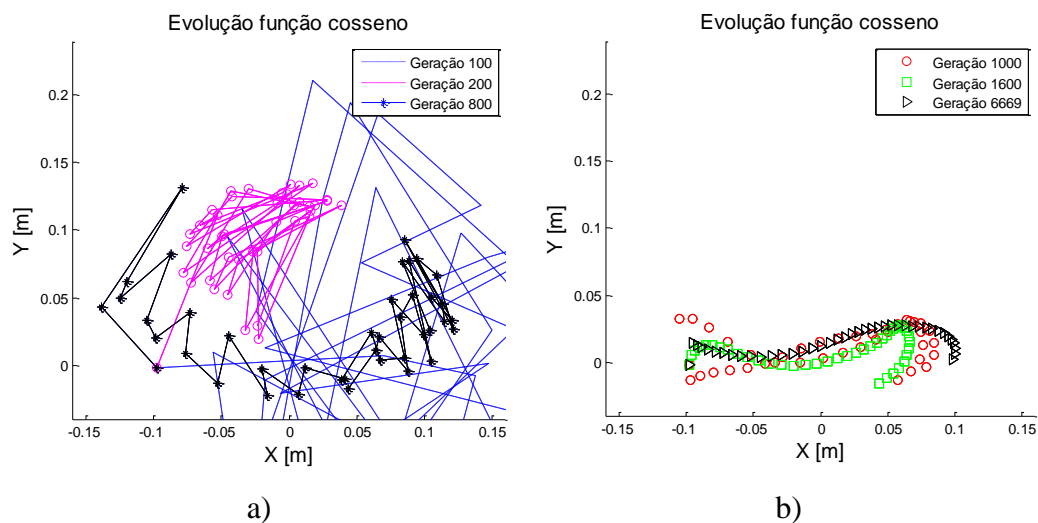


Figura 77. Evolução trajetória do pé, para função cosseno

4.3.3. Passo completo

Passo completo A

Para esta forma de representação do passo, explicada anteriormente, as configurações penalizadas foram: se o pé não avança na direção X, se a posição da coordenada Y do pé é negativa, se o centro de massa está fora do espaço abrangido

pelo pé de apoio, se a posição Y do quadril sai fora da faixa de altura considerada. Parâmetros do GA foram os mesmos anteriores: gerações 8000, população 3000, seleção roleta, mutação 65% de dois pontos e crossover 5%. O tempo de cálculo computacional foi de 649.586s, aproximadamente 7,5 dias.

A Figura 78 apresenta a resposta da trajetória do pé em um passo para os ângulos fornecidos diretamente pelo GA (ver seção 3.4.3).

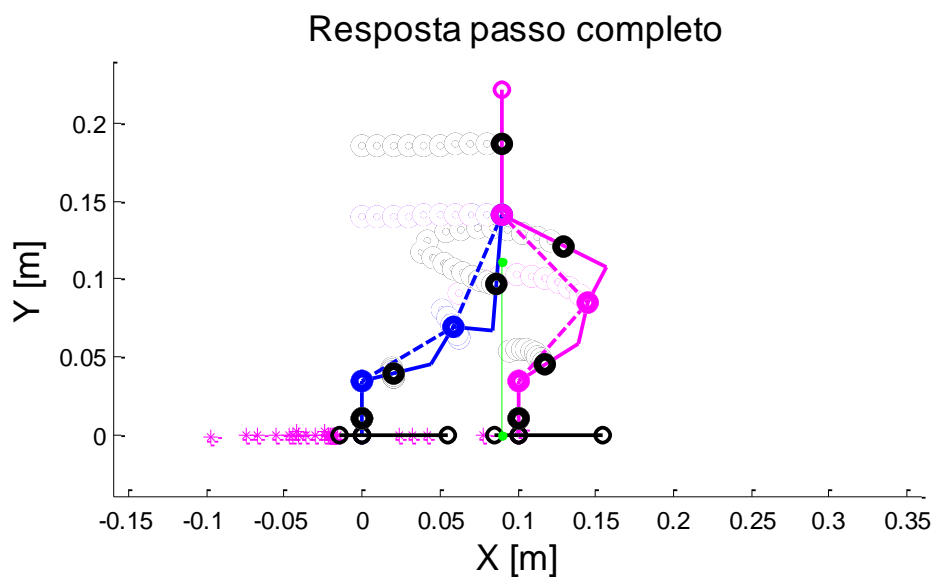


Figura 78. Resposta do GA para passo completo

A Figura 79 apresenta os ângulos dos seis motores durante 8s, dois passos equivalentes a um período do padrão de marcha.

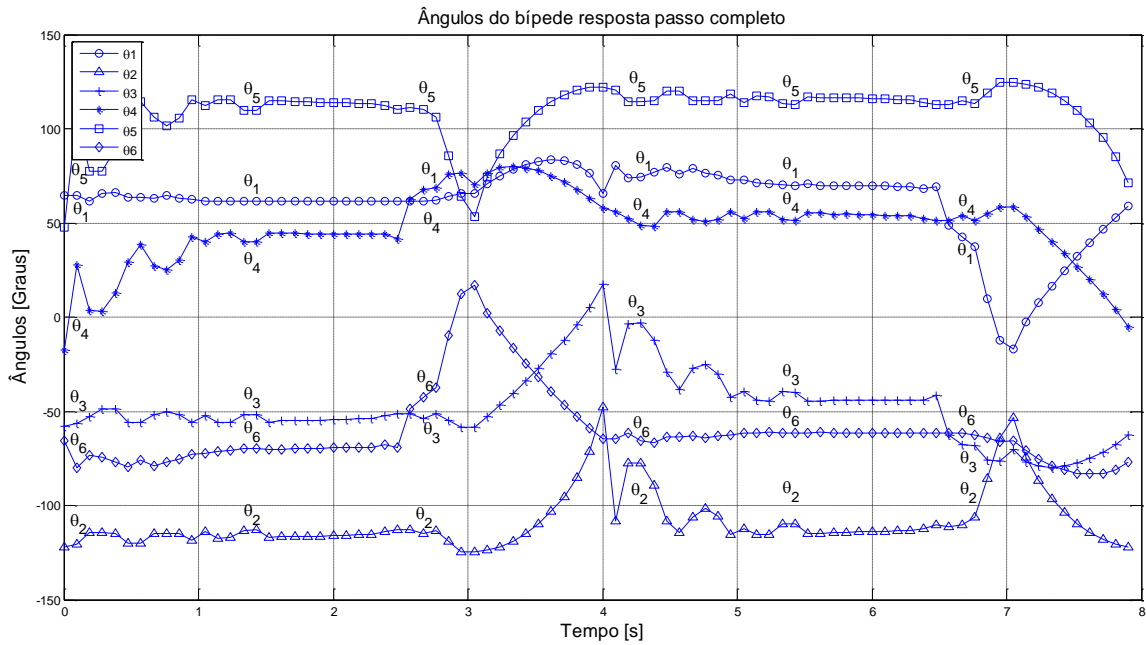


Figura 79. Ângulos motores, passo completo

A Figura 80 apresenta o incremento da energia consumida durante dois passos e a configuração do robô bípede em diversos instantes.

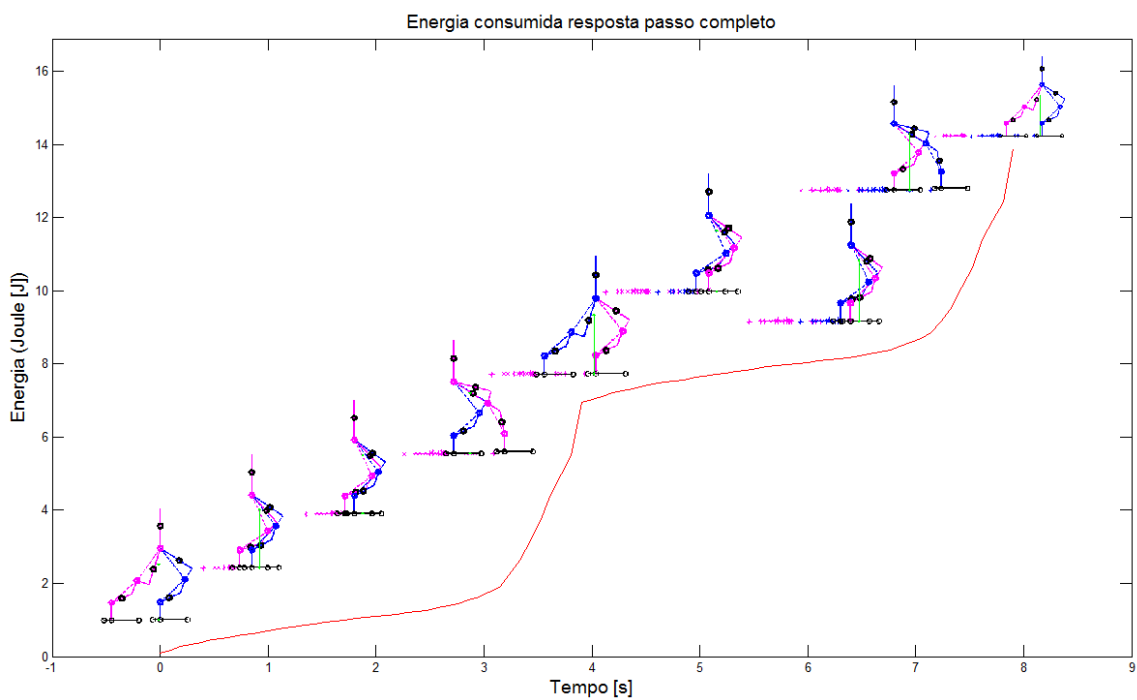


Figura 80. Energia consumida, passo completo

A energia consumida durante o período do padrão de marcha foi de 13,89J e a relação de energia sobre deslocamento foi de 35,2J/m.

A Figura 81 apresenta a evolução da aptidão do melhor indivíduo de cada geração. Selecionaram-se as gerações número 100, 500, 1000, 2000, 3000 e 3500 para observar a trajetória do pé.

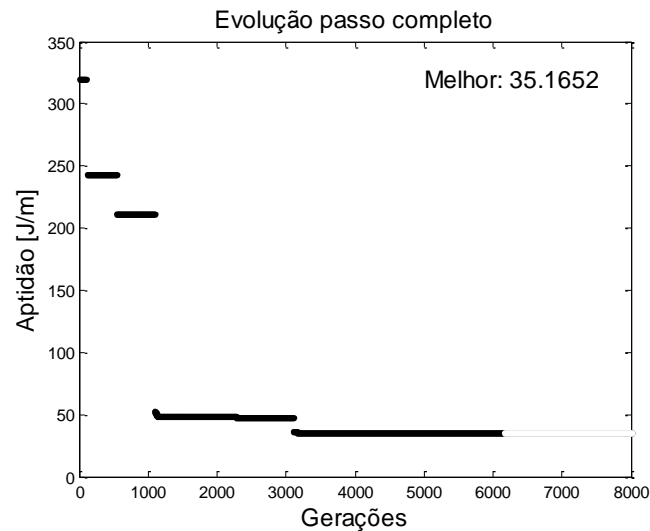


Figura 81. Evolução para passo completo.

A Figura 82 apresenta a evolução da trajetória do pé.

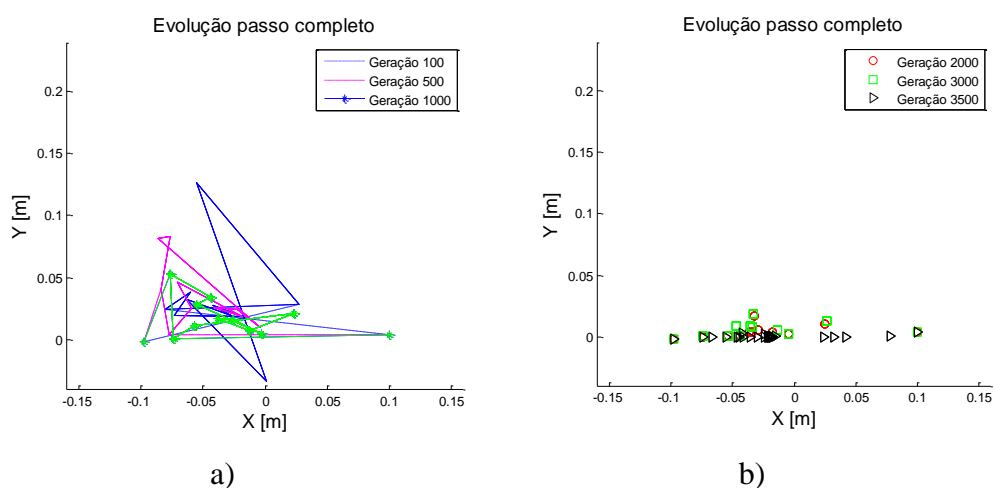


Figura 82. Evolução da trajetória do pé, para passo completo

Note que a resposta praticamente não levanta os pés do chão para economizar energia, quase que arrastando-os no solo. Fisicamente não é possível implementar essa resposta no robô bípede montado, pois ele tem suporte lateral nos pés, como mostra a Figura 67, o que produziria uma colisão entre os dois pés e eventual queda. Por esta razão, adicionou-se a penalização se o pé não avança na direção Y, seja positiva se é para subir, seja negativa se é para descer, vide Seção 3.3 “Penalizações nos Indivíduos”.

Passo completo B

Neste caso, as configurações penalizadas foram: se o pé não avança na direção Y positivo quando $X < 0$, se o pé não avança na direção Y negativa quando $X > 0$, se o pé não avança na direção X, se a posição da coordenada Y do pé é negativa, se o centro de massa está fora do espaço abrangido pelo pé de apoio, e se a posição Y do quadril sai fora da faixa de altura considerada. Os parâmetros do GA foram os mesmos anteriores. O tempo de cálculo computacional foi de 499.506s, aproximadamente 6 dias.

