



UNIVERSIDADE DO ESTADO DA BAHIA
DEPARTAMENTO DE CIÊNCIAS EXATAS E DA TERRA
CURSO DE GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO

EMMANUEL MARMUND ARGOLLO

**DESENVOLVIMENTO DE UM MÓDULO PARA A EXECUÇÃO DO MOVIMENTO
DE CORRIDA EM UM AGENTE HUMANOIDE SIMULADO**

SALVADOR - BAHIA

2017

EMMANUEL MARMUND ARGOLLO

DESENVOLVIMENTO DE UM MÓDULO PARA A EXECUÇÃO DO MOVIMENTO DE
CORRIDA EM UM AGENTE HUMANOIDE SIMULADO

Monografia apresentada ao curso de Sistemas de Informação do Departamento de Ciências Exatas e da Terra da Universidade do Estado da Bahia- UNEB, como requisito parcial à obtenção do grau de bacharel em Sistemas de Informação. Área de Concentração: Matemática Aplicada

Orientador: Msc. Marco Antônio Costa Simões

SALVADOR - BAHIA

2017

EMMANUEL MARMUND ARGOLLO

DESENVOLVIMENTO DE UM MÓDULO PARA A EXECUÇÃO DO MOVIMENTO DE
CORRIDA EM UM AGENTE HUMANOIDE SIMULADO

Monografia apresentada ao curso de Sistemas de Informação do Departamento de Ciências Exatas e da Terra da Universidade do Estado da Bahia- UNEB, como requisito parcial à obtenção do grau de bacharel em Sistemas de Informação. Área de Concentração: Matemática Aplicada

Aprovada em:

BANCA EXAMINADORA

Msc. Marco Antônio Costa Simões (Orientador)
Universidade do Estado da Bahia – UNEB

Prof. Dr. Augusto Loureiro da Costa
Universidade Federal da Bahia - UFBA

Prof. Dr. Robson Marinho da Silva
Universidade do Estado da Bahia – UNEB

Prof. Dr. Diego Gervasio Frías Suárez
Universidade do Estado da Bahia – UNEB

AGRADECIMENTOS

Agradeço primeiramente a minha família por sempre me apoiarem, em especial meus pais, que sempre fizeram questão de priorizar minha educação e terem me ajudado a me tornar a pessoa que sou hoje.

Agradeço também a minha namorada por todo o suporte que me deu durante esse processo e por sempre me motivar em momentos de desânimo.

Aos meus amigos de longa data por entenderem quando eu sumia por excesso de trabalhos e sempre estarem presentes quando precisei. Às novas amizades criadas durante o curso, em especial aos membros e ex-membros do ACSO por todas as risadas, discussões e viradas de noite trabalhando. Aos amigos que fiz durante o Ciência sem Fronteiras, foi um ano essencial para o meu desenvolvimento e não teria sido tão bom sem vocês.

Aos pesquisadores do ACSO pela oportunidade de participar desse grupo desde 2011 e vê-lo crescer até onde já chegou, e espero que continue crescendo. E a todos os professores da UNEB, pelo conhecimento passado e por estarem sempre dispostos a ajudar e tirar dúvidas.

"If you wanna change the world, all you have to do is just look at it differently... am I right?"

(Ryuji Sakamoto)

RESUMO

O desenvolvimento de habilidades motoras para robôs humanoides é uma área de pesquisa bastante ampla, envolvendo diversas áreas do conhecimento. Dentre as habilidades de locomoção, o movimento de andar já foi reproduzido de maneira eficiente com cálculos dinâmicos, entretanto o movimento de corrida ainda é um desafio para a locomoção robótica. O trabalho aqui apresentado está inserido na liga de simulação 3D da RoboCup, que utiliza agentes humanoides em um ambiente simulado para o desenvolvimento de estratégias multiagente e movimentos eficientes para a executar tais estratégias. Nesse ambiente, a capacidade de se mover rapidamente de um ponto a outro é um dos maiores fatores para o desempenho do agente. Desse modo, esse projeto tem como objetivo desenvolver um módulo inicial para a execução do movimento de corrida por esse agente. Para tal, é estudado e utilizado o modelo de massa e mola e estratégias para transição entre o movimento de corrida e andar. O módulo desenvolvido nesse projeto obteve sucesso em demonstrar as etapas principais do movimento de corrida, podendo ser aprimorado futuramente com melhorias em seus componentes.

Palavras-chave: Robótica, Modelagem matemática, Agente humanoide

ABSTRACT

The development of motor skills for humanoid robots is a very broad area of research, involving several areas of knowledge. Among the locomotion abilities, the walking movement has already been reproduced efficiently with dynamic calculations, however the running movement is still a challenge for robotic locomotion. The work presented here is inserted in RoboCup 3D simulation league, which uses humanoid agents in a simulated environment for the development of multiagent strategies and efficient movements to execute such strategies. In this environment, the ability to move quickly from one point to another is one of the major factors for agent performance. As such, this project aims to develop an initial engine for the execution of the running movement by this agent. To achieve this, the spring-mass model and transition strategies between the running and walking movements are studied and implemented. The engine developed in this project has succeeded in demonstrating the main steps of the running movement, and can be enhanced in the future with improvements in its components.

Keywords: Robotics, Mathematica modeling, humanoid agent.

LISTA DE FIGURAS

Figura 1 – Robô NAO físico (esquerda) e simulado (direita)	15
Figura 2 – Campo oficial da liga de simulação 3D	16
Figura 3 – Sincronização entre o agente e o servidor.	17
Figura 4 – Estrutura das 22 juntas do robô.	18
Figura 5 – Tipos de juntas utilizadas no robô NAO simulado.	19
Figura 6 – Parâmetros de DH.	25
Figura 7 – Etapas do movimento de andar humano.	31
Figura 8 – Polígono de suporte nas etapas de suporte único e suporte duplo.	32
Figura 9 – Forças atuantes no pé de suporte de um agente, resultando no ZMP.	33
Figura 10 – O modelo <i>Cart-Table</i> aplicado no robô NAO.	35
Figura 11 – Visualização esquemática do modelo <i>Cart-Table</i>	36
Figura 12 – Diagrama do controle de previsão do ZMP.	37
Figura 13 – Trajetórias do ZMP e do CoM.	37
Figura 14 – O modelo do pêndulo invertido com vara telescópica aplicado ao robô NAO.	38
Figura 15 – Visualização esquemática do modelo do pêndulo invertido.	39
Figura 16 – Movimento de andar sem o equilíbrio ativo (a) e com ele (b).	40
Figura 17 – Comparação entre as etapas dos movimentos de corrida(acima) e caminhada(abaixo).	41
Figura 18 – Visualização esquemática do modelo SLIP	42
Figura 19 – Solução radial para $r(t)$	45
Figura 20 – Comparação entre o movimento da perna de balanço em um humano e o modelo proposto por Seyfarth et al. (2003).	48
Figura 21 – Autômato híbrido representando os padrões dos movimentos de andar, correr e transições entre eles.	49
Figura 22 – Modelo SLIP na fase de suporte duplo e abstração do sistema como um SLIP único com mola axial e de torção.	50
Figura 23 – Módulo de andar omnidirecional, com todos seus sub-módulos e interações.	52
Figura 24 – Fluxo de execução de chute baseado na cinemática inversa.	52
Figura 25 – Etapas do desenvolvimento do projeto.	53
Figura 26 – Proposta do Módulo de corrida e caminhada.	54
Figura 27 – Automato adaptado para a transição entre movimentos.	56

Figura 28 – Modelo do pendulo invertido com VLO em destaque.	57
Figura 29 – Representação da curva de Bézier gerada pelos pontos p_1 , p_2 e p_3 . Em azul a representação da perna do agente e em verde, a curva que ela deve seguir. .	59
Figura 30 – Representação da curva de Bézier gerada pelos pontos p_1 , p_2 , p_3 e p_4 . Em azul a representação da perna do agente e em verde, a curva de Bézier que ela deve seguir. Após alcançar o ponto p_4 , no ápice da etapa de voo, a perna segue a curva em vermelho, para se preparar para o pouso.	60
Figura 31 – Posição ideal do pé com relação ao CoM durante o movimento.	61
Figura 32 – Posição calculada para a posição do pé com relação ao CoM.	62
Figura 33 – Ferramenta <i>Magma Challenge Benchmark</i> com desafio de chute selecionado.	65
Figura 34 – Protótipo do modelo de corrida. Em vermelho e preto, as pernas esquerda e direita do agente, respectivamente e em azul, o centro de massa do mesmo. .	66
Figura 35 – Comparação da posição dos pés executadas versus desejadas nos eixos X Y e Z respectivamente. Os gráficos de cima representando as posições para o pé direito, e os de baixo para o esquerdo. Em azul o movimento planejado e em preto o executado.	69
Figura 36 – Velocidades enviadas para o servidor para as juntas das pernas. À esquerda, as juntas da perna esquerda, e à direita, as da perna direita. Em amarelo, magenta e ciano as juntas <i>YawPitch</i> , <i>Row</i> e <i>Pitch</i> do quadril, respectivamente. Em azul a junta <i>Pitch</i> do joelho. E em verde e vermelho as juntas <i>Pitch</i> e <i>Roll</i> do tornozelo respectivamente. As linhas tracejadas verticais representando as etapas do movimento e as horizontais a velocidade máxima das juntas. . . .	69
Figura 37 – Comparação da posição dos pés executadas versus desejadas nos eixos X Y e Z respectivamente. Os gráficos de cima representando as posições para o pé direito, e os de baixo para o esquerdo. Em azul o movimento planejado e em preto o executado com limites de velocidades nas juntas e em vermelho, o movimento executado sem os limites de velocidade.	70
Figura 38 – Força aplicada na sola dos pés do agente durante o movimento. Em azul o pé direito e em preto o esquerdo. As linhas tracejadas verticais representando a separação entre etapas de postura e voo.	71