A Figura 83 apresenta a resposta da trajetória do pé em um passo para os ângulos fornecidos diretamente pelo GA (ver Seção 3.4.3 “Cromossomo passo completo”).

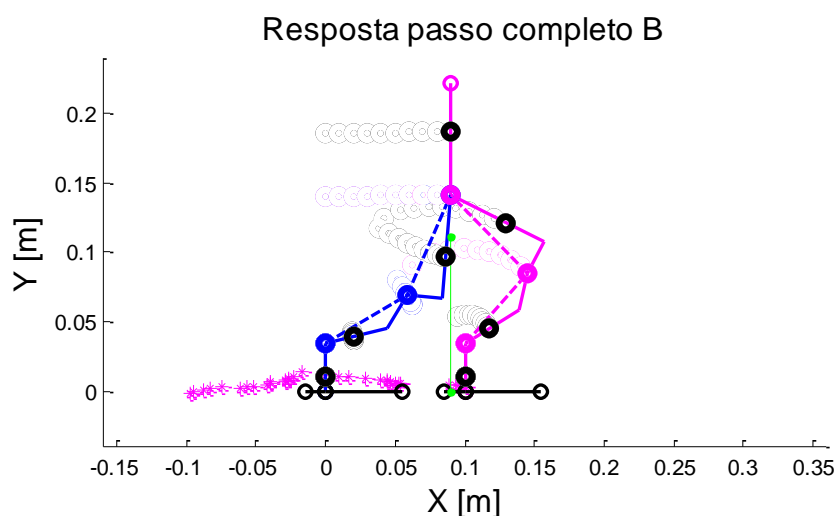


Figura 83. Resposta AG, passo completo B

A Figura 84 apresenta os ângulos dos seis motores durante 8s, dois passos equivalentes a um período do padrão de marcha.

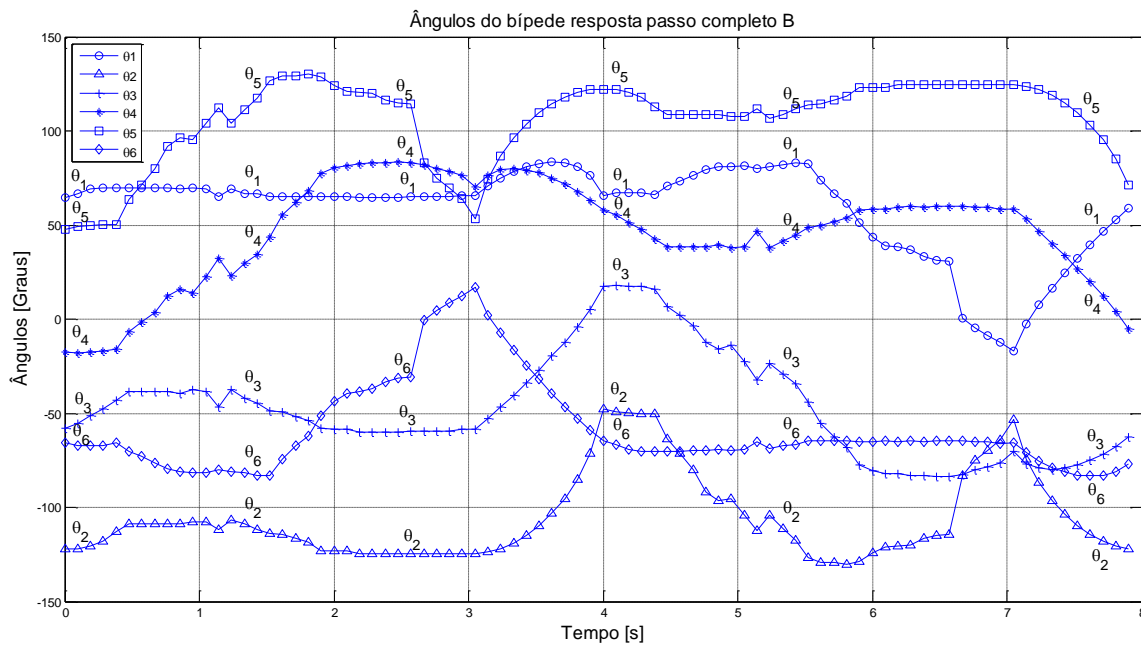


Figura 84. Ângulos motores, passo completo B

A Figura 85 apresenta o incremento da energia consumida durante dois passos e a configuração do robô bípede em diversos instantes.

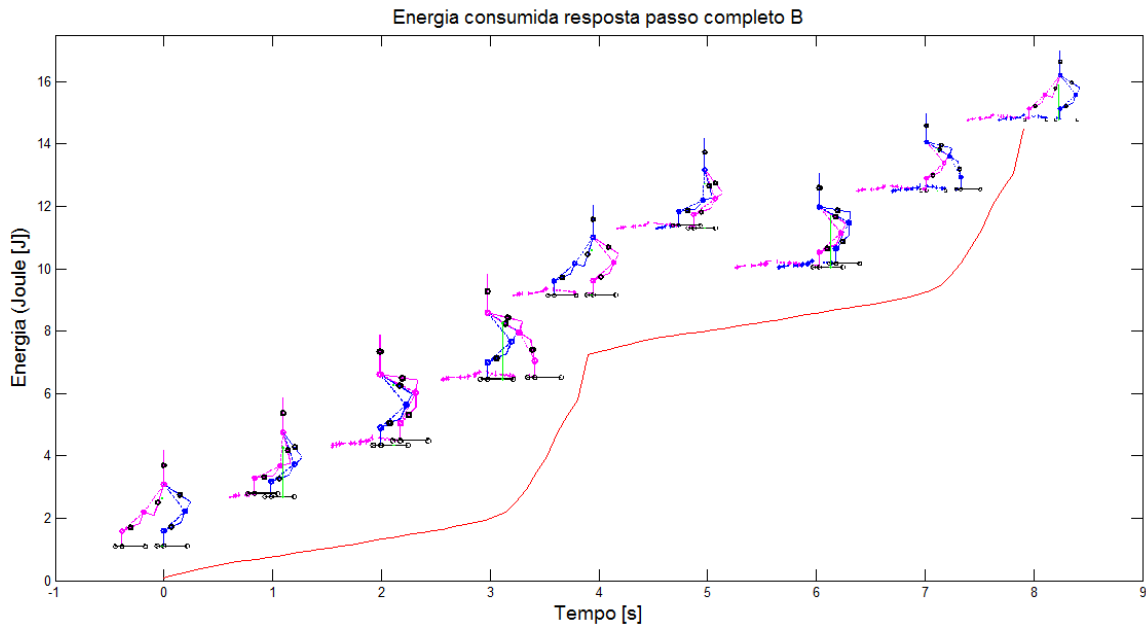
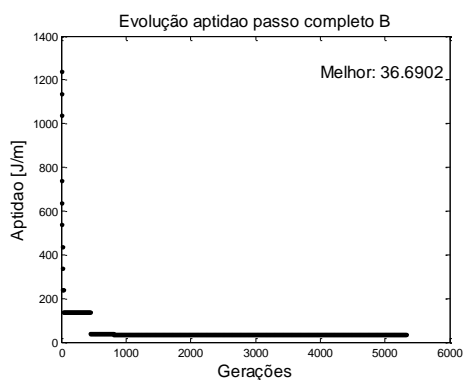


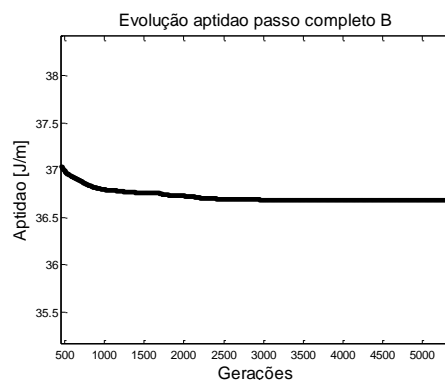
Figura 85. Energia consumida, passo completo B

A energia consumida durante o período do padrão de marcha foi de 14,5J e a relação de energia sobre deslocamento foi de 36,7J/m.

A Figura 86a) apresenta a evolução da aptidão do melhor indivíduo de cada geração, sendo a Figura 86b) um zoom entre os valores das gerações na faixa de 500 e 5000.



a)



b)

Figura 86. Evolução para o passo completo B

A Figura 87 apresenta a evolução da trajetória do pé.

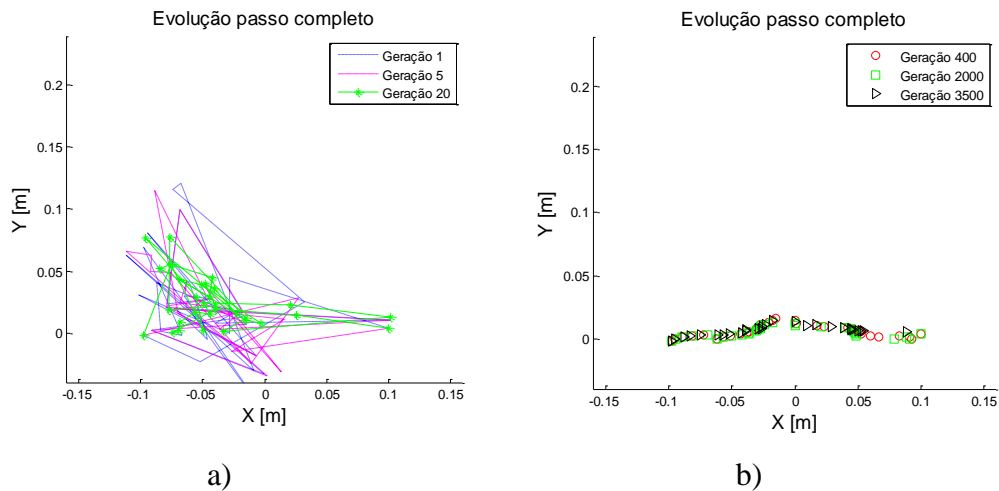


Figura 87. Evolução trajetória do pé, passo completo B

4.3.4. Sub-passos

Estudaram-se dois tipos de penalização no caso de sub-passos, gerando as respostas denominadas A e B (ver Seção 3.4.4 “Cromossomo sub-passo” e 3.3 “Penalizações nos Indivíduos”).

Resposta A

Nesse caso, as configurações penalizadas foram: se o pé não avança na direção X, se a posição da coordenada Y do pé é negativa, se o centro de massa está fora do espaço abrangido pelo pé de apoio, se a posição Y do quadril sai fora da faixa de altura considerada. Usaram-se os mesmos parâmetros de GA anteriores. O tempo de cálculo computacional foi de 13528s, aproximadamente 3 horas e 45 minutos.

A Figura 88 apresenta a resposta da trajetória do pé em um passo para os ângulos fornecidos pelo GA.

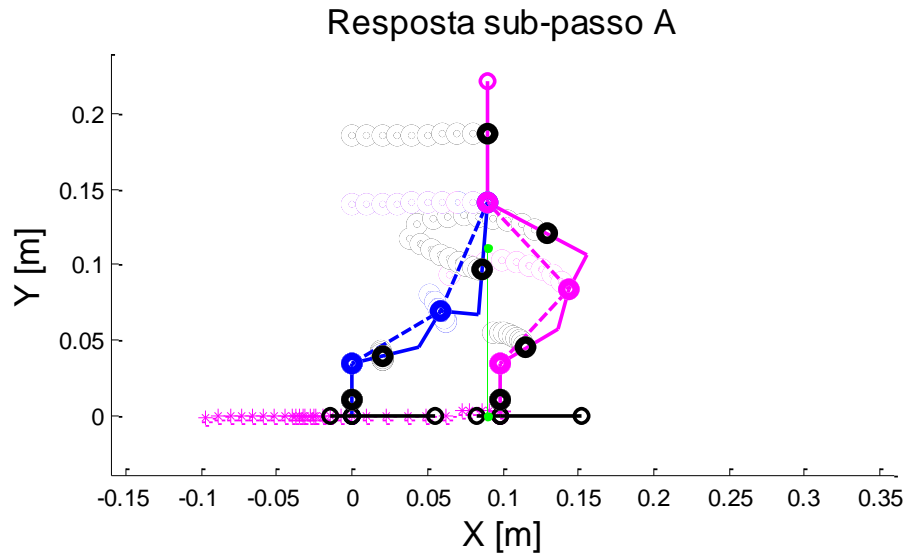


Figura 88. Resposta do GA, sub-passo resposta A

A Figura 89 apresenta os ângulos dos seis motores durante 8s, dois passos equivalentes a um período do padrão de marcha.

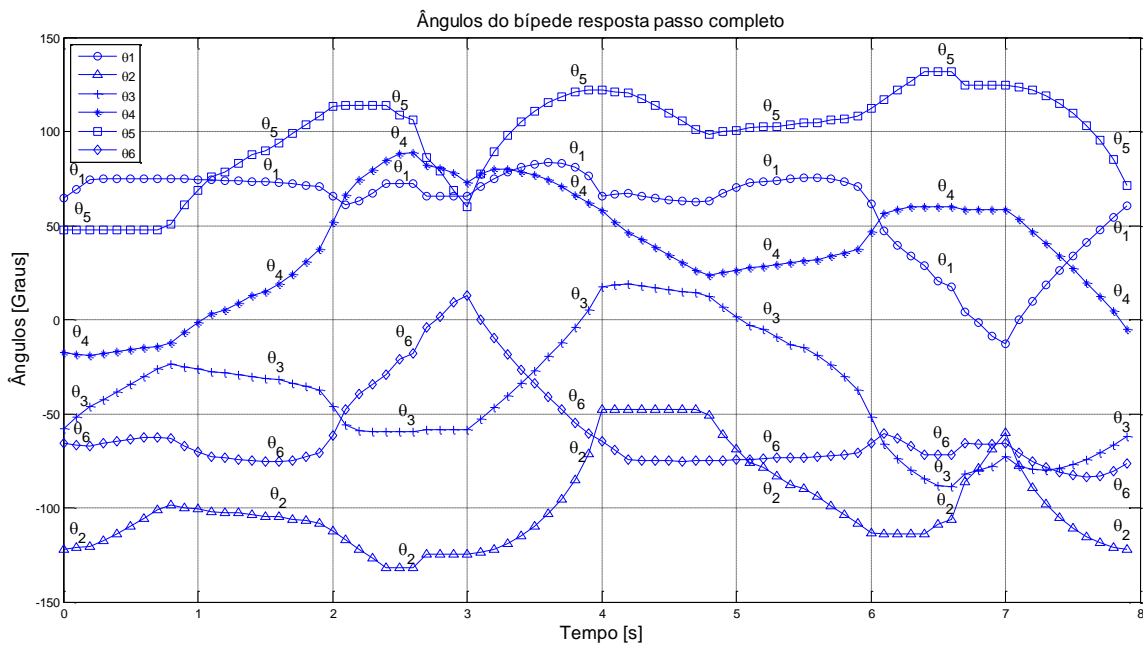


Figura 89. Ângulos dos motores, sub-passo resposta A

A Figura 90 apresenta o incremento da energia consumida durante dois passos, e a configuração do robô bípede em diversos instantes.

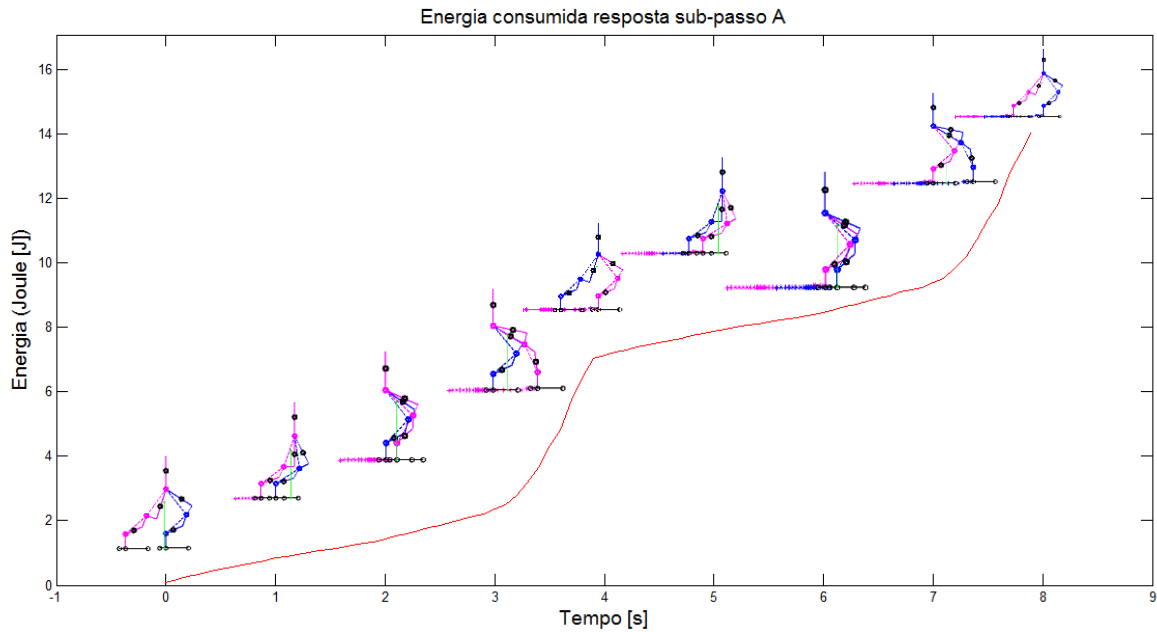


Figura 90. Energia consumida, sub-passo resposta A

A energia consumida durante o período do padrão de marcha foi de 14,08J a uma distância 0,391m, e a relação de energia sobre deslocamento foi de 36,05J/m.

A resposta de sub-passo não tem gráfico de evolução geral, já que é uma construção com evoluções sucessivas por cada fotograma. Fisicamente não é possível implementar a resposta já que o bípede final montado tem suporte lateral nos pés, como discutido anteriormente. Por este motivo adicionou-se uma punição se o pé não avança na direção Y, seja positiva se é para subir, seja negativa se é para descer, como descrito a seguir na chamada Resposta B.

Resposta B

Neste caso, as configurações penalizadas foram: se o pé não avança na direção Y positiva quando $X < 0$, se o pé não avança na direção Y negativa quando $X > 0$, se o pé não avança na direção X, se a posição da coordenada Y do pé é negativa, se o centro de massa está fora do espaço abrangido pelo pé de apoio, e se a posição Y do quadril sai fora da faixa de altura considerada. Os parâmetros do GA foram os mesmos. O tempo de cálculo computacional foi de 17.849s, aproximadamente 5

horas. A Figura 91 apresenta a resposta da trajetória do pé em um passo para os ângulos fornecidos pelo GA (ver Seção 3.4.4 “Cromossomo sub-passo”).

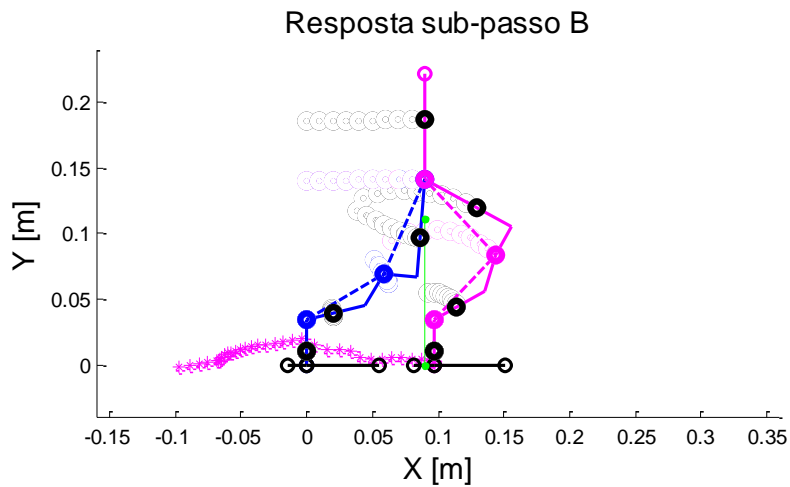


Figura 91. Resposta GA, sub-passo B

A Figura 92 apresenta os ângulos dos seis motores durante 8s, dois passos equivalentes a um período do padrão de marcha.

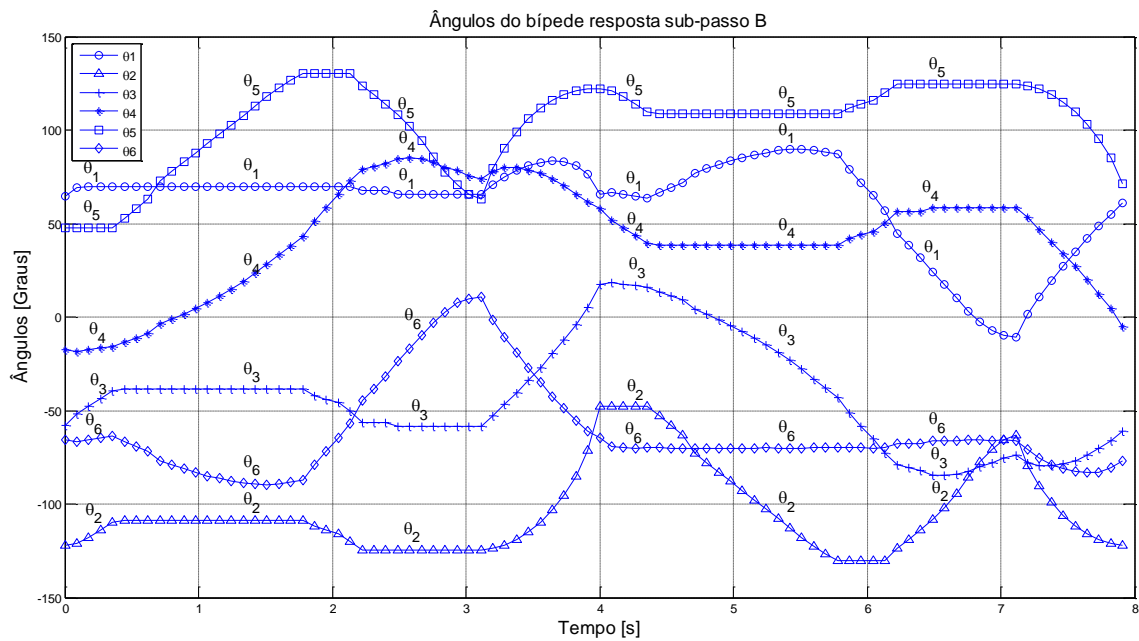


Figura 92. Ângulos motores, sub-passo B

A Figura 93 apresenta o incremento da energia consumida durante dois passos e a configuração do robô bípode em diversos instantes.

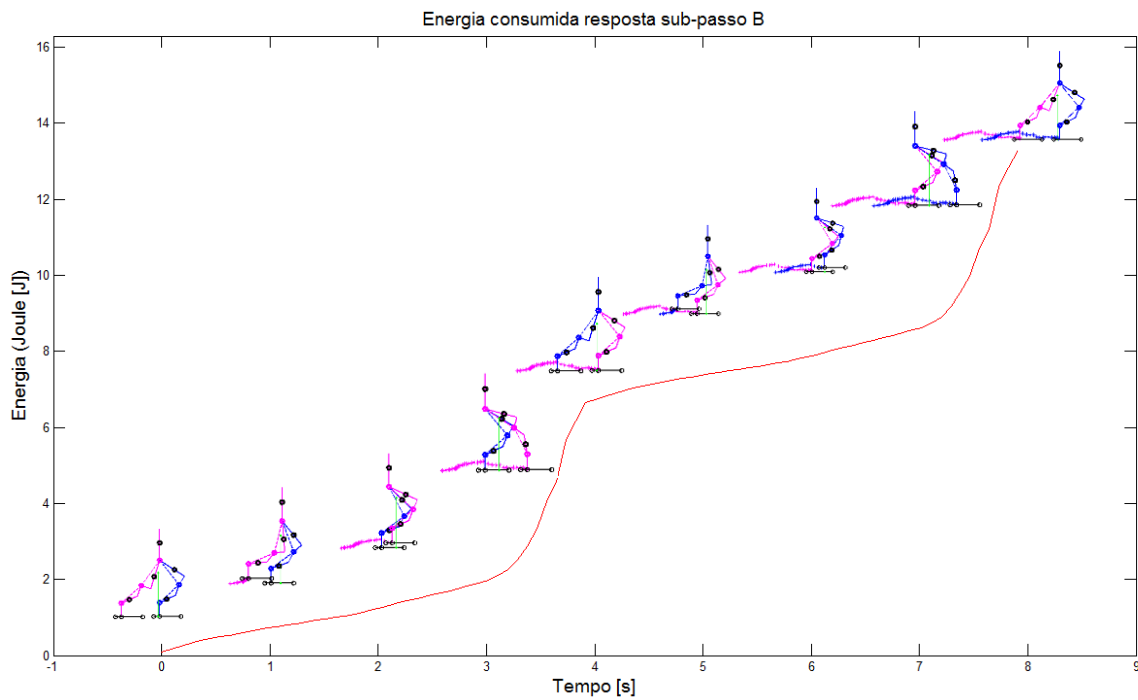


Figura 93. Energia consumida, sub-passo B

A energia consumida durante o período do padrão de marcha foi de 13,29J o deslocamento foi 0,388m, a relação de energia sobre deslocamento foi de 34,25 J/m.

4.3.5. Melhor indivíduo

A Figura 94 apresenta as respostas da trajetória do pé em um passo para os ângulos encontrados segundo os diversos métodos de GA propostos nesse trabalho.

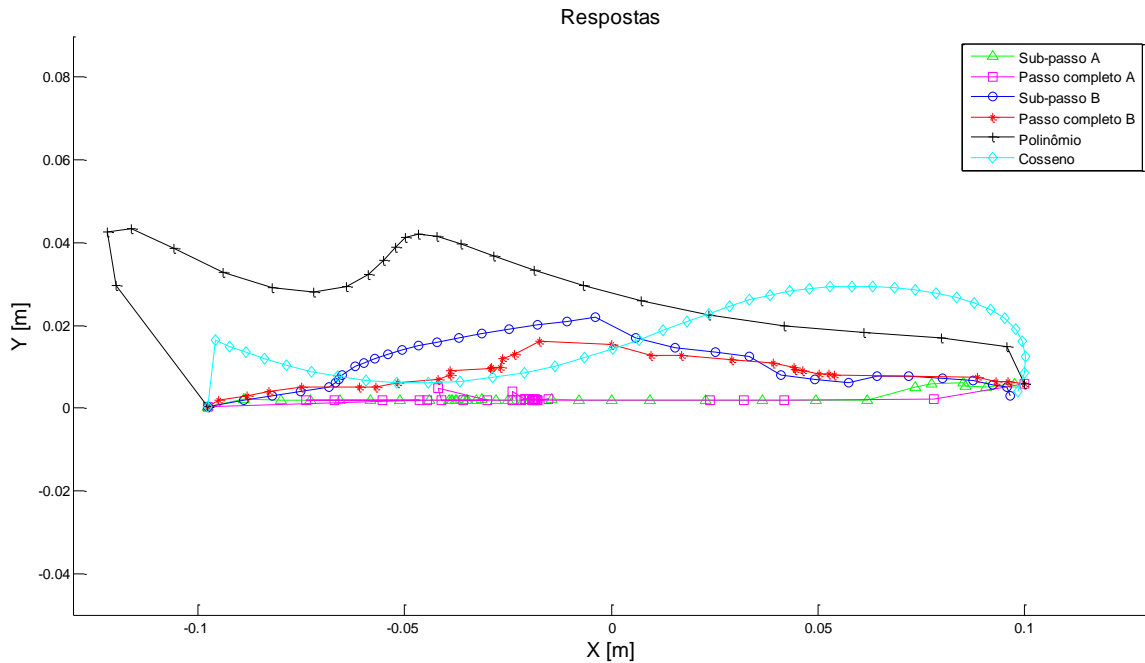


Figura 94. Trajetórias do pé, respostas por GA

A Tabela 9 apresenta o resumo dos dados das respostas. Dessa tabela pode-se comparar a relação energia sobre deslocamento e observar que a melhor resposta é para GA com sub-passo na resposta B, com *energia consumida de 13,29 J, deslocamento de 0,388 metros e relação energia sobre deslocamento de 34,25 J/m.*

Outra vantagem do enfoque com sub-passo foi o menor tempo de cálculo, se comparado com o método que aplica o GA ao passo completo.