LISTA DE TABELAS

Tabela 1 – Parâmetros de DH para cadeia de juntas da perna esquerda.	26
Tabela 2 – Parâmetros de DH para cadeia de juntas da perna direita.	26
Tabela 3 – Notação utilizada para descrever o autômato de transição de movimentos . .	49
Tabela 4 – Notação dos principais parâmetros do módulo de correr e andar.	55
Tabela 5 – Parâmetros utilizados nos testes de validação do movimento.	67
Tabela 6 – Valores previstos pelo protótipo para o movimento.	68
Tabela 7 – Comparação dos resultados previstos pelo protótipo com os coletados após a execução.	71

LISTA DE ABREVIATURAS E SIGLAS

CoM	<i>Center of Mass</i>
DH	<i>Denavit-Hartenberg</i>
DOF	<i>Degrees of Freedom</i>
FRP	<i>Force-Resistance Perceptor</i>
GCoM	<i>Ground projection of Center of Mass</i>
IP	<i>Inverted Pendulum</i>
PID	<i>Proportional-Integral-Derivative</i>
SLIP	<i>Spring-Loaded Inverted Pendulum</i>
SPL	<i>Standard Platform League</i>
VLO	<i>Vertical Leg Orientation</i>
ZMP	<i>Zero Moment Point</i>

SUMÁRIO

1	INTRODUÇÃO	13
2	LIGA DE SIMULAÇÃO 3D	15
2.1	Ambiente de Simulação	16
2.2	Modelo do robô NAO simulado	18
3	CINEMÁTICA ROBÓTICA	21
3.1	Transformações afins	21
3.1.1	Matriz de transformação afim	21
3.1.2	Matriz de translação	22
3.1.3	Matriz de rotação	23
3.1.4	Parâmetros de Denavit Hartenberg	24
3.2	Cinemática Direta	25
3.2.1	Cinemática direta do robô NAO	26
3.3	Cinemática Inversa	27
3.3.1	Solução geométrica	27
4	MOVIMENTO ANDAR EM ROBÔS HUMANOIDES	31
4.1	Medidas de estabilidade do movimento	31
4.1.1	Projeção do centro de massa	32
4.1.2	Zero Moment Point	33
4.2	Modelos do Movimento Andar	34
4.2.1	Modelo <i>Cart-Table</i>	35
4.2.2	Modelo do pêndulo invertido	38
4.2.3	Equilíbrio ativo com retroalimentação	40
5	MODELOS PARA O MOVIMENTO DE CORRER	41
5.1	Modelo <i>Spring-Mass</i>	42
5.1.1	Solução aproximada para CoM	42
5.1.2	Controle da perna de balanço	47
5.2	Modelagem unificada para os movimentos de andar e correr	48
6	TRABALHOS RELACIONADOS	51
6.1	Módulos de execução de movimentos	51

7	MÓDULO DE CORRIDA E CAMINHADA PARA ROBÔS HUMANOIDES	53
7.1	Módulo de Corrida e Caminhada	54
7.1.1	Autômato de transição entre movimentos	55
7.1.2	Previsão do Período da Fase	56
7.1.3	Controle da trajetória do CoM	57
7.1.4	Controle de trajetória da perna de balanço	58
7.1.5	Controle de oscilação lateral	60
7.1.6	Posição dos pés	62
8	TESTES E RESULTADOS	65
8.1	Procedimento experimental	65
8.2	Resultados	67
9	CONSIDERAÇÕES FINAIS	72
9.1	Trabalhos Futuros	72
	REFERÊNCIAS	74

1 INTRODUÇÃO

A RoboCup é uma iniciativa com objetivo de incentivar a pesquisa na área de robótica e inteligência artificial de maneira atrativa e desafiadora. Com isso em mente, foi decidida a criação de um desafio com uma meta alvo de longo prazo, surgindo assim o futebol de robôs, com o objetivo de em meados do século XXI, conseguir ser criado um time de robôs humanoides capaz de derrotar o time campeão da FIFA (ASADA; KITANO, 1999). Para alcançar esse objetivo, o desafio foi dividido em diversas ligas ou categorias, dentre elas, está a liga de simulação 3D. Criada em 2004, a liga de simulação 3D surgiu com intuito de continuar o desenvolvimento de pesquisas na área de coordenação multi-agentes, iniciada pela liga de simulação 2D, mas acrescentando o grau extra de dificuldade de possuir agentes com sensores e atuadores simulados, bem como toda a física do ambiente. Desse modo, pesquisas que não poderiam ser realizadas em robôs físicos, seja por questão de tempo, financiamento, ou limites de *hardware* passam a ser mais acessíveis (AKIYAMA et al., 2014). Dentro da liga de simulação 3D, a UNEB é representada pelo time BahiaRT (SIMOES et al., 2016).

Para um aproveitamento positivo, um time da liga de futebol de robôs simulados 3D precisa ter uma base de movimentos eficiente, capaz de sustentar as decisões estratégicas do agente. Esses movimentos podem ser executados com o uso de comandos estáticos, posicionando o robô em determinadas poses, que em sequência formam o movimento (*Keyframe Motion*) ou pela geração dinâmica do movimento em tempo de execução. Enquanto movimentos mais simples como o de levantar e alguns chutes podem ser executados por *Keyframe Motion*, movimentos mais complexos como o de andar precisam ser calculados iterativamente durante a sua execução (STROM et al., 2009). Atualmente, o único movimento dinâmico em uso pelo BahiaRT é o de andar.

O movimento dinâmico de caminhada humanoide é um comportamento já executado e estudado por todos os times da liga de simulação 3D, apesar de cada time possuir abordagens diferentes para a otimização e variações do modelo. O movimento de andar é composto por duas fases, a fase de suporte único, quando apenas um dos pés está tocando o chão, e a de suporte duplo, quando o peso do corpo é sustentado pelas duas pernas (KAJITA et al., 2001). Em comparação, o movimento de corrida humanoide não possui a fase de suporte duplo, sendo substituída pela fase aérea ou de voo, na qual nenhuma parte do corpo toca o chão e o centro

de massa demonstra uma trajetória balística (GEYER et al., 2005). A substituição dessa fase no movimento de corrida resulta em sua vantagem sobre o movimento de andar. Devido à sua fase de voo, o movimento de corrida consegue percorrer maiores distâncias em um passo, não limitadas pelo tamanho de suas pernas, como o movimento de andar. Além disso, devido ao seu movimento vertical, a força de reação do chão em relação ao pé gera uma fricção maior, permitindo movimentos mais rápidos (TAKENAKA et al., 2009).

Apesar de suas vantagens, o movimento de corrida apresenta um nível de complexidade maior do que o de andar, e nenhum dos times da liga de simulação 3D conseguiu demonstrar uma implementação satisfatória do mesmo, sendo um dos maiores desafios atuais da liga (AKIYAMA et al., 2014). Esse trabalho se propõe a preencher essa lacuna com o desenvolvimento de um módulo inicial para a execução desse movimento. Para transicionar entre o estado estático e o movimento de corrida é necessário passar pelo movimento de caminhada, portanto o módulo proposto nesse trabalho procura incluir ambos os movimentos, bem como um método para realizar a transição entre os dois.