Resposta	Fotogramas	Tempo de amostragem [s]	Deslocamento D [m]	Energia E [J]	Relação E/D	Tempo de cálculo
Heurística	21	0,19	0,39512	16,504 2	41,77	Não aplica
Polinômio	36	0,111	0,39512	15,607 9	39,5017	26 horas
Cosseno	51	0,078	0,39512	14,554 8	36,8366	9 horas 15 minutos
Passo completo A	42	0,095	0,39512	13,894 4	35,1652	7,5 dias
Sub-passo A	40	0,100	0,39052	14,079 8	36,0517	3 horas 45 minutos
Passo completo B	42	0,095	0,39512	14,497	36,6902	6 dias
Sub-passo B	45	0,089	0,38808	13,292 2	34,2515	5 horas

Tabela 9. Resumo, dados das respostas GA

5 Conclusões

Neste trabalho foi montado um protótipo de robô bípede planar, com a finalidade de reproduzir os ângulos para ele fornecidos e validar fisicamente a estabilidade da marcha quase-estática sobre uma superfície plana sem obstáculos.

Também se observou o caminhar de uma galinha como guia para obtenção de uma marcha quase-estática de modo heurístico; simularam-se em “Matlab” e “Solidworks”, implementaram-se os ângulos no protótipo e determinaram-se condições para o GA.

As simulações em “Matlab” foram construídas utilizando modelamento cinemático e estático.

Foram ajustados os parâmetros de controlador com o bloco “Pidcontroller” de “Simulink” e mediante algoritmos genéticos para malhas de controle cinemático, onde o melhor comportamento no ajuste dos parâmetros foi o Pid auto ajustado.

Geraram-se padrões de marcha com algoritmos genéticos visando quatro diferentes opções de configuração do cromossomo: indiretamente com o ajuste de funções polinomiais e trigonométricas; e diretamente com passo completo e sub-passo.

Como discutido anteriormente, dentre as 6 respostas obtidas pelos algoritmos genéticos, a melhor resposta foi a configuração de cromossomo sub-passo B, com uma *energia consumida de 13,29 J, um deslocamento de 0,388 m, e relação energia sobre deslocamento de 34,25 J/m*; em relação à caminhada heurística, a relação foi diminuída em 18%, e respeito a configuração de cromossomo polinômio, a relação diminuiu em 13,3%.

Outra vantagem do enfoque com sub-passo foi o menor tempo de cálculo, se comparado com o método que aplica o GA ao passo completo.

5.1. Sugestões para trabalhos futuros

Recomenda-se uma caracterização do motor AX-18 utilizado no robô bípede, que neste trabalho usou a referência de Mensink (2008) para o motor AX-12 [30].

Sugere-se também utilizar programação genética para ampliar o número de funções como gerador de padrões de marcha, permitindo ampliar as possíveis soluções e não limitar a funções conhecidas como polinômios e cossenos, as quais foram utilizadas nesta dissertação.

Trabalhos futuros podem também utilizar software especializado para simulação dinâmica 3D com detetor de colisões, como exemplo: Adams, Ode, Redysim, entre outros; e incluir uma caminhada dinâmica.

Recomenda-se utilizar controlador físico e interação com software para leitura de sensores e implementação de controle retroalimentado.

6 Referências bibliográficas

- [1] Defense Advanced Research Projects Agency., *DARPA*, [Online]. Available: <http://www.darpa.mil/about-us/about-darpa>. [Acesso em 27 08 2016].
- [2] Espectrum IEEE, *IEE Spectrum* [Online]. Available: <http://spectrum.ieee.org/automaton/robotics/humanoids/darpa-robotics-challenge-amazing-moments-lessons-learned-whats-next>. [Acesso em 27 08 2016].
- [3] Espectrum IEEE, *IEEE Espectrum* [Online]. Available: <http://spectrum.ieee.org/automaton/robotics/humanoids/drc-finals-course>. [Acesso em 27 08 2016].
- [4] Espectrum IEEE, *IEEE Espectrum* [Online]. Available: <http://spectrum.ieee.org/automaton/robotics/humanoids/how-kaist-drc-hubo-won-darpa-robotics-challenge>. [Acesso em 27 08 2016].
- [5] National Aeronautics and Space Administration NASA, *NASA* [Online]. Available: <http://www.nasa.gov/feature/nasa-looks-to-university-robotics-groups-to-advance-latest-humanoid-robot>.
- [6] Boston Dynamics, *BostonDynamics* [Online]. Available: http://www.bostondynamics.com/robot_Atlas.html. [Acesso em 27 08 2016].
- [7] Boston Dynamics, *BostonDynamics* [Online]. Available: http://www.bostondynamics.com/robot_petman.html. [Acesso em 27 08 2016].
- [8] *American Honda Motor Co. Inc.* , Asimo, [Online]. Available: <http://asimo.honda.com/>. [Acesso em 27 08 2016].

- [9] Cable News Network, *CNN Money Internacional* [Online]. Available: <http://money.cnn.com/2013/12/16/technology/google-boston-dynamics-robots/index.html>. [Acesso em 27 08 2016].
- [10] NBC, *NBC News*, [Online]. Available: <http://www.nbcnews.com/tech/tech-news/google-s-next-generation-atlas-robot-turns-heads-n525061>. [Acesso em 27 08 2016].
- [11] Waseda University, *Waseda University Humanoid* [Online]. Available: http://www.humanoid.waseda.ac.jp/booklet/kato_2.html. [Acesso em 27 08 2016].
- [12] S. Kajita, *Introduction to humanoid robotics*, Springer, 2014.
- [13] Nakaoka, A. Nakazawa, K. Yokoi, H. Hirukawa e K. Ikeuchi, *Generating whole body motions for a biped humanoid robot from captured human dances.*, *IEEEExplore*, 2003.
- [14] K. Yamane, *Simulating and Generating Motions of Human Figures.*, Springer, 2004.
- [15] J. Kuffner e S. LaValle, RRT-connect: An efficient approach to single, *IEEEExplore*, 200.
- [16] K.-S. Hwang, J.-L. Lin e a. K.-H. Yeh, Learning to Adjust and Refine Gait, Patterns for a Biped Robot, *IEEEExplore*, vol. 45, nº 12, 2015.
- [17] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal and M. Kawato, Learning from demonstration and adaptation of biped locomotion, *Robotics and Autonomous Systems* 47, pp. 79-71, 2004.
- [18] B. I. Kazem, A. I. Mahdi e A. T. Oudah, Motion Planning for a Robot Arm by Using Genetic Algorithm, *Jordan Journal of Mechanical and Industrial Engineering*, vol. 2, nº 3, pp. 131-136, 2008.
- [19] S. Števo, I. Sekaj e M. Dekan, Optimization of Robotic Arm Trajectory Using Genetic Algorithm, *The International Federation of Automatic Control Cape Town*, pp. 24-29, 2014.
- [20] T. ARAKAWA e T. FUKUDA, Natural Motion Trajectory Generation of Biped Locomotion Robot using Genetic Algorithm, *IEEEExplore*, 1996.

- [21] J. Rose e J. G. Gamble, *Human Walking*, Gamble, 1994.
- [22] J. Sánchez Lacuesta, *Biomecánica de la Marcha Humana Normal y Patológica*, Valencia: Instituto de Biomecánica de Valencia, 2006.
- [23] D. Zaldivar, E. Cuevas e R. Rojas, *Bipedal robot description* 2005.
- [24] Robotis, *Robotis* [Online]. Available: <http://support.robotis.com/en/>. [Acesso em 27 08 2016].
- [25] Robotis, *Robotis e-manual* Robotis, [Online]. Available: <http://support.robotis.com/en/>. [Acesso em 27 08 2016].
- [26] Asada e Stoline, *Robot Analysis and Control*, 1985.
- [27] The Regents of the University of Michigan, *Control tutorials for matlab & simulink*, University of Michigan, [Online]. Available: <http://ctms.engin.umich.edu/CTMS/index.php?example=MotorPosition§ion=ControlPID>. [Acesso em 27 08 2016].
- [28] F. Monasterio Huelin, *Motor DC, etapa de potencia y PWM*, 2016.
- [29] Electro craf corporation, *Dc motors speed controls servo systems, an engineering handbook*, Pergamon Press, 1977.
- [30] A. Mensink, *Characterization and modeling of a Dynamixel servo*, 2008.
- [31] Mathematical computing software for engineers and scientists., *Mathworks*, [Online]. Available: <http://www.mathworks.com>. [Acesso em 27 08 2016].
- [32] K. Passino e N. Quijano, *Proportional-Integral-Derivative Control with Derivative Filtering and Integral Anti-Windup for a DC Servo*, 2002.
- [33] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.
- [34] C. Darwing, *On the origen of the species*, London: Jonh Murray, 1859.
- [35] M. Aurelio Pacheco, *ICA: Laboratório de Inteligência Computacional Aplicada*, [Online]. Available: www.ica.ele.puc-rio.br.

- [36] L. Chambers, *Practical handbook of genetic algorithms*, 1995.
- [37] T. Blicke, *Theory of Evolutionary Algorithms and Application to System Synthesis*, Zúrique, 1996.
- [38] EIROforum, *scienceinschool*, EIROforum, [Online]. Available: <http://www.scienceinschool.org/es/2011/issue21/ostrich>. [Acesso em 27 08 2016].
- [39] National Geographic Society. , National Geographic Kids, *National Geographic Society*. , [Online]. Available: <http://kids.nationalgeographic.com/animals/ostrich/#ostrich-grass.jpg>. [Acesso em 27 08 2016].
- [40] The MathWorks, Inc., Mathwork Documentation, The *MathWorks, Inc.*, 2016. [Online]. Available: <http://www.mathworks.com/help/control/ug/two-degree-of-freedom-2-dof-pid-controllers.html>. [Acesso em 20 09 2016].
- [41] The MathWorks, Inc., MathWorks Documentation, The *MathWorks, Inc.*, 2016. [Online]. Available: <http://www.mathworks.com/help/slcontrol/ug/pid-tuning-algorithm.html>.
- [42] R. Victor M. Alfaro, Sintonización de los controladores pid de 2gdl: desempeño, robustez y fragilidad, *Escuela de ingeniería eléctrica, universidad de Costa Rica*, 2016.
- [43] H. Astrom K, Automatic tuning of simple regulators with specification on phase and amplitude margins, *Automatic*, pp. 645-651, 1984.

Apêndice A Desenhos SolidWorks

Na Figura 95 pôde-se observar o robô humanoide do kit “Bioloid GP”, do qual utilizaram-se peças para montar o robô bípede planar tratado neste documento, os desenhos do motor e acessórios foram baixados da web do fabricante [24] (Robotis – suport – downloads - Links to Drawings - Dynamixel, Bioloid, Frames), mas outros elementos estão indisponíveis, pelo que foi necessário desenha-los. A Figura 96 mostra o desenho.



Figura 95. Robô Bioloid GP

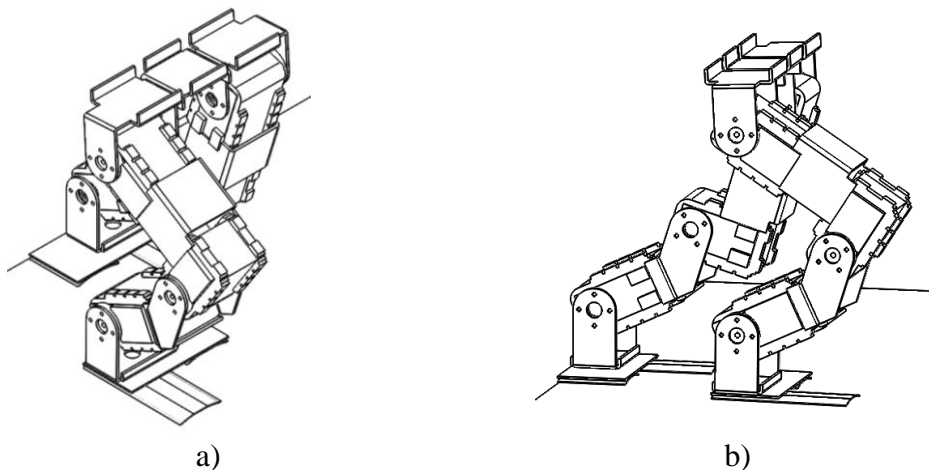


Figura 96. Robô bípede planar, desenho Solidworks

A Tabela 10 mostra a montagem da representação de cada elo do bípede e a localização do centro de massa (cm).

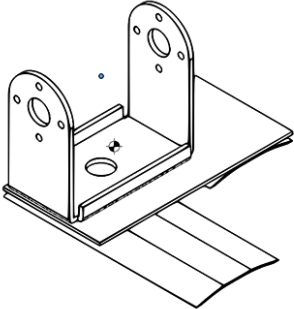
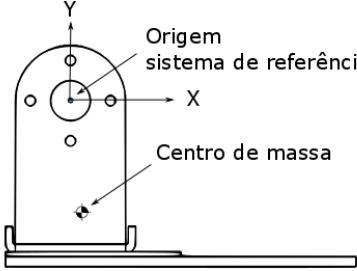
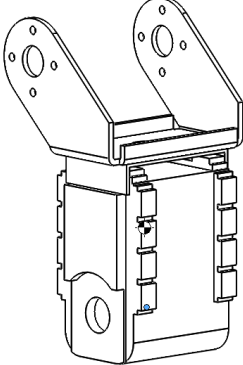
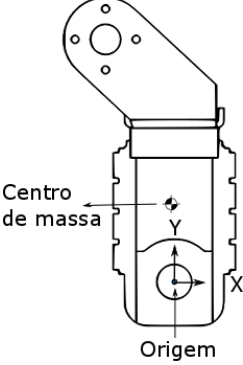
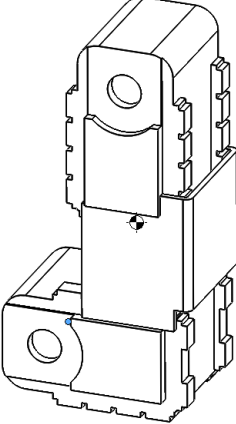
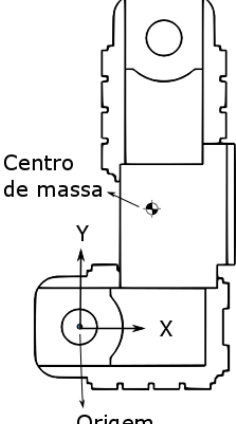
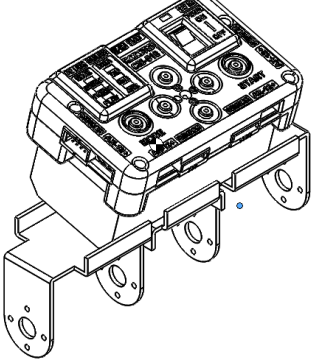
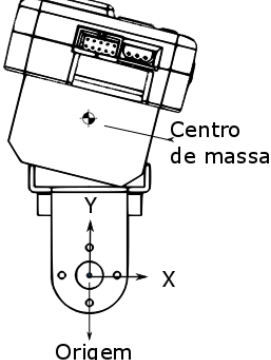
	 <p>Origem sistema de referência</p> <p>X</p> <p>Centro de massa</p> <p>Y</p>	<p>Base pé</p> <p>Centro de massa:</p> <p>$X = -2.55\text{mm}$</p> <p>$Y = 22.17\text{mm}$</p> <p>$Z = -2.29\text{mm}$</p> <p>Massa: 15,43g</p>
a) Base pé, desenho	b) Base pé, localização cm	c) Dados cm
	 <p>Centro de massa</p> <p>Y</p> <p>X</p> <p>Origem sistema de referência</p>	<p>Tíbia</p> <p>Centro de massa:</p> <p>$X = -2.55\text{mm}$</p> <p>$Y = 22.17\text{mm}$</p> <p>$Z = -2.29\text{mm}$</p> <p>Massa: 15,43g</p>
a) Tíbia, desenho	b) Tíbia, localização cm	c) Dados cm
	 <p>Centro de massa</p> <p>Y</p> <p>X</p> <p>Origem sistema de referência</p>	<p>Fêmur</p> <p>Centro de massa:</p> <p>$X = -18.70\text{mm}$</p> <p>$Y = -30.42\text{mm}$</p> <p>$Z = 0.16\text{mm}$</p> <p>Massa: 120,86g</p>
a) Fêmur, desenho	b) Fêmur localização cm	c) Dados cm
	 <p>Centro de massa</p> <p>Y</p> <p>X</p> <p>Origem sistema de referência</p>	<p>Tronco</p> <p>Centro de massa:</p> <p>$X = -40.19$</p> <p>$Y = 45.35$</p> <p>$Z = -0.18$</p> <p>Massa: 172,2g</p>
a) Tronco, desenho	b) Tronco Localização cm	c) Dados cm

Tabela 10. Elos robô bípede

A Tabela 11 apresenta as dimensões em milímetros das peças utilizadas na montagem do bípede. Os dados do motor são mostrados no apêndice B.

FR04-H101	FR04-S101
FR04-H102	FR04-SC110
FR04-HC110	FR04-SC101

Tabela 11. Dimensões peças robô bípede

Apêndice B. Dados motor Dynamixel AX-18A

- Peso: 55.9g
- Dimensões: 32mm * 50mm * 40mm
- Resolução: 0.29°
- Relação de redução: 254:1
- Torque parado: 1.8N.m (12.0V, 2.2A)
- Velocidade sem carga: 97rpm (12V)
- Grau de rotação: 0° ~ 300°, ou giro contínuo
- Faixa de temperatura: -5°C ~ +75°C
- Tensão: 9 ~ 12V (Tensão recomendada 11.1V)
- Comando de sinal: Pacote digital
- Tipo de protocolo: Comunicação serial assíncrona semi-duplex (8bit, 1parada, não paridade)
- ID : 254 ID (0~253)
- Velocidade de comunicação: 7343bps ~ 1 Mbps
- Retroalimentação: Posição, temperatura, torque, tensão.
- Controle: PID
- Material: Plástico de engenharia.
- Controladores: ArbotiX-M, Robotis CM-530, Robotis USB2Dynamixel, Robotis CM-700, Robotis Open CM 9.
- Ambiente de controle: software proprietário Roboplus; bibliotecas C/C++, C#, Labview, Matlab, Visual Basic.

Torque parado é o máximo torque estático e instantâneo. Movimentos estáveis são possíveis com robôs projetados para cargas de 1/5 ou menos do que o torque parado.

A Tabela 12 apresenta o desenho e a localização do centro de massa

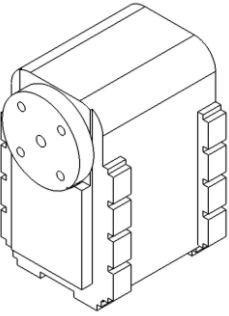
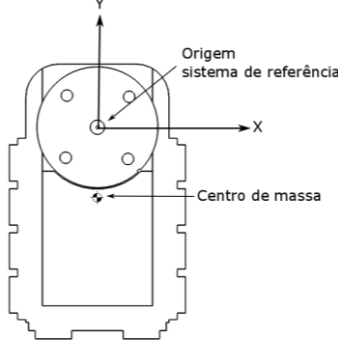
	 <p>Origem sistema de referência</p> <p>X</p> <p>Centro de massa</p> <p>Y</p>	<p>Motor</p> <p>Centro de massa:</p> <p>X = -0.00mm</p> <p>Y = -12.79mm</p> <p>Z = 0.69mm</p> <p>Massa: 55,9g</p>
<p>a) Desenho motor AX-18A</p>	<p>b) Localização cm</p>	<p>c) Dados</p>

Tabela 12. Motor AX-18A

Os dados de localização do centro de massa são aproximados, já que não se tem razão da distribuição dos componentes internos.

A Figura 97 apresenta as dimensões do motor AX-18A

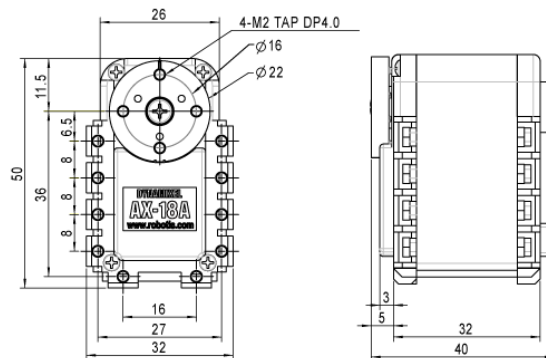


Figura 97. Motor AX-18A

<http://support.robotis.com/en/>

Apêndice C. Principais linhas de código

Apresenta-se as principais linhas de código no Matlab

Guardar o melhor indivíduo de cada geração Algoritmo genético

Abrir a função “gaplotbestf”, copia-la e cola-la para guardar como uma nova.

Adicionar as linhas de código:

No início da função

```
global NomeVariavelGlobal %(declarar a variável também no programa principal)
```

Dentro dos casos do “switch flag”

```
[Fvalmg,i] = min(state.Score);
Temporal = [state.Generation, state.Population(i,:), Fvalmg,
meanf(state.Score)];%(geração, melhor, aptidão, meia)
NomeVariavelGlobal = [NomeVariavelGlobal;temporal];
```

Declaração da variável global

```
global NomeVariavelGlobal
```

Limite inferior e superior das variáveis

```
LB=[LinfVar1 LinfVar2 LinfVarx];
UB=[LsupVar1 LsupVar2 LsupVarx];
```

Declaração função objetivo (u: vector de variáveis para ajustar)

```
fitness = @(u) NomeFunçãoObjetivo(DadoEnt1, DadoEntN,u);
```

Configuração das opções

```
Options =
gaoptimset('PopulationSize', númeroindividuos, ...
'Generations', Gerações, ...
'StallGenLimit', ceil(Gerações *2/3), ...
'PopInitRange', [LB;UB], ...
'SelectionFcn', @selectionroulette, ...
'CrossoverFcn', @crossovertwopoint, ...
'Display', 'off', ...
'MutationFcn', {@mutationadaptfeasible PorcentagemMutaçao}, ...
'CrossoverFraction', PorcentagemCrossover, ...
'UseParallel', 'always', ...
'TolFun', 1e-3, ...
'PlotFcns', {@gaplotbestfNOVA});
[Saidas]= ga(fitness,numVar, [], [], [], [], LB,UB, [], options);
```


Salvar os dados como estrutura

```

NumPasta=1;
FolderName = ['NomedaPasta_' num2str(NumExperimento)];
mkdir(FolderName);
NomeEstrutura.bestsolution= Saidal %(melhor individuo final);
NomeEstrutura.Generaciones= Gerações;
NomeEstrutura.PoblacionTamano= númeroindividuos;
NomeEstrutura.HistoricoDoMelhor=NomeVariavelGlobal;
NomeEstrutura.time= toc;
save([FolderName '\'NomedaPasta_'
num2str(NumExperimento), '.mat'], 'NomeEstrutura')

```

Conversão de ângulos

```

sizedeta4mot=size(teta4mot); %(matriz de k fotografamas por 4
ângulos)
nfotogramas=sizedeta4mot(1);
teta1rfg=(90-teta4mot(:,1));
teta2rfg=-teta4mot(:,2);
teta3rfg=-(180-(tetacorpo+180-(teta4mot(:,1)+teta4mot(:,2))));
teta4rfg=-((tetacorpo+180-(teta4mot(:,1)+teta4mot(:,2)))-
teta4mot(:,3));
teta5rfg=-teta4mot(:,4);
teta6rfg=-(270-
(teta4mot(:,1)+teta4mot(:,2)+teta4mot(:,3)+teta4mot(:,4)));
tetarfg=[teta1rfg, teta2rfg, teta3rfg, teta4rfg, teta5rfg,
teta6rfg];
tetasim6mot=[90*ones(nfotogramas,1) -teta1rfg
45*ones(nfotogramas,1) (-135+-teta2rfg) 90*ones(nfotogramas,1)
teta3rfg 180*ones(nfotogramas,1) teta4rfg -90*ones(nfotogramas,1)
(135+-teta5rfg) -45*ones(nfotogramas,1) -teta6rfg];
tetarobotfisicog=[teta1rfg teta2rfg teta3rfg teta4rfg teta5rfg
teta6rfg];
tetarobotfisicoDig=round((tetarobotfisicog*3.41)+511.5);
tetaDigrealfisico=[tetarobotfisicoDig(:,3),
tetarobotfisicoDig(:,2), tetarobotfisicoDig(:,1),
tetarobotfisicoDig(:,4) tetarobotfisicoDig(:,5)
tetarobotfisicoDig(:,6)];

```

Cinemática direta

```

A0ai=zeros(4,4,nelos+1,nfoto); %zerar matrizes
for j=1:nfoto
    A0ai(:, :, 1, j)=eye(4);
end
A0aLcmi=zeros(4,4,nelos+1,nfoto);
for j=1:nfoto
    A0aLcmi(:, :, 1, j)=eye(4);
end
for j=1:nfoto
    %Cinemática dos centros de massa
    for i=1:nelos
        AimlaLcmi(:, :, i, j)=[cos(teta(j,i)) -sin(teta(j,i))
0    Lcm(i)*cos(teta(j,i));
                                sin(teta(j,i))  cos(teta(j,i))
0    Lcm(i)*sin(teta(j,i));
                                0                0
1    0;
                                0                0
0    1];

```

```

                                % Cinemática dos elos
                                Aim1ai(:, :, i, j) = [cos(teta(j, i)) -sin(teta(j, i))
0    L(i)*cos(teta(j, i));                                sin(teta(j, i))  cos(teta(j, i))
0    L(i)*sin(teta(j, i));                                0                0
1    0;                                                    0                0
0    1];
                                end
                                for i=2:nelos+1
                                    A0ai(:, :, i, j)=A0ai(:, :, (i-1), j)*Aim1ai(:, :, (i-1), j);
                                end
                                A0aicop=A0ai;
                                for i=2:nelos+1
                                    A0aLcmi(:, :, i, j)=A0aicop(:, :, i-1, j)*Aim1aLcmi(:, :, i-
1, j);
                                end
                                summpos=0;                % Centro de massa
                                for i=1:nelos+1
                                    summpos=A0aLcmi(1:3, 4, i, j)*masa(i)+summpos;
                                end
                                sumpos=A0ai(1:3, 4, 9, j)*masa(10);
                                cm(:, j)=(summpos+sumpos)/sum(masa);
                                end
                                end

tetalcal4mot=(90*pi/180)-tetalrf6mot;
teta2cal4mot=-teta2rf6mot;
tetacorpo=tetalcal4mot+teta2cal4mot+teta3rf6mot;%%% Rad
tetacorpogrados=tetacorpo*180/pi;
alfalcal=tetalcal4mot+(17.0028*pi/180);
alfa2cal=teta2cal4mot-(35.4219*pi/180);
J=[-L1*sin(alfalcal)-L2*sin(alfalcal+alfa2cal) -
L2*sin(alfalcal+alfa2cal) 0;
L1*cos(alfalcal)+L2*cos(alfalcal+alfa2cal)
L2*cos(alfalcal+alfa2cal) 0;
1                1                1];
                                % velocidades lineares do extremo
xe=L1*cos(alfalcal)+L2*cos(alfalcal+alfa2cal);
ye=L1*sin(alfalcal)+L2*sin(alfalcal+alfa2cal);
velvelang=J*[alfa1pto; alfa2pto; alfa3pto];
xepto=velvelang(1);
yepto=velvelang(2);
tetacorpopto=velvelang(3);%%% Rad/s