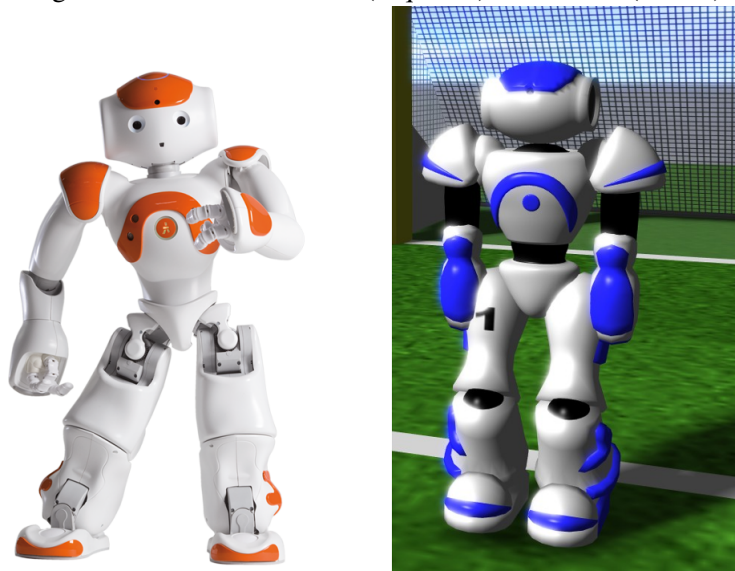
Esse trabalho está estruturado em 7 capítulos. No capítulo 2 é descrita a liga de simulação 3D da RoboCup, onde esse trabalho está inserido, o ambiente de simulação e o agente robótico utilizado para validar esse trabalho. No capítulo 3 são apresentadas as características gerais da cinemática robótica, com foco nas soluções utilizadas para o modelo de robô utilizado nesse trabalho. O capítulo 4 apresenta modelos do movimento andar, bem como medidas de estabilidade do mesmo. Já no capítulo 5, serão apresentados modelos de controle para o movimento de corrida. No capítulo 7, é apresentada a proposta do projeto, com sua metodologia e detalhes do módulo. O capítulo 8 apresenta o plano de testes e os resultados apresentados pelo projeto e por fim, no capítulo 9 é apresentada a conclusão do projeto, e os possíveis trabalhos futuros.

2 LIGA DE SIMULAÇÃO 3D

As ligas de simulação sempre estiveram presentes na RoboCup. Desde seus anos iniciais, a liga de simulação 2D fez bastante sucesso devido às suas abstrações dos problemas de locomoção dos robôs, permitindo que o trabalho fosse mais focado na estratégia dos agentes. Em 2004, foi criada a liga de simulação 3D, partindo da mesma ideia de simplificar problemas de *hardware*, para que o trabalho pudesse ser voltado ao desenvolvimento de modelos e *softwares*. A liga de simulação 3D entretanto, se diferenciou da liga anterior por simular robôs físicos, para quais precisariam ser desenvolvidos movimentos de maneira similar aos robôs reais.

A partir desse princípio, a liga se voltou para o desenvolvimento de capacidades de movimento que pudessem ser facilmente aplicáveis nos robôs físicos. Para facilitar essa portabilidade, em 2008 os novos agentes da liga foram introduzidos, modelados com base no robô NAO, desenvolvido pela empresa Aldebaran Robotics em 2006. Esse modelo foi escolhido por ser o robô oficial utilizado pela *Standard Platform League (SPL)*, a liga de plataforma padrão para robôs humanoides. O agente modelado possui algumas alterações em comparação com sua versão física, mas ainda assim, grande parte do que é desenvolvido para a liga de simulação 3D pode ser diretamente aproveitado pelos robôs físicos. Isso permite que diversas equipes trabalhem paralelamente em ligas físicas e simuladas. A figura 1 apresenta o robô original e sua versão simulada.

Figura 1 – Robô NAO físico (esquerda) e simulado (direita)



A liga de simulação 3D iniciou com apenas 2 agentes por time. Após diversos

avanços, tanto nos times, no simulador e nas máquinas utilizadas para executar a simulação, hoje são realizadas partidas similares ao futebol real, com 11 jogadores em cada time. O campo também foi ajustado para ser proporcional ao tamanho dos agentes, como visto na figura 2, tendo 20 metros de largura por 30 metros de comprimento, sendo assim proporções um pouco inferiores a um campo de futebol oficial. Muito foi desenvolvido durante esse tempo, os movimentos dos robôs ficaram mais diversificados e aos poucos, estratégias que já eram utilizadas na liga de simulação 2D passaram a ser possíveis nessa liga.

Figura 2 – Campo oficial da liga de simulação 3D



Fonte – O autor

2.1 AMBIENTE DE SIMULAÇÃO

De acordo com as regras da competição, o ambiente de simulação 3D deve ser executado de forma distribuída entre quatro máquinas. Essa distribuição ocorre da seguinte forma:

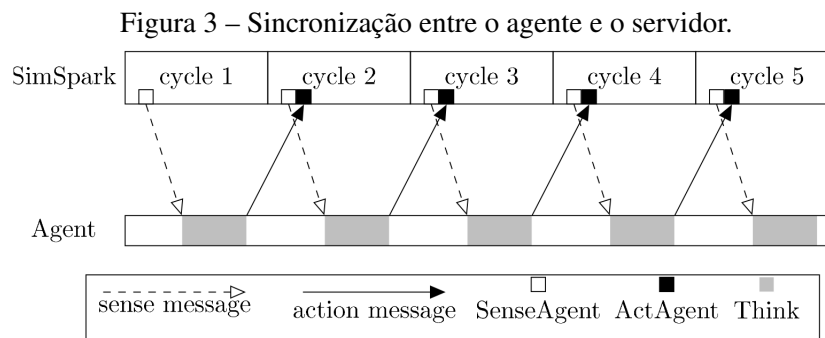
- O primeiro computador é responsável por gerenciar as partidas da liga, inicializando automaticamente o servidor e os times nas devidas máquinas. Durante a execução das partidas essa máquina também é responsável por renderizar a partida como monitor. Atualmente isso é feito através da ferramenta Roboviz (STOECKER; VISSER, 2012).
- Outra máquina é dedicada à execução do servidor de simulação, o *SimSpark* (BOEDCKER; ASADA, 2008), que é responsável por calcular a física do ambiente de maneira mais próxima ao mundo real possível. Simulando forças como gravidade, atrito, fricção

dentre outras.

- Por fim, cada time participando da partida tem um computador dedicado para executar seus agentes. Cada agente é representado por um processo independente, resultando em cada uma dessas máquinas executando 11 processos. Hoje em dia, essas máquinas também são responsáveis por executar um agente *proxy*, que garante a estabilidade da conexão entre os agentes e o servidor.

Uma vez iniciados, o servidor e os agentes se conectam através de transmissões de mensagens TCP/IP. A simulação é realizada como um sistema de tempo real, com ciclos de 20 milissegundos, ou seja, o agente possui apenas esse tempo para realizar todas as tomadas de decisões necessárias, com base nas informações recebidas pelo servidor, e enviar suas ações de volta para o mesmo. Caso isso não ocorra, o agente perderá a oportunidade de realizar as ações esperadas nesse ciclo, resultando em um comportamento inconsistente.

Para realizar essa simulação, o simulador possui dois métodos essenciais: O *sense agent* e o *act agent*, que podem ser vistos na figura 3.



O método *sense agent* gera todas as informações sensoriais que cada agente irá receber naquele ciclo de execução, como visão, audição, estado dos motores corporais dentre outros. Essas informações são estruturadas em forma de mensagem e enviadas ao agente.

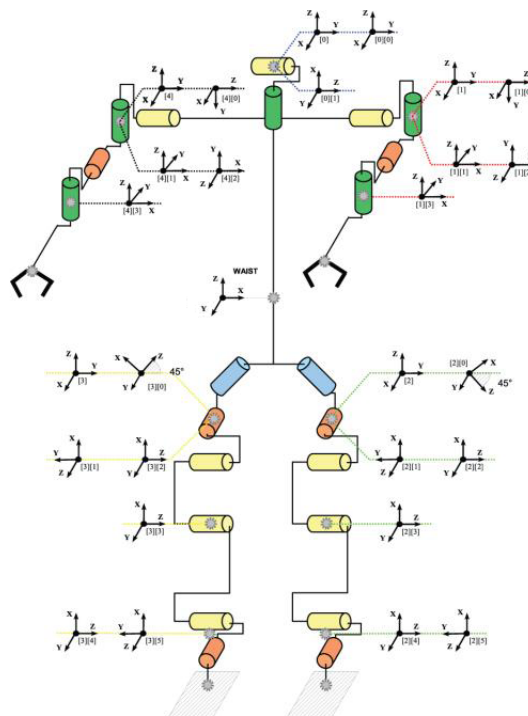
A cada ciclo, após enviar a mensagem *sense*, o servidor recebe e processa a última mensagem de tipo *act* enviada pelo agente, e a executa no ambiente simulado. Como a mensagem *act* é processada apenas após o servidor enviar sua mensagem *sense* do ciclo, o agente só recebe as informações do resultado de sua ação no ciclo seguinte.