```

Cinemática inversa

```

% disp('Radianos. Filal: cotovelo para acima, fila2: cotovelo para
abaixo')
beta=atan2(ye, xe);
betag=beta*180/pi;
r=sqrt((xe^2)+(ye^2));
xcomp=r*cos(beta);
ycomp=r*sin(beta);
alfacodoarriba=acos(((L2^2)-(L1^2)-(r^2))/(-2*L1*r));
alfagcodoarriba=alfacodoarriba*180/pi;
alfacodoabajo=-acos(((L2^2)-(L1^2)-(r^2))/(-2*L1*r));
alfagcodoabajo=alfacodoabajo*180/pi;
tetalinvcodoarriba=beta+alfacodoarriba;
tetalinvgcodoarriba=tetalinvcodoarriba*180/pi;

```

```

tetalinvcodoabajo=beta+alfacodoabajo;
tatalinvcgodoabajo=tetalinvcodoabajo*180/pi;
xlcodigoarriba=L1*cos(tetalinvcodoarriba);
ylcodigoarriba=L1*sin(tetalinvcodoarriba);
deltaxcodigoarriba=xe-xlcodigoarriba;
deltaycodigoarriba=ye-ylcodigoarriba;
tetalmasteta2codoarriba=atan2(deltaycodigoarriba,deltaxcodigoarriba);
teta2invcodoarriba=tetalmasteta2codoarriba-tetalinvcodoarriba;
teta2invcodoarribag=teta2invcodoarriba*180/pi;
teta2mirar=alfagrado(1,2)+35.42; %%% teta 2 deve ser positivo
para cumprir configuração perna humana

if teta2mirar>0
    tetalcal4mot=alfagrado(1,1)-17;
    teta2cal4mot=alfagrado(1,2)+35.42;
    alfa1cal=alfagrado(1,1)*pi/180; %%% Radianos
    alfa2cal=alfagrado(1,2)*pi/180; %%% Radianos
else
    tetalcal4mot=alfagrado(2,1)-17;
    teta2cal4mot=alfagrado(2,2)+35.42;
    alfa1cal=alfagrado(2,1)*pi/180; %%% Radianos
    alfa2cal=alfagrado(2,2)*pi/180; %%% Radianos
end

tetal1rf6mot=(90-tetalcal4mot)*pi/180; %%% Radianos
teta2rf6mot=(-teta2cal4mot)*pi/180; %%% Radianos
teta3rf6mot=(tetacorporo-tetalcal4mot-teta2cal4mot)*pi/180; %%%
Radianes
tetacorporoptorad=tetacorporopto*pi/180; %%% Rad/s
J=[-L1*sin(alfa1cal)-L2*sin(alfa1cal+alfa2cal) -
L2*sin(alfa1cal+alfa2cal) 0;
L1*cos(alfa1cal)+L2*cos(alfa1cal+alfa2cal)
L2*cos(alfa1cal+alfa2cal) 0;
1 1 1];
Jinv=inv(J);

alfapto=Jinv*[xepto;yepto;tetacorporoptorad];
alfalpto=alfapto(1);
alfa2pto=alfapto(2);
alfa3pto=alfapto(3); %%% Rad/s

```

Gráficos

```

for j=1:nfoto %Gráficos elos
    for i=1:5
        if Der==1
            helo(i,j)=line([A0ai(1,4,i,j)
A0ai(1,4,i+1,j)], [A0ai(2,4,i,j) A0ai(2,4,i+1,j)], [A0ai(3,4,i,j)
A0ai(3,4,i+1,j)], 'color','b','LineWidth',2);
        else
            helo(i,j)=line([A0ai(1,4,i,j)
A0ai(1,4,i+1,j)], [A0ai(2,4,i,j) A0ai(2,4,i+1,j)], [A0ai(3,4,i,j)
A0ai(3,4,i+1,j)], 'color','m','LineWidth',2);
        end
    end
    for i=6:nelos
        if Der==1
            helo(i,j)=line([A0ai(1,4,i,j)
A0ai(1,4,i+1,j)], [A0ai(2,4,i,j) A0ai(2,4,i+1,j)], [A0ai(3,4,i,j)
A0ai(3,4,i+1,j)], 'color','m','LineWidth',2);
        end
    end
end

```

```

        else
            helo(i,j)=line([A0ai(1,4,i,j)
A0ai(1,4,i+1,j)], [A0ai(2,4,i,j) A0ai(2,4,i+1,j)], [A0ai(3,4,i,j)
A0ai(3,4,i+1,j)], 'color','b','LineWidth',2);
        end
    end

    hold on %gráfica Pés
    helo(nelos+1,j)=line([A0ai(1,4,1,j)
A0ai(1,4,1,j)+Lapied], [0 0], [0
0], 'color','k','LineWidth',2,'Marker','o');
    helo(nelos+2,j)=line([A0ai(1,4,1,j) A0ai(1,4,1,j)-
Lapiet], [0 0], [0 0], 'color','k','LineWidth',2,'Marker','o');
    helo(nelos+3,j)=line([A0ai(1,4,nelos+1,j)
A0ai(1,4,nelos+1,j)+Lapied], [A0ai(2,4,nelos+1,j)
A0ai(2,4,nelos+1,j)], [0
0], 'color','k','LineWidth',2,'Marker','o');
    helo(nelos+4,j)=line([A0ai(1,4,nelos+1,j)
A0ai(1,4,nelos+1,j)-Lapiet], [A0ai(2,4,nelos+1,j)
A0ai(2,4,nelos+1,j)], [0
0], 'color','k','LineWidth',2,'Marker','o');

    if Der==1 %gráfica elos simplificados
    cinemática
        felo(1)=line([A0ai(1,4,2,j)
A0ai(1,4,4,j)], [A0ai(2,4,2,j) A0ai(2,4,4,j)], [0
0], 'color','b','LineWidth',2,'Marker','o','LineStyle','--');
        felo(2)=line([A0ai(1,4,4,j)
A0ai(1,4,6,j)], [A0ai(2,4,4,j) A0ai(2,4,6,j)], [0
0], 'color','b','LineWidth',2,'Marker','o','LineStyle','--');
        felo(3)=line([A0ai(1,4,6,j)
A0ai(1,4,10,j)], [A0ai(2,4,6,j) A0ai(2,4,10,j)], [0
0], 'color','m','LineWidth',2,'Marker','o','LineStyle','--');
        felo(4)=line([A0ai(1,4,10,j)
A0ai(1,4,12,j)], [A0ai(2,4,10,j) A0ai(2,4,12,j)], [0
0], 'color','m','LineWidth',2,'Marker','o','LineStyle','--');
    else
        felo(1)=line([A0ai(1,4,2,j)
A0ai(1,4,4,j)], [A0ai(2,4,2,j) A0ai(2,4,4,j)], [0
0], 'color','m','LineWidth',2,'Marker','o','LineStyle','--');
        felo(2)=line([A0ai(1,4,4,j)
A0ai(1,4,6,j)], [A0ai(2,4,4,j) A0ai(2,4,6,j)], [0
0], 'color','m','LineWidth',2,'Marker','o','LineStyle','--');
        felo(3)=line([A0ai(1,4,6,j)
A0ai(1,4,10,j)], [A0ai(2,4,6,j) A0ai(2,4,10,j)], [0
0], 'color','b','LineWidth',2,'Marker','o','LineStyle','--');
        felo(4)=line([A0ai(1,4,10,j)
A0ai(1,4,12,j)], [A0ai(2,4,10,j) A0ai(2,4,12,j)], [0
0], 'color','b','LineWidth',2,'Marker','o','LineStyle','--');
    end

    for i=1:length(PtoLcm)%Gráfica Pontos dos centros de massa
    plot(A0aLcmi(1,4,PtoLcm(i),j),A0aLcmi(2,4,PtoLcm(i),j),'ok','linewidth',
grosorpto)
    end
    for i=1:2
        if Der==1
            plot(A0ai(1,4,PtoMot(i),j),A0ai(2,4,PtoMot(i),j),'ob','linewidth',
grosorpto)
        else

```

```

plot(A0ai(1,4,PtoMot(i),j),A0ai(2,4,PtoMot(i),j),'om','linewidth',
grosorpto)
    end
    end
    for i=3:length(PtoMot)
        if Der==1
plot(A0ai(1,4,PtoMot(i),j),A0ai(2,4,PtoMot(i),j),'om','linewidth',
grosorpto)
            else
plot(A0ai(1,4,PtoMot(i),j),A0ai(2,4,PtoMot(i),j),'ob','linewidth',
grosorpto)
                end
            end
        end

        if cm(1,j)>A0ai(1,4,1,j)-Lapiet &&
cm(1,j)<A0ai(1,4,1,j)+Lapied %Gráfica projeção centro de
massa
            hcm(j)=line([cm(1,j) cm(1,j)],[0 cm(2,j)],[0
cm(3,j)'],'color','g','LineWidth',1,'Marker','. ');
            else
            hcm(j)=line([cm(1,j) cm(1,j)],[0 cm(2,j)],[0
cm(3,j)'],'color','r','LineWidth',1,'Marker','. ');
            end

        hold on
        pause(tpausa)
        if movcont==1
            set(helo(:,j),'visible','off')
            set(hcm(j),'visible','off')
            set(felo,'visible','off')
            for i=1:length(PtoLcm)

plot(A0aLcmi(1,4,PtoLcm(i),j),A0aLcmi(2,4,PtoLcm(i),j),'ow','linew
idth',grosorpto)
                end
            end
            for i=1:length(PtoMot)

plot(A0ai(1,4,PtoMot(i),j),A0ai(2,4,PtoMot(i),j),'ow','linewidth',
grosorpto)
                end
            end
        end
    end
end

```

Apêndice D. Exportação de “SolidWorks” para “Simulink de Matlab”

O primeiro passo é instalar a “biblioteca” de “Matlab” que permite a exportação, para isso ir a página web <http://www.mathworks.com> (1), e na parte superior selecionar “Products”, link de produtos e serviços (2), logo na seção “Simulink Product Family” (3), clicar em “Simscape Multibody” (4), como mostra a Figura 98

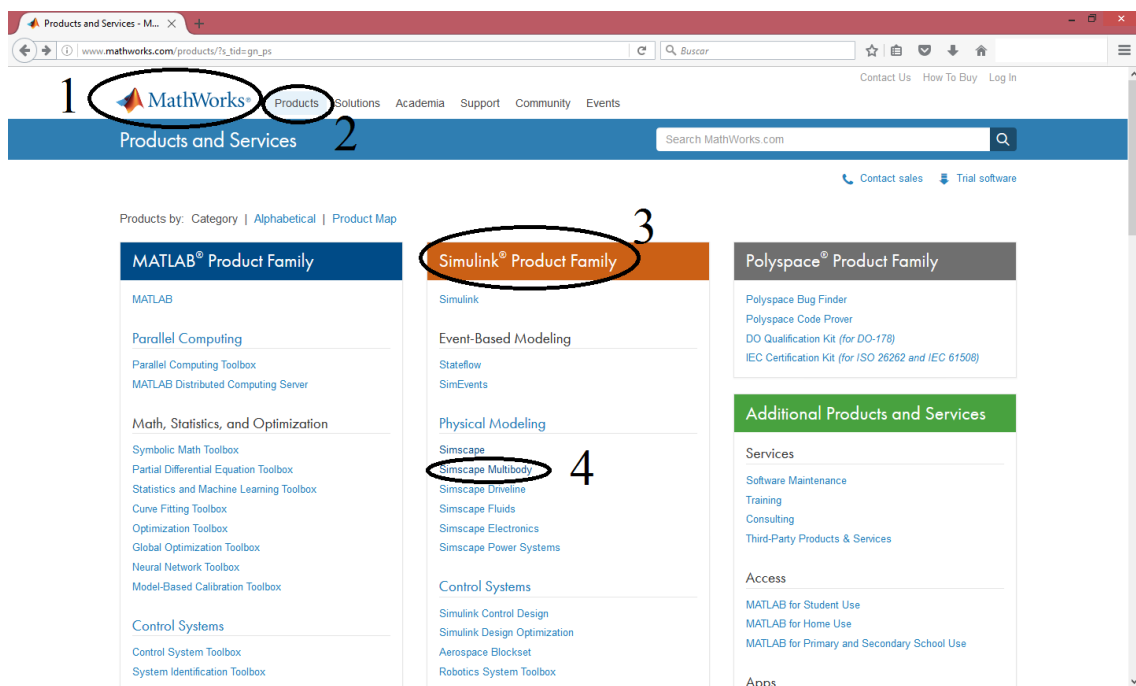


Figura 98. Web baixar SimEscape

Uma vez dentro de “Simscape Multibody”, acessar em “Importing from CAD” (5), e nessa seção clicar no link “installing a plug-in” (6), como apresenta a Figura 99 e Figura 100.