2.2 MODELO DO ROBÔ NAO SIMULADO

Para executar suas ações dentro do ambiente simulado o agente envia mensagens *act* para o servidor, contendo o conjunto de ações que deseja realizar, representadas como velocidades para cada atuador das juntas motoras. Em robôs reais, tais atuadores seriam equivalentes a servos e motores. Para realizar o controle das juntas, a mensagem enviada tem o formato ($\langle nome \rangle \langle velocidade \rangle$), contendo o identificador da junta e a velocidade desejada a ser executada para a mesma. Uma vez enviada, o servidor continuará a executar essa velocidade nessa junta até que receba uma nova velocidade para o mesmo identificador. Para poder regular as velocidades e objetivos angulares das juntas, os agentes recebem na mensagem *sense* do servidor os dados coletados pelos perceptores de cada junta, com o formato ($HJ(n \langle nome \rangle)(ax \langle ax \rangle)$), contendo o nome da junta e seu ângulo atual.

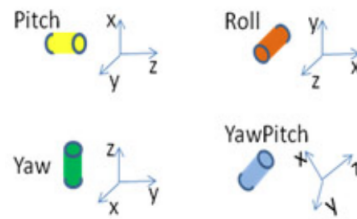
O robô simulado usado pela liga, com base no robô NAO, possui 22 juntas, vistas na figura 4. Essas juntas são classificadas em 4 categorias, de acordo com seu eixo de rotação em relação ao torso do agente, que representa o centro do seu eixo cartesiano. As quatro categorias podem ser visualizadas na figura 5. As juntas, para o modelo padrão do agente, possuem velocidade máxima de rotação de 6.1395rad/s .

Figura 4 – Estrutura das 22 juntas do robô.



Fonte – Retirado de (BOEDECKER; ASADA, 2008)

Figura 5 – Tipos de juntas utilizadas no robô NAO simulado.



Fonte – Adaptado de (BURKHARD; DOMANSKA, 2012)

- **Yaw:** Rotaciona em torno do eixo Z.
- **Roll:** Rotaciona em torno do eixo Y.
- **Pitch:** Rotaciona em torno do eixo X.
- **YawPitch:** Junta presente apenas no quadril do agente, que apresenta uma rotação simultânea aos eixos Z e X, sendo uma junta intermediária entre a *Yaw* e *Pitch*.

Para poder analisar seu estado físico, além das posições de suas juntas, o agente possui três tipos de sensores corporais: um perceptor *GyroRate*, localizado no torso do agente, que traz informações sobre mudanças na orientação do corpo do mesmo, representada nos 3 eixos (X, Y e Z); um *Acelerômetro*, também localizado no torso, apresentando a aceleração aplicada ao corpo do agente nos eixos liberdade; e dois sensores de força de resistência, *Force-Resistance Perceptor* (FRP), localizados nas solas dos pés do agente. Esse sensor apresenta a força aplicada no pé do agente em cada eixo, e a posição relativa ao pé onde a força está sendo aplicada.

Além das juntas responsáveis por movimentar o corpo do agente e dos sensores presentes no mesmo, é importante destacar as dimensões e as partes do corpo do agente. Apesar do agente ser renderizado com formas geométricas complexas, para uma visualização mais interessante do jogo e mais semelhante ao robô NAO, ao qual ele é baseado, o agente simulado é composto de formas simples, retângulos e esferas apenas. Toda a física aplicada ao agente, incluindo colisões, são feitas com base nesse modelo.

Graças à facilidade de se alterar o modelo simulado do agente, foram criados modelos do robô com pequenas alterações, com base na premissa que uma vez provado que certos modelos podem ter capacidades melhores no ambiente simulado, teriam resultados semelhantes no mundo real. Esses modelos, chamados de heterogêneos, atualmente são representados por:

- **Tipo 1:** Possui pernas e braços mais longos do que o modelo padrão.
- **Tipo 2:** Possui algumas juntas capazes de se mover mais rápido, e outras mais lentas, do

que a velocidade padrão.

- **Tipo 3:** Possui quadris mais largos, além de pernas e braços mais longos do que o modelo padrão.
- **Tipo *Toe*:** Possui uma junta extra do tipo *pitch* no seu pé, para representar os dedos dos pés em um ser humano.

3 CINEMÁTICA ROBÓTICA

Kofinas (2012) define uma cadeia de cinemática robótica como um manipulador articulado que interage com o ambiente e é comumente descrito como um conjunto de ligações robóticas conectadas por juntas rotatórias. A partir da rotação dessas juntas, as ligações se movem alterando a posição de diferentes partes da cadeia. A quantidade de juntas em uma cadeia é chamada de graus de liberdade ou *Degrees of Freedom* (DOF). A partir de diferentes combinações para as juntas da cadeia, movimentos mais complexos podem ser criados.

A cinemática robótica é o estudo de cadeias cinemáticas com múltiplos DOF. A partir do estudo da cinemática robótica, pode-se transformar o espaço de juntas, a combinação das diferentes rotações das juntas em uma cadeia, em espaço cartesiano e vice-versa. Com essa conversão, torna-se mais efetivo o controle e o planejamento do movimento da cadeia.

3.1 TRANSFORMAÇÕES AFINS

Uma transformação afim é uma função que modifica o espaço de um ponto ou vetor, preservando suas proporções. Exemplos de transformações afim são: translação, rotação, contração, dilatação, reflexão, dentre outros. Uma transformação afim parte e resulta em um espaço n -dimensional, com $n \geq 2$. Para esse trabalho, serão demonstradas apenas translações e rotações, em um espaço tridimensional.

3.1.1 Matriz de transformação afim

Para realizar transformações afins no espaço n -dimensional é utilizada uma matriz de transformação afim, de dimensões $(n + 1) \times (n + 1)$. Essa matriz pode ser representada da seguinte forma:

$$\mathbf{A} = \left[\begin{array}{c|c} X & Y \\ \hline 0 \dots 0 & 1 \end{array} \right] \quad (3.1)$$

Onde X é uma matriz $n \times n$, e Y é um vetor $n \times 1$. A última linha da matriz é composta de n zeros, seguidos por um 1. A partir dessa matriz, qualquer transformação afim

pode ser aplicada a um ponto p , multiplicando a matriz por esse ponto.

$$\left[\begin{array}{c|c} X & Y \\ \hline 0 \dots 0 & 1 \end{array} \right] \times \begin{bmatrix} p_1 \\ \vdots \\ p_n \\ 1 \end{bmatrix} \quad (3.2)$$

As matrizes de transformação afim possuem duas propriedades principais: o produto de qualquer número de transformações afim é uma transformação afim, e uma matriz de transformação afim é invertível se sua componente X for invertível.

$$\hat{\mathbf{A}} = \mathbf{A}_1 \mathbf{A}_2 \mathbf{A}_3 \dots \mathbf{A}_k = \left[\begin{array}{c|c} \hat{X} & \hat{Y} \\ \hline 0 \dots 0 & 1 \end{array} \right] \quad (3.3)$$

$$\mathbf{A}^{-1} = \left[\begin{array}{c|c} X^{-1} & -X^{-1}Y \\ \hline 0 \dots 0 & 1 \end{array} \right] \quad (3.4)$$

Para o uso na cinemática robótica, um ponto no espaço cartesiano será representado por uma matriz de transformação afim, com a componente X representando a sua orientação no espaço, e Y a sua posição.

$$\mathbf{A} = \left[\begin{array}{c|c} R & P \\ \hline 0 \dots 0 & 1 \end{array} \right] \quad (3.5)$$

3.1.2 Matriz de translação

A translação de um ponto no espaço cartesiano é a movimentação deste a uma certa distância em uma determinada direção. Essa distância e direção são representadas na forma de um vetor d . Em um espaço tridimensional esse vetor pode ser representado como $d = (d_x, d_y, d_z)$. Como visto anteriormente, para realizar uma transformação afim, é necessária uma matriz de transformação, no caso da translação, a componente X da matriz se torna uma matriz identidade,

e na Y é utilizado o vetor d . Com isso pode-se representar a translação de um ponto no espaço cartesiano tridimensional como:

$$\begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} \quad (3.6)$$

3.1.3 Matriz de rotação

A rotação em um espaço cartesiano rotaciona um ponto, ou um conjunto de pontos, em uma direção por um ângulo. No espaço tridimensional, existem 3 matrizes de rotação padrão, rotacionando o ponto nos eixos x , y e z . Rotações mais complexas podem ser realizadas a partir de multiplicações dessas matrizes. Diferente da translação, a rotação ocupa apenas a componente X da matriz, com a componente Y sendo composta de zeros. Em (3.7), (3.8) e (3.9) são exibidas as matrizes de rotação para um ângulo θ em um espaço tridimensional.

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.7)$$

$$R_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.8)$$

$$R_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.9)$$

3.1.4 Parâmetros de Denavit Hartenberg

Na área de cinemática robótica, Denavit e Hartenberg (DENAVID; HARTENBERG, 1955) (HARTENBERG; DENAVIT, 1964) desenvolveram uma convenção para a representação de ligações em uma cadeia cinemática, na forma de quatro parâmetros, que ficaram conhecidos como os parâmetros de *Denavit-Hartenberg* (DH). A partir desses parâmetros, pode-se construir uma matriz de transformação afim, que pode ser utilizada para a realização de cálculos de cinemática direta.

Existem duas principais notações para o uso dos parâmetros de DH, mas o mais utilizado na robótica atual é o definido por Craig (2005). Para calcular esses parâmetros, partimos das seguintes premissas:

- O eixo Z_i aponta para a direção do eixo da junta.
- O eixo X_i aponta no sentido da perpendicular entre os eixos Z_i e Z_{i+1} . Caso exista interseção entre os dois X_i é posicionado na normal do plano que contém Z_i e Z_{i+1} .
- O eixo Y_i completa o sistema de coordenadas segundo a regra da mão direita.

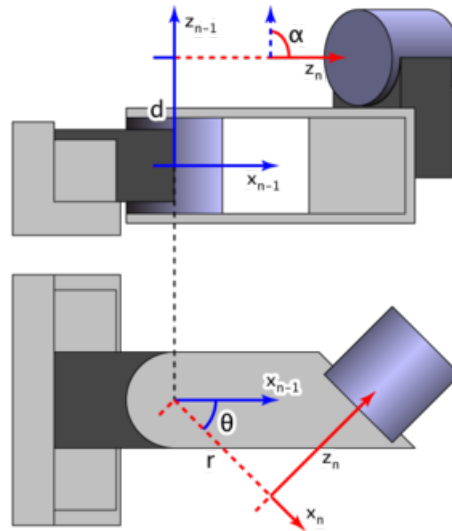
Com base nessas regras, os parâmetros de DH, representados na figura 6, são definidos como:

- d : A distância de X_i para X_{i-1} , seguindo o eixo Z_i
- a : A distância de Z_{i-1} para Z_i , seguindo o eixo X_i
- θ : O ângulo entre X_i e X_{i-1} , ao redor do eixo Z_i
- α : O ângulo entre Z_{i-1} e Z_i , ao redor do eixo X_i

Esses parâmetros são então aplicados como translações e rotações afins, construindo a matriz de transformação de DH, segundo a seguinte fórmula:

$$T_{DH} = R_x(\alpha)T_x(a)R_z(\theta)T_z(d) \quad (3.10)$$

Figura 6 – Parâmetros de DH.



Fonte – Retirado de Tekkotsu
1

Resultando na matriz:

$$\mathbf{T}_{DH} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & a \\ \sin \theta \cos \alpha & \cos \theta \cos \alpha & -\sin \alpha & -d \sin \alpha \\ \sin \theta \sin \alpha & \cos \theta \sin \alpha & \cos \alpha & d \cos \alpha \\ \hline 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.11)$$

3.2 CINEMÁTICA DIRETA

Como visto anteriormente, a cinemática robótica procura transformar o espaço de juntas em espaço cartesiano e vice-versa. A partir disso, a cinemática direta é responsável por, a partir dos valores angulares de cada junta do robô, obter a posição e orientação do ponto final de uma cadeia cinemática. A cinemática direta pode ser resolvida utilizando os parâmetros de DH.

Para realizar o cálculo da cinemática direta, primeiro é considerada a posição da junta inicial da cadeia em relação ao centro do eixo cartesiano, geralmente posicionado no centro de massa do robô. Para isso, é criada uma matriz de translação afim A_{Base} , utilizando o vetor entre esse ponto e o início da cadeia. Essa matriz é multiplicada por n matrizes DH, que representam cada junta da cadeia. Em seguida, são realizadas multiplicações por matrizes de rotação afim, para alinhar a orientação final com a orientação do eixo cartesiano, caso as juntas

¹ Disponível em <http://wiki.tekkotsu.org/index.php/Denavit-Hartenberg_conventions>; Acesso mai. 2017.

estejam em repouso. Por fim, caso exista alguma distância entre a última junta da cadeia e o manipulador, ela é representada pela matriz A_{End} .

$$T_{Base}^{End} = A_{Base}^{DH_0} T_{DH_0}^{DH_1} \dots T_{DH_{n-1}}^{DH_n} R_z(\theta_z) R_y(\theta_y) R_x(\theta_x) A_{DH_n}^{End} \quad (3.12)$$

3.2.1 Cinemática direta do robô NAO

Kofinas (2012) realiza em sua tese o cálculo da cinemática direta do robô NAO físico, e de todas as suas cadeias cinemáticas. Para nosso estudo, será focada apenas a cadeia cinemática das pernas dos agentes, que é facilmente convertida do robô físico para os modelos simulados, com exceção do tipo *toe*, que possui uma junta extra. Ele as define como:

$$T_{Base}^{End} = A_{Base}^0 T_0^1 T_1^2 T_2^3 T_3^4 T_4^5 T_5^6 R_z(\pi) R_y(-\frac{\pi}{2}) A_6^{End} \quad (3.13)$$

Com os valores para as matrizes de DH vistos nas tabelas 1 e 2.

Tabela 1 – Parâmetros de DH para cadeia de juntas da perna esquerda.

Frame (Junta)	a	α	d	θ
LHipYawPitch	0	$-\frac{3\pi}{4}$	0	$\theta_1 - \pi/2$
LHipRoll	0	$-\frac{\pi}{2}$	0	$\theta_2 + \frac{\pi}{4}$
LHipPitch	0	$\frac{\pi}{2}$	0	θ_3
LKneePitch	$-ThighLength$	0	0	θ_4
LAnklePitch	$-TibiaLength$	0	0	θ_5
LAnkleRoll	0	$-\frac{\pi}{2}$	0	θ_6

Fonte – Adaptado de (KOFINAS, 2012)

Tabela 2 – Parâmetros de DH para cadeia de juntas da perna direita.

Frame (Junta)	a	α	d	θ
RHipYawPitch	0	$-\frac{\pi}{4}$	0	$\theta_1 - \pi/2$
RHipRoll	0	$-\frac{\pi}{2}$	0	$\theta_2 - \frac{\pi}{4}$
RHipPitch	0	$\frac{\pi}{2}$	0	θ_3
RKneePitch	$-ThighLength$	0	0	θ_4
RAnklePitch	$-TibiaLength$	0	0	θ_5
RAnkleRoll	0	$-\frac{\pi}{2}$	0	θ_6

Fonte – Adaptado de (KOFINAS, 2012)

3.3 CINEMÁTICA INVERSA

Se a cinemática direta converte o espaço de juntas em coordenadas cartesianas, a cinemática inversa realiza o processo contrário. A partir de uma matriz afim, ou de um ponto no espaço com sua orientação, a cinemática inversa calcula os valores angulares objetivo para cada junta da cadeia, para que a junta final esteja nesse ponto. Dependendo da especificação da cadeia e da quantidade de juntas, é possível que exista mais de uma solução para a cinemática inversa, nesse caso, tal ambiguidade deve ser resolvida de acordo com o problema em questão.

Existem diversos métodos utilizados para encontrar, com diferentes graus de precisão, o espaço de juntas a partir do espaço cartesiano. Para esse trabalho será apresentar a solução geométrica.

3.3.1 Solução geométrica

Assim como a cinemática direta, já existem soluções numéricas para a cinemática inversa da cadeia de juntas da perna do robô NAO. Nesse trabalho será apresentada a solução geométrica apresentada em (FERREIRA et al., 2012), adaptada de (RÖFER et al., 2011). Da mesma maneira, essa solução não leva em consideração a junta extra do tipo *toe*.