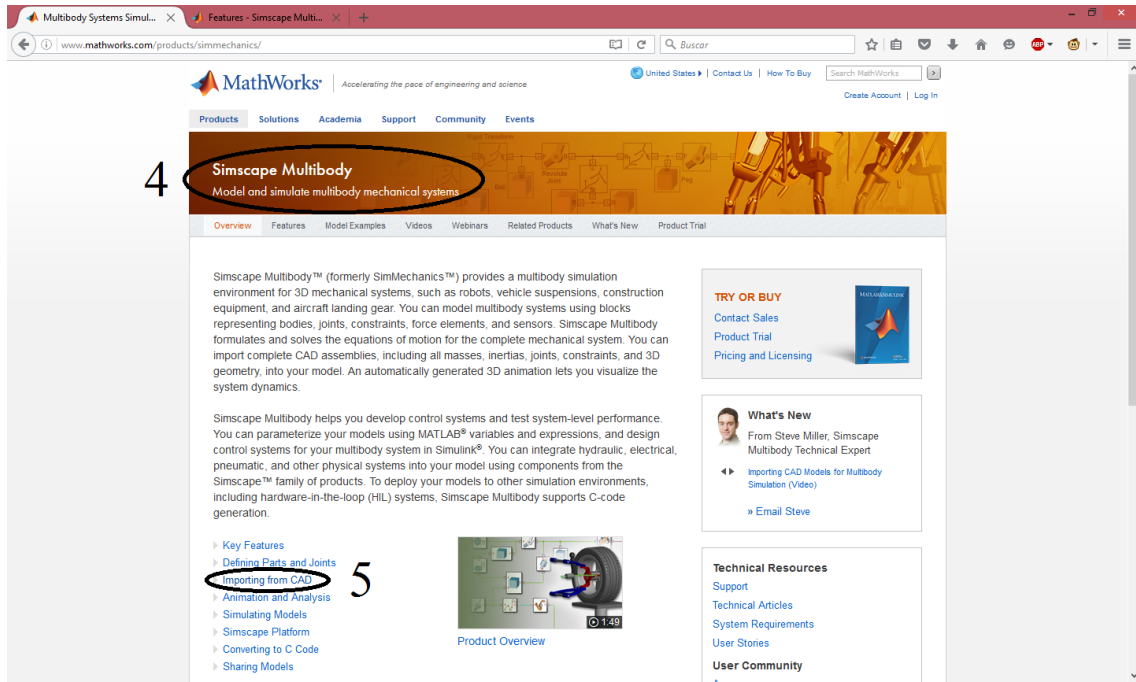


Figura 99. Web SimEscape

PUC-Rio - Certificação Digital N° 1421937/CA

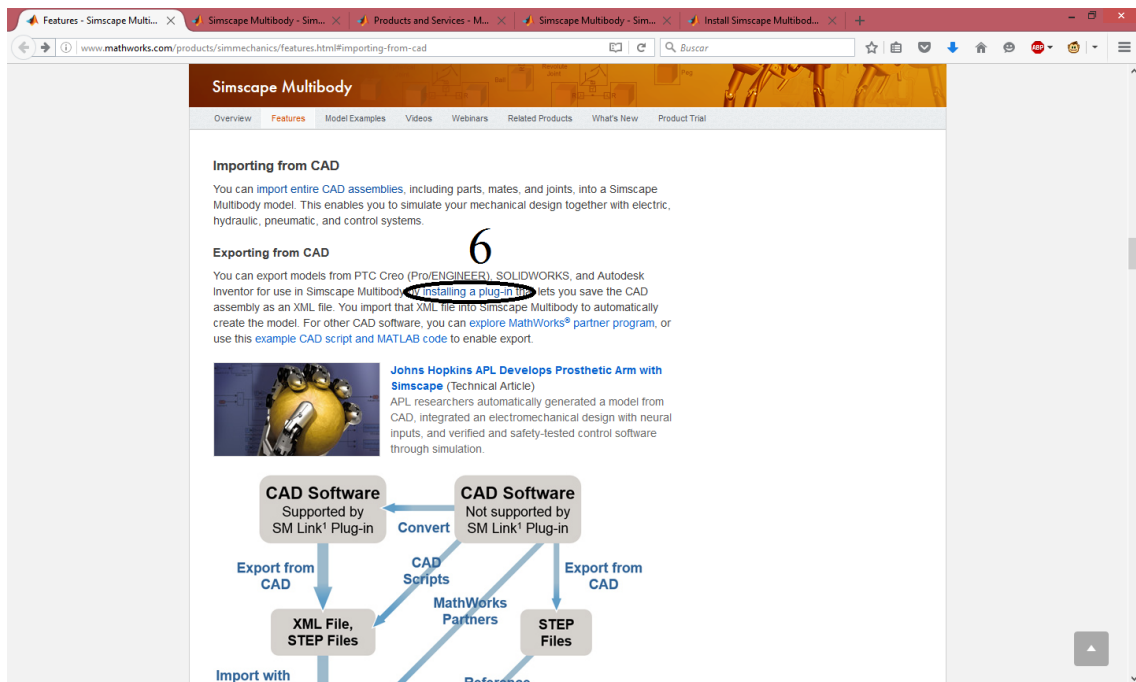


Figura 100. Link baixar

Fazer o registro da Figura 101 com os dados solicitados para poder baixar (7).

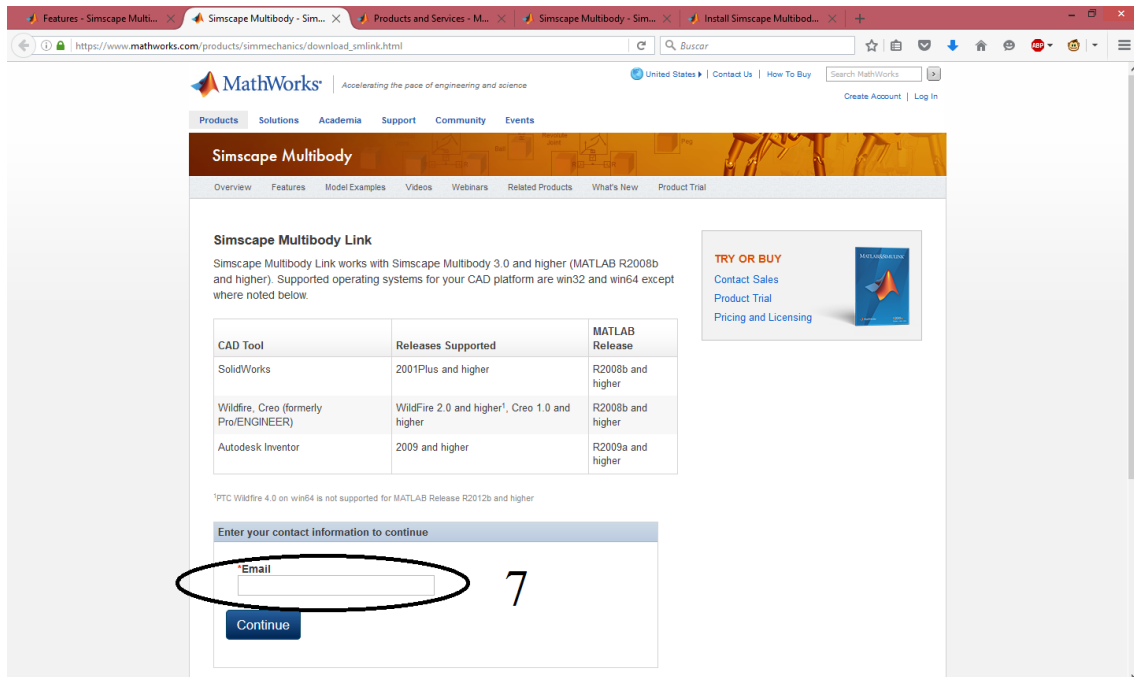


Figura 101. Registro para Simscape

Como mostra a Figura 102 baixar os arquivos segundo a versão do “Matlab” instalado (8) em uma mesma pasta.

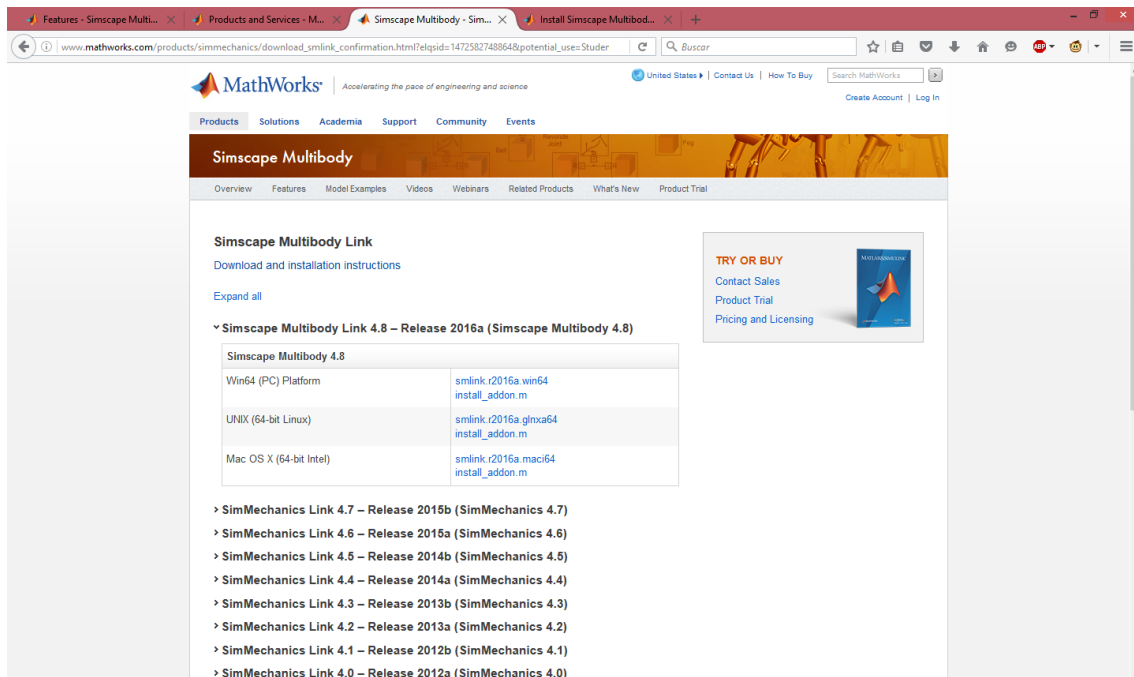


Figura 102. Arquivos SimEscape

Logo abrir o “Matlab” em modo administrador, e direcionar na pasta com os arquivos baixados. Escrever no “workspace”:
`install_addon('NomeDoArquivoZip')`

Uma vez instalado se mostra a mensagem da Figura 103.

Após para habilitar o enlace com “Solidworks”, também escrever: [smlink linksw](#).

```
>> install_addon('smlink.r2010b.win64.zip')
Installing smlink...
Extracting archive smlink.r2010b.win64.zip to C:\Program Files\MATLAB\R2010b...
Adding directories for smlink to path...
Installation of smlink complete.

To view documentation, type "doc smlink".
>> smlink_linksw
Registering dll: regsvr32 "C:\Program Files\MATLAB\R2010b\bin\win64\cl_sldwks2sm.dll"
```

Figura 103. Mensagem instalação

Abrir “SolidWorks”, e nas opções, ir a complementos (11), aparece uma janela na qual deve-se ativar o complemento “SimMechanics Link” (12), como apresenta a Figura 104.

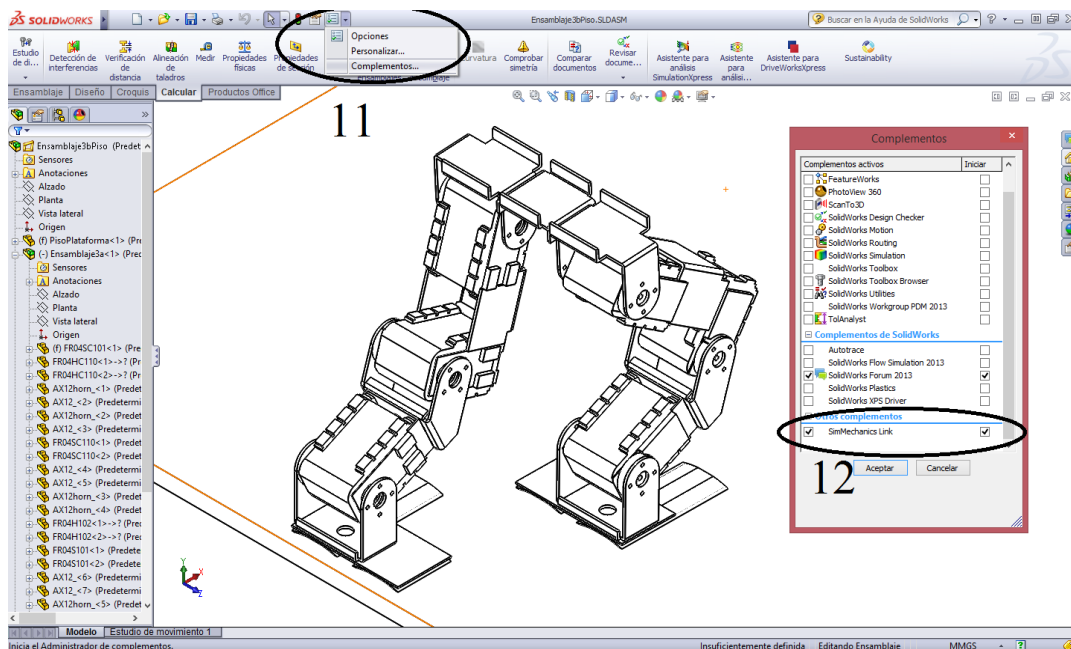


Figura 104. Ativação em Solidworks

Dentro das opções de guardar como, pode-se guardar como tipo “SimMechanicsLink - .xml”, ou ir ao menu, na opção “SimMechanicsLink”, “Export”, “SimMechanics First Generation”, “SimMechanics Second Generation”. Pode-se guardar como primeira ou segunda geração (13) como mostra a Figura 105.