O cálculo utiliza como entrada uma matriz de transformação afim T_{CoM}^{Foot} , representando a posição e orientação do pé do agente em relação a seu centro de massa ou *Center of Mass* (CoM). Para iniciar os cálculos, é necessário encontrar a matriz representando a posição do pé em relação ao início da cadeia, ou seja a primeira junta do quadril, T_{Hip}^{Foot} . Para lidar com a junta *YawPitch* do agente, toda a matriz é rotacionada 45° , resultando na seguinte operação, com l_{dist} sendo a distância entre as duas pernas:

$$T_{Hip}^{Foot} = R_x \left(\frac{\pi}{4} \right) T_y \left(\frac{l_{dist}}{2} \right) T_{Com}^{Foot} \quad (3.14)$$

Em seguida, a matriz é invertida, obtendo a posição do quadril em relação ao pé:

$$T_{Foot}^{Hip} = (T_{Hip}^{Foot})^{-1} \quad (3.15)$$

A partir do triângulo formado pelas partes da perna do agente (*LowerLeg* e *UpperLeg*), em conjunto com a matriz T_{Foot}^{Hip} , pode-se obter o ângulo da junta do joelho do agente a partir do suplementar do ângulo interno, calculado pela lei dos cossenos.

$$\theta_{knee} = \pi - \arccos\left(\frac{(l_{upperLeg})^2 + (l_{lowerLeg})^2 - (l_{T_{Foot}^{Hip}})^2}{2 * l_{upperLeg} * l_{lowerLeg}}\right) \quad (3.16)$$

Utilizando essa mesma regra é obtida a primeira componente da junta *Pitch* do tornozelo:

$$\theta_{footPitch_1} = \arccos\left(\frac{(l_{lowerLeg})^2 + (l_{T_{Foot}^{Hip}})^2 - (l_{upperLeg})^2}{2 * l_{lowerLeg} * l_{trans}}\right) \quad (3.17)$$

Para obter mais informações sobre as juntas são criados dois novos triângulos, projetando a posição do quadril no chão, e utilizando a posição final do pé, para criar triângulos retângulos nos planos *XZ* e *YX*. Com isso pode-se calcular o ângulo da junta *Roll* e a segunda componente da *Pitch* do tornozelo.

$$\theta_{footPitch_2} = \text{atan2}\left(x, \sqrt{y^2 + z^2}\right) \quad (3.18)$$

$$\theta_{footRoll} = \text{atan2}(y, z) \quad (3.19)$$

Onde x , y e z , são as componentes do vetor de translação da matriz T_{Foot}^{Hip} . Com as duas componentes da junta *Pitch*, o seu valor final pode ser calculado como:

$$\theta_{footPitch} = -(\theta_{footPitch_1} + \theta_{footPitch_2}) \quad (3.20)$$

Desse modo, as juntas da parte inferior da perna estão calculadas, restando apenas as três juntas do quadril. Para realizar esse cálculo, primeiro é criada uma matriz a partir dos

ângulos já calculados, e estes são multiplicados pela matriz T_{Foot}^{Hip} :

$$A = (R_x(-\theta_{footRoll})R_y(-\theta_{footPitch} - \theta_{knee}))^{-1} \quad (3.21)$$

$$T_{Hip} = A \cdot T_{Foot}^{Hip} \quad (3.22)$$

Com essa matriz, pode-se calcular o ângulo de todas as juntas do quadril, a partir da análise da matriz de rotação obtida pela multiplicação das três juntas conhecidas. Como foi realizada a rotação inicial por 45° , pode-se tratar as juntas do quadril como: $Yaw(z)$, $Roll(x)$ e $Pitch(y)$.

$$\mathbf{R}_{Hip} = \mathbf{R}_z(\theta_z)\mathbf{R}_x(\theta_x)\mathbf{R}_y(\theta_y) = \begin{bmatrix} c_y c_z - s_x s_y s_z & -c_x s_z & c_z s_y + c_y s_x s_z & 0 \\ c_z s_x s_y + c_y s_z & c_x c_z & -c_y c_z s_x + s_y s_z & 0 \\ -c_x c_y & s_x & c_x c_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.23)$$

A partir dessa matriz é notado que o ângulo θ_x pode ser calculado como $\arcsin((R_{Hip})_{3,2})$.

As outras juntas podem ser obtidas com cálculos como:

$$\frac{-(R_{Hip})_{1,2}}{(R_{Hip})_{2,2}} = \frac{\cos\theta_x * \sin\theta_z}{\cos\theta_x * \cos\theta_z} = \frac{\sin\theta_z}{\cos\theta_z} = \tan\theta_z \quad (3.24)$$

Desse modo, pode-se calcular as juntas $Pitch$ e Yaw como:

$$\theta_{hipYawPitch} = \theta_z = \text{atan2}(-(R_{Hip})_{1,2}, (R_{Hip})_{2,2}) \quad (3.25)$$

$$\theta_{hipPitch} = \theta_y = \text{atan2}(-(R_{Hip})_{3,1}, (R_{Hip})_{3,3}) \quad (3.26)$$

Por fim, a rotação inicial feita no eixo x por 45° é compensada, obtendo o ângulo da junta *Roll*:

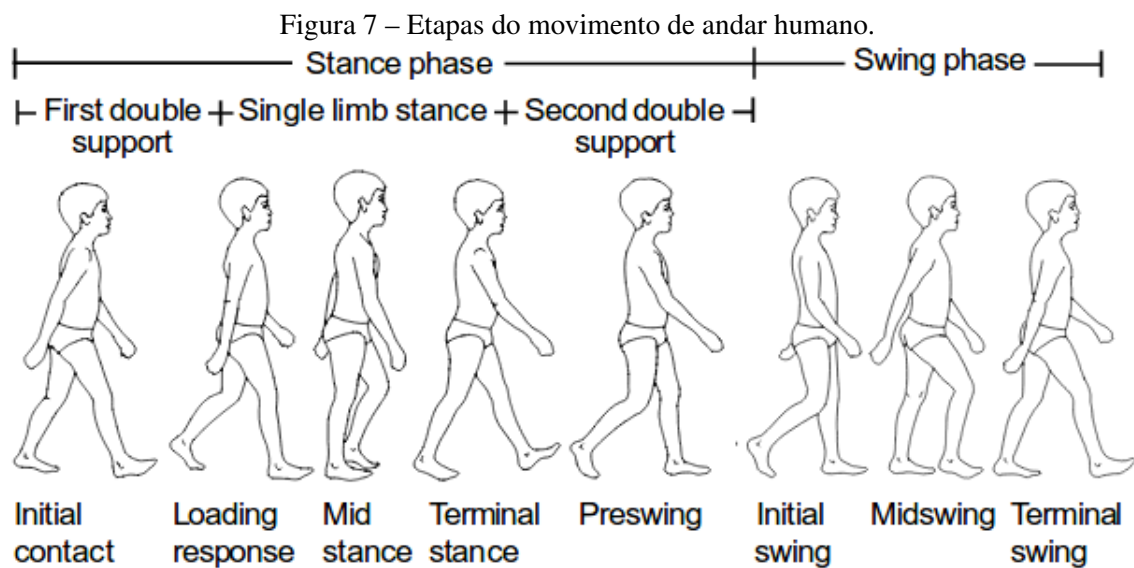
$$\theta_{hipRoll} = \theta_x - \frac{\pi}{4} \quad (3.27)$$

Obtendo, assim, o valor do ângulo de todas as juntas da cadeia com base na posição da última junta em relação ao CoM. Os valores das juntas *YawPitch* e *Roll* precisarão ser ajustados dependendo se a perna calculada foi a esquerda ou a direita.

4 MOVIMENTO ANDAR EM ROBÔS HUMANOIDES

A pesquisa no desenvolvimento do movimento do andar em robôs humanoides é uma área de grande interesse no campo da robótica. Diversos trabalhos já foram desenvolvidos e novos ainda estão em andamento (KAJITA et al., 2001) (MACALPINE et al., 2012) (GLASER; DORER, 2013). Robôs humanoides são criados procurando imitar o ser humano tanto em aparência quanto em comportamentos.

O movimento de caminhada humano é composto de diversas etapas, que podem ser vistas na figura 7. Para a execução em agentes humanoides, essas etapas são simplificadas utilizando como base onde o peso do corpo do agente está equilibrado. Dessa forma, o agente intercala entre duas etapas. Uma delas é a etapa de suporte único, quando apenas um dos pés toca o chão enquanto o outro balança do ponto anterior para o seguinte. A outra etapa, de suporte duplo, é caracterizada pelo agente ter seu peso equilibrado nos dois pés.



Fonte – Retirado de (VAUGHAN et al., 1992)

Nesse capítulo, serão discutidas medidas de validação de estabilidade do movimento andar, assim como dois dos principais modelos para a execução desse movimento.

4.1 MEDIDAS DE ESTABILIDADE DO MOVIMENTO

O movimento dinâmico de caminhada possui grande complexidade e devido a isso é suscetível a diversas forças, que podem prejudicar sua estabilidade. Para poder avaliar

numericamente o grau de estabilidade desse movimento, foram criadas algumas medidas. Essas medidas podem ser utilizadas tanto para avaliar a qualidade do movimento executado, como para criar um controlador para o movimento. Para esse trabalho, serão focadas duas principais medidas, a projeção do CoM, *Ground projection of Center of Mass* (GCoM) e o *Zero Moment Point* (ZMP).

4.1.1 Projeção do centro de massa

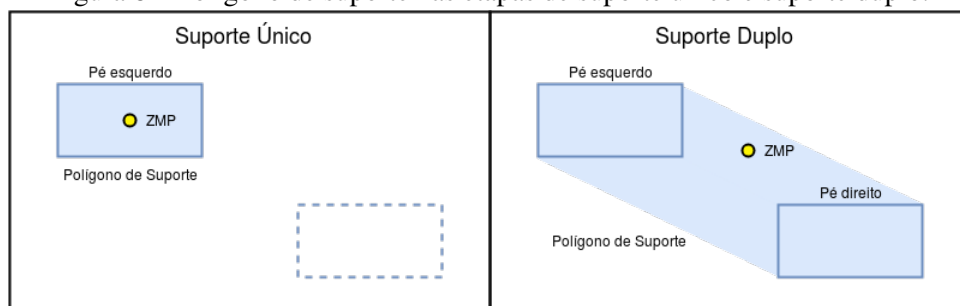
A projeção do centro de massa de um robô no chão pode ser facilmente utilizada para determinar o grau de estabilidade do movimento do agente. O GCoM pode ser calculado a partir das fórmulas apresentadas por Shafii (2015):

$$m = \sum_{i=1}^n m_i \quad (4.1)$$

$$OG = \frac{1}{m} \sum_{i=1}^n OG_i \cdot m_i \quad (4.2)$$

Onde m_i é a massa da parte i do corpo do agente, e OG_i determina a posição desta mesma parte no sistema de coordenadas O e é localizada no plano do chão. No estudo de medidas de estabilidade do movimento, um conceito bastante utilizado é o do polígono de suporte, visto na figura 8. Durante a etapa de suporte único do movimento do robô, o polígono de suporte se equivale à superfície do pé que está tocando o chão. Com base nisto, afirma-se que um agente cuja GCoM esteja sempre dentro do polígono de suporte é definitivamente estável.

Figura 8 – Polígono de suporte nas etapas de suporte único e suporte duplo.



Fonte – O autor

Entretanto, embora a GCoM seja boa para calcular a estabilidade de um robô estático,

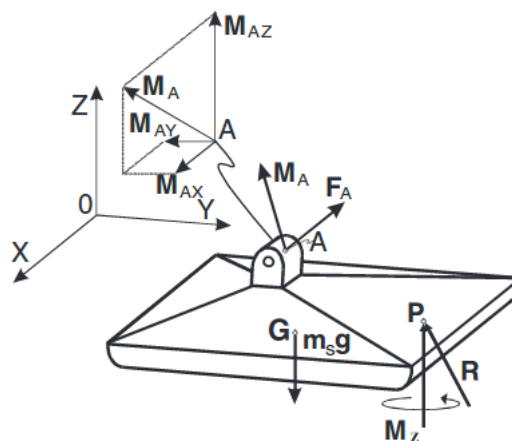
durante seu movimento, é muito comum que essa projeção saia do polígono de suporte. Uma outra medida para verificar a estabilidade do agente, e uma das melhores conhecidas na área de locomoção robótica, é o ZMP, visto na seção seguinte.

4.1.2 Zero Moment Point

O Ponto de Momento Zero, ZMP, foi apresentado inicialmente por Vukobratovic e Juricic (1969). A partir do cálculo da posição do ZMP diversos modelos de locomoção foram desenvolvidos. Por definição, o ZMP é o ponto onde a influência de todas as forças atuando no mecanismo podem ser substituídas por uma única força (VUKOBRATOVIĆ; BOROVAC, 2004). Além disso, de acordo com Harada et al. (2006), o momento tangencial ao redor desse ponto, possui valor nulo, simplificando cálculos de estabilidade.

Na figura 9 são vistas as forças que agem sobre o pé de suporte do agente. Simplificando o mecanismo acima do tornozelo, pode-se representar sua influência como a força F_A , e o momento M_A . A força da gravidade também age sobre o pé, atuando no centro de massa e representada por G . A reação do chão sobre o pé é então representada no ponto P . A reação final do chão é composta pela força R e momento M .

Figura 9 – Forças atuantes no pé de suporte de um agente, resultando no ZMP.



Fonte – Retirado de (VUKOBRATOVIĆ; BOROVAC, 2004)

Com base nesses valores, e assumindo que o pé está em completo repouso sobre o chão, as fórmulas do equilíbrio estático para o pé de suporte são definidas em (VUKOBRATO-

VIĆ; BOROVAC, 2004) como:

$$R + F_A + m_S g = 0 \quad (4.3)$$

$$\vec{OP} \times \vec{R} + \vec{OG} \times m_S g + M_A + M_Z + \vec{OA} \times F_A = 0 \quad (4.4)$$

O ZMP é afetado pela massa e inércia de cada parte do corpo do agente. Portanto, o ponto em si, localizado no eixo cartesiano projetado no chão, é calculado por Vukobratovic et al. (2012) como:

$$p_x = \frac{\sum_{i=1}^n m_i (\ddot{z}_i + g) x_i - \sum_{i=1}^n m_i \ddot{x}_i z_i - \sum_{i=1}^n I_{iy} \ddot{\Omega}_{iy}}{\sum_{i=1}^n (\ddot{z}_i + g) m_i} \quad (4.5)$$

$$p_y = \frac{\sum_{i=1}^n m_i (\ddot{z}_i + g) y_i - \sum_{i=1}^n m_i \ddot{y}_i z_i - \sum_{i=1}^n I_{ix} \ddot{\Omega}_{ix}}{\sum_{i=1}^n (\ddot{z}_i + g) m_i} \quad (4.6)$$

Para ambas as equações, x_i , y_i e z_i representam a posição do centro de massa da parte do corpo i do robô. Do mesmo modo, m_i é a massa da respectiva parte do corpo e g é a constante gravitacional. I_{ix} e Ω_{ix} são, respectivamente, o centroide do momento de inércia e o deslocamento angular no eixo x . Logo, I_{iy} e Ω_{iy} representam os mesmos valores, no eixo y . Com base no princípio do polígono de suporte, pode-se garantir a estabilidade de um certo movimento caso o ZMP se encontre sempre dentro do polígono de suporte, uma vez que quando ele se aproxima das bordas do mesmo, o corpo do agente pode rotacionar ao redor da borda do pé.

4.2 MODELOS DO MOVIMENTO ANDAR

Como dito anteriormente, vários modelos de movimentação já foram criados para o movimento de caminhada. Dentre os diversos modelos já desenvolvidos, serão discutidos

dois modelos que utilizam como base o ZMP. O modelo *Cart-Table*, cuja versão modificada é atualmente utilizada pelo time BahiaRT, e o modelo do pêndulo invertido.

4.2.1 Modelo *Cart-Table*

Nesse modelo, durante a fase de suporte único do movimento de andar, o deslocamento e equilíbrio do centro de massa pode ser representado pelo modelo *Cart-Table* (KAJITA et al., 2003). Esse modelo descreve o movimento como um carro de massa m , representando o CoM do agente, equilibrado sobre uma mesa de massa desprezível. A figura 10 apresenta o modelo representado sobre o robô NAO.

Figura 10 – O modelo *Cart-Table* aplicado no robô NAO.



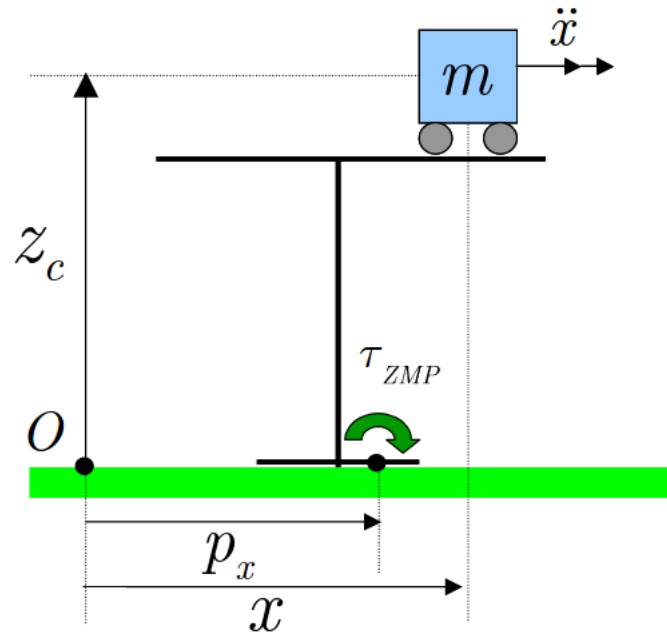
Fonte – Retirado de (SHAFII, 2015)

Para executar a modelagem de um movimento tridimensional, são calculados dois conjuntos do modelo *Cart-Table*. Um deles responsável pela movimentação no plano sagital e o outro para o movimento no plano coronal. Na figura 11 pode-se ver a representação do modelo no plano sagital, ou eixo XZ.

O centro de massa é representado pela massa m . x e z_c são valores com relação ao sistema de coordenadas cartesianas com relação à origem. A movimentação do carro sobre a mesa, representada pela aceleração \ddot{x} , em conjunto com o efeito da gravidade g gera um torque ao redor do ponto T_{ZMP} , que representa o ponto onde a pressão do peso do carro está sendo aplicado no chão.

Entretanto, de acordo com o que vimos na seção sobre o ZMP, para estabelecer

Figura 11 – Visualização esquemática do modelo *Cart-Table*.



Fonte – Retirado de (KAJITA et al., 2003)

o equilíbrio, o movimento deve manter o ZMP dentro do polígono de suporte, nesse caso representado pelo pé da mesa. Assumindo esse ponto P como o ZMP, Kajita et al. (2003) conclui que o momento angular, ou o torque nesse ponto deve ter valor nulo, chegando à fórmula:

$$T_{ZMP} = mg(x - p_x) - m\ddot{x}z_c = 0 \quad (4.7)$$

Utilizando então essa equação como base, Shafii (2015) deriva o ZMP como:

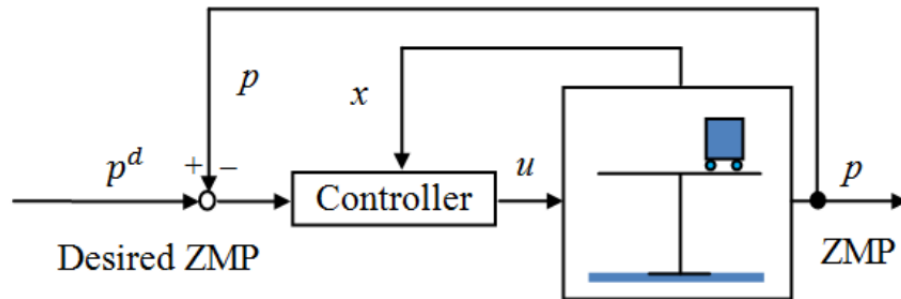
$$p_x = x - \frac{z_c}{g}\ddot{x} \quad (4.8)$$

$$p_y = y - \frac{z_c}{g}\ddot{y} \quad (4.9)$$

Uma vez definido o ZMP desejado durante a execução do movimento, é necessário calcular a posição desejada do CoM com base nesses pontos. A partir da geração do centro de massa, a cinemática inversa pode ser utilizada para executar o movimento. Kagami et al. (2002) propõe uma solução numérica relacionando a trajetória do ZMP com a trajetória do CoM.

Entretanto, a estratégia mais aceita e utilizada para a caminhada bípede é a proposta em (KAJITA et al., 2003). O método proposto por Kajita et al. (2003) é o controlador por previsão do ZMP.

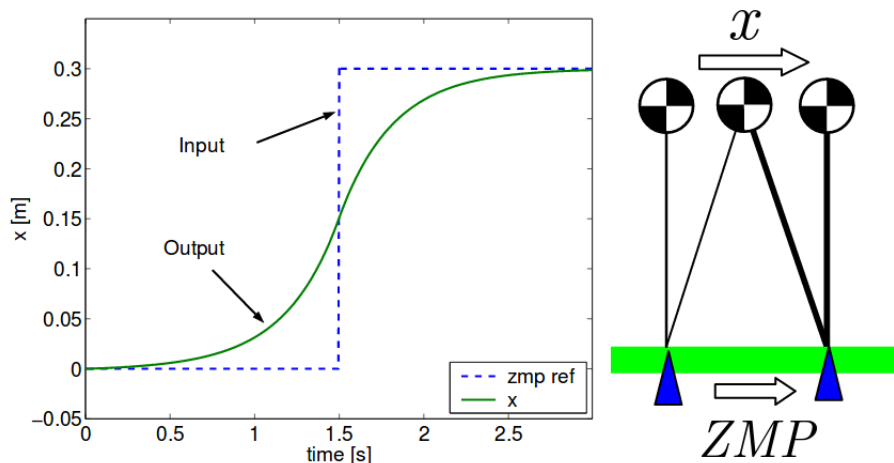
Figura 12 – Diagrama do controle de previsão do ZMP.



Fonte – Retirado de (SHAFII, 2015)

Kajita et al. (2003) trata a geração do centro de massa com base no ZMP como o problema de controle de um servo motor, utilizando como entrada do controlador a derivada da aceleração calculada pelo modelo *Cart-Table*, em função do tempo, representada como $u = \ddot{x} = \frac{d\dot{x}}{dt}$. A figura 12 apresenta o controlador digital do ZMP. Esse modelo, então, utiliza o modelo de controlador proposto em (KATAYAMA et al., 1985) para obter uma previsão de trajetória do CoM robusta. Park e Youm (2007) apresenta o cálculo detalhado do modelo completo. A figura 13 ilustra as trajetórias ideais para o ZMP e CoM, com a trajetória do CoM calculada com base no controlador de previsão do ZMP.

Figura 13 – Trajetórias do ZMP e do CoM.



Fonte – Retirado de (KAJITA et al., 2003)

4.2.2 Modelo do pêndulo invertido

Uma das desvantagens do modelo *Cart-Table* apresentado anteriormente é o fato de que, devido às abstrações do modelo, o CoM se mantém sempre na mesma altura, não variando durante o movimento. Essa invariação da altura resulta em um movimento onde o joelho do robô deve ser dobrado durante o movimento para manter a altura do quadril, resultando em um torque que pode até dobrar o custo energético do movimento (KUO; DONELAN, 2010). Nessa seção, será discutido um modelo que evita esse problema, o modelo do pêndulo invertido, *Inverted Pendulum* (IP).

Apesar do modelo do pêndulo invertido simples, onde o CoM é considerado um ponto equilibrado em uma vara de massa insignificante ser amplamente utilizado para representar o movimento de andar humano (SAKKA et al., 2010), para esse trabalho será apresentado um modelo mais visto na literatura quanto à robótica, cuja massa é equilibrada em uma vara telescópica, com altura variável. Assim como o modelo *Cart-Table*, para simplificação a massa é considerada completamente concentrada no tronco do robô, e a perna de suporte não possuindo massa. A figura 14 apresenta este modelo aplicado no robô NAO.

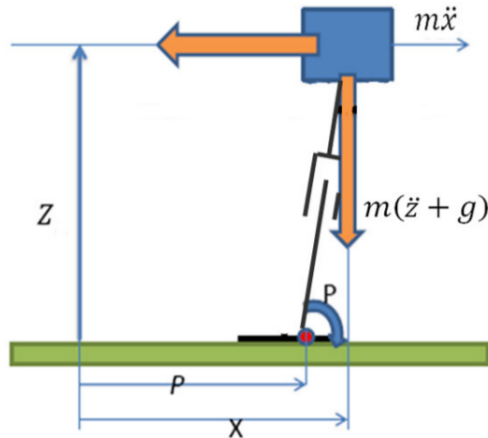
Figura 14 – O modelo do pêndulo invertido com vara telescópica aplicado ao robô NAO.



Fonte – Retirado de (SHAFII, 2015)

Na figura 15, pode-se ver um representação esquemática do modelo no plano sagital. Assim como para o modelo *Cart-Table* foram usados dois conjuntos para representar a movimentação tridimensional, são modelados dois pêndulos invertidos, um no eixo XZ e outro no eixo YZ .

Figura 15 – Visualização esquemática do modelo do pêndulo invertido.



Fonte – Retirado de (SHAFII, 2015)

A massa do robô é representada por m , com suas acelerações nos eixos X e Z representadas como \ddot{x} e \ddot{z} , respectivamente. z e x são as distâncias do centro de massa para a origem do eixo de coordenadas, enquanto o ponto P representa o ponto onde a pressão da massa age sobre o chão, com P_x sendo a distância do mesmo para a origem no eixo X . Shafii (2015) define o momento T_P ao redor do ponto P como:

$$T_P = m(g + \ddot{x})(x - P_x) - m\ddot{x}z \quad (4.10)$$

Sabendo que o robô está em uma posição equilibrada se o momento a redor do ponto onde a pressão está sendo aplicada é zero (TAJIMA et al., 2009), o ponto P é então considerado o ZMP, assumindo $T_P = 0$. As equações abaixo apresentam o cálculo do mesmo no plano sagital e coronal.

$$P_x = x - \frac{z}{g + \ddot{z}}\ddot{x} \quad (4.11)$$

$$P_y = y