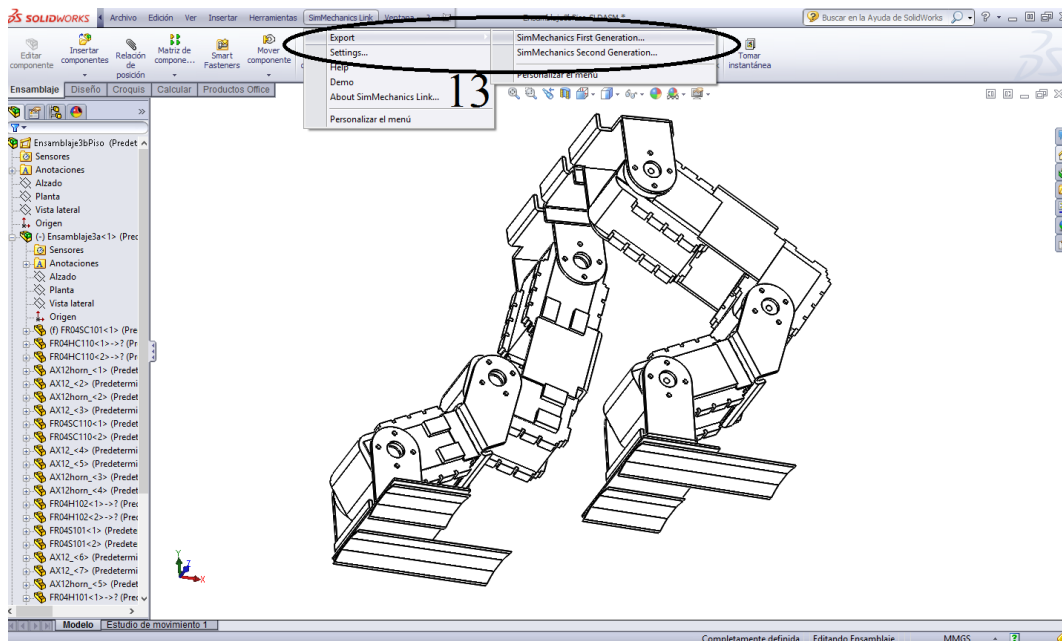


Figura 105. Salvar em solidWorks .xml

Para realizar a importação, no “workspace” escrever:

```
mech_import('NomeArquivoSolidWorks.xml') % Primeira geração
smimport(' NomeArquivoSolidWorks.xml') % Segunda geração
```

Finalmente, são gerados arquivos e o modelo em “Simulink de Matlab” pronto para ser utilizado, a Figura 106 apresenta com blocos adicionados pelo usuário. Os arquivos adicionais são importantes para a visualização 3D como o da Figura 107, correto funcionamento do programa, caso mudar os arquivos de pasta, deve-se incluir aqueles também.

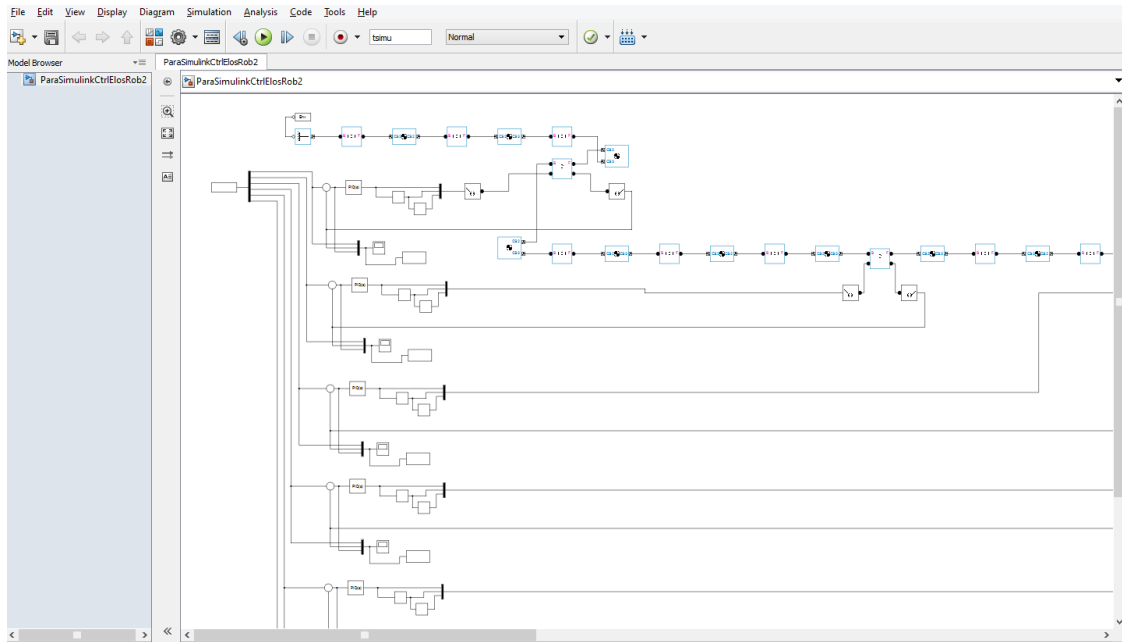


Figura 106. Interface Simulink

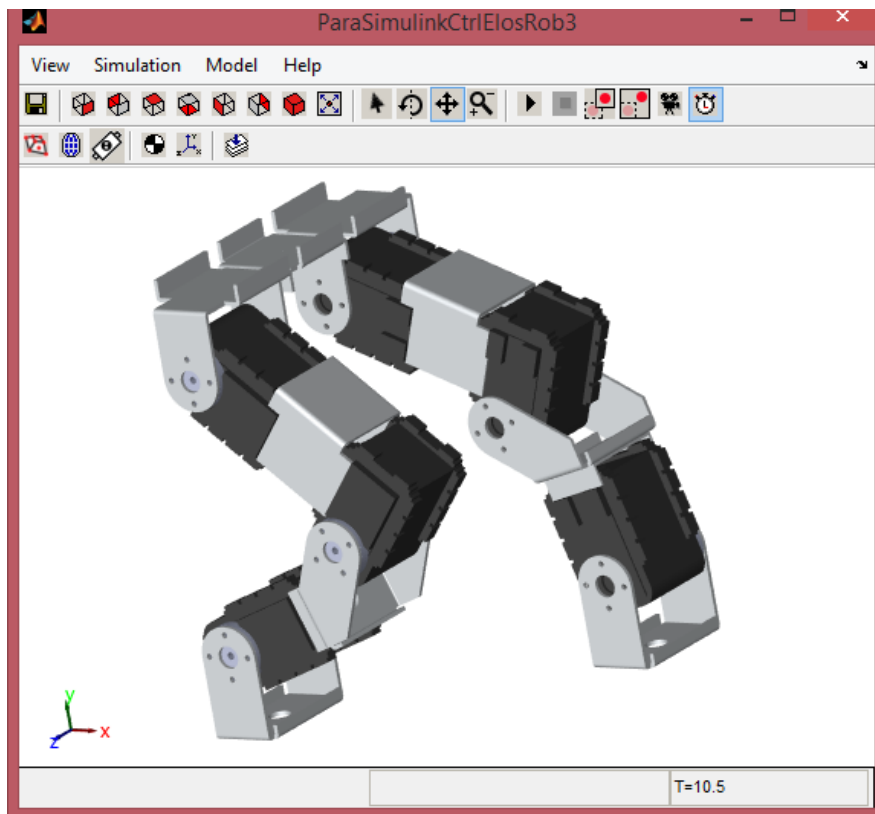


Figura 107. Modelo 3D SimMechanics

Apêndice E. Desempenho e robustez

O bloco de “Simulink”, “Pidcontroller”, é um controlador de dois graus de liberdade, tem duas entradas, a sinal de referência e a sinal retroalimentada, como se mostra na Figura 108.

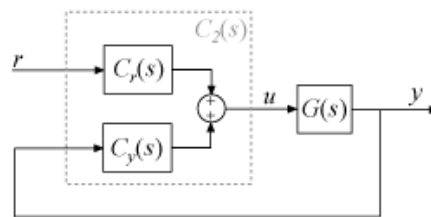


Figura 108. Arquitetura de controle de dois graus de liberdade

Cada componente $C_r(S)$ e $C_y(S)$ é um controlador pid, com diferentes pesos nos termos proporcional e derivativo. Expressos assim:

$$\begin{aligned} C_r(S) &= bK_p + \frac{K_i}{S} + \frac{cK_d S}{T_f S + 1} \\ C_y(S) &= -\left[K_p + \frac{K_i}{S} + \frac{K_d S}{T_f S + 1}\right] \end{aligned} \quad (30)$$

Onde:

b : o peso para o termo proporcional

c : peso para o termo derivativo. [40]

Os objetivos do auto ajuste inclui: estabilidade de malha fechada, a saída do sistema continua limitada para entradas limitadas; desempenho adequado, o sistema de malha fechada controla as alterações de referência e suprime distúrbios tão rapidamente quanto possível, quanto maior for a largura de banda da malha (frequência do ganho unitário em malha aberta), mais rápido o controlador responde a alterações na referência ou distúrbios no loop; robustez adequada, o projeto da malha tem suficiente margem de ganho e fase para permitir erros de modelagem ou variações na dinâmica do sistema.

O algoritmo da MathWorks® para sintonizar controladores PID atende a esses objetivos por meio do ajuste dos ganhos PID para alcançar um bom equilíbrio entre desempenho e robustez. Por padrão, o algoritmo escolhe uma frequência de crossover (largura de banda da malha), com base na dinâmica de plantas e desenhos para uma margem de fase alvo de 60° . Quando você interativamente alterar o tempo de resposta, largura de banda, a resposta transitória, ou margem de fase usando a interface PID Tuner, o algoritmo calcula novos ganhos PID.

Para uma robustez fornecida (margem de fase mínimo), o algoritmo de sintonia escolhe um projeto de controlador que equilibra as duas medidas de desempenho, acompanhamento de referência e rejeição de distúrbios. Pode-se mudar o foco do projeto para favorecer uma dessas medidas de desempenho. Para fazer isso, deve-se usar a caixa de diálogo Opções no PID Tuner.

Quando mudar o foco do projeto, o algoritmo tenta ajustar os ganhos para favorecer o acompanhamento de referência ou rejeição de distúrbios, ao conseguir a mesma margem de fase mínima. Os parâmetros ajustáveis estão no sistema, o mais provável é que o algoritmo PID possa alcançar o foco desejado, sem sacrificar a robustez. Por exemplo, definir o foco do projeto é mais provável que seja eficaz para controladores PID do que para os controladores de P ou PI. Em todos os casos, o ajuste fino do desempenho do sistema depende fortemente das propriedades da planta. Para algumas plantas, mudar o foco de criação tem pouco ou nenhum efeito [41].

Desempenho

O desempenho de uma malha retroalimentada de controle refere-se as características que avaliam o comportamento devido a câmbios na entrada do sistema, ou seja, mudanças no valor desejado, na perturbação ou nos dois juntos. Considera-se que seja ótimo respeito a função de custo que penaliza a integral do erro absoluto, definido como:

$$J_e \doteq \int_0^{\infty} |e(t)| dt = \int_0^{\infty} |r(t) - y(t)| dt \quad (31)$$

A qual deve se avaliar para as alterações no valor desejado J_{er} tanto como na perturbação da carga J_{ed} . Os parâmetros do controlador para desempenho ótimo serão:

$$J_e \doteq \int_0^{\infty} |e(t)| dt = \int_0^{\infty} |r(t) - y(t)| dt \quad (32)$$

Onde:

$\bar{\theta}_c$: parâmetros do controlador

$\bar{\theta}_p$: parâmetros da planta

Não se deve deixar o análise das características dinâmicas da ação de controle para evitar danificar o atuador, que as mudanças não sejam abruptas nem extremas. A avaliação das variações da ação de controle pode-se realizar utilizando o índice da variação total, definido por:

$$TV_u \doteq \sum_{k=1}^{\infty} |u_{k+1} - u_k| \quad (33)$$

O qual deve ser o menor possível. É uma indicação da suavidade da ação de controle. Também deve-se considerar os valores extremos $|U_{\max}|$ e $|U_{\min}|$.

A robustez refere-se à estabilidade da malha de controle devido a variações das características dinâmicas do processo controlado, respeito as do modelo nominal de parâmetros fixos utilizado para o projeto do controlador.

Como medida da estabilidade relativa ou robustez da malha de controle, utilizaram-se tradicionalmente a margem de ganância junto com a margem de fase (A_m, ϕ_m). Porem têm-se substituído pela sensibilidade máxima, definida por:

$$Ms \doteq \max_{\omega} |S(j\omega)| = \max_{\omega} \frac{1}{|1 + C_y(j\omega)P(j\omega)|} \quad (34)$$

Com valor recomendado entre 2 e 1,2. A vantagem além de ser um único índice, garante valores mínimos das margens de ganancia e fase

$$\begin{aligned} A_m &> \frac{M_s}{M_s - 1} \\ \phi_m &> 2 \operatorname{sen}^{-1} \left(\frac{1}{2M_s} \right) \end{aligned} \quad (35)$$

O desempenho robusto da malha de controle lograra-se combinando o índice de desempenho fornecido por (32) junto com a verificação da robustez mínima medida com (34) [42], [43].

Fragilidade do controlador

Em quanto a robustez é um indicador de quanto podem mudar as características do processo controlado sem virar instável, sal fragilidade está relacionada com a perda de estabilidade devido a variações nos parâmetros do próprio controlador.

O indicador de fragilidade delta épsilon $FI_{\Delta\epsilon}$ definido por Alfaro [xx] relaciona a perda de robustez quando variam os parâmetros do controlador determinada quantidade, está definido como:

$$FI_{\Delta\epsilon} \doteq \frac{M_{s\Delta\epsilon m}}{M_{s0}} - 1 \quad (36)$$

Onde a sensibilidade máxima extrema $M_{s\Delta\epsilon m}$ representa a maior perda de robustez do sistema de controle quando todos os parâmetros variam uma quantidade $\delta_\epsilon = \pm \epsilon$, e M_{s0} é a sensibilidade máxima nominal.

Considera-se um controlador como frágil se a malha de controle perde mais do que 50% de robustez quando todos os parâmetros são perturbados até um 20%. Além um controlador será elástico, se a malha de controle não perde mais de 10% de robustez quando seus parâmetros são perturbados até um 20%. A utilização da fragilidade permite o projeto de controladores com desempenho robusto [42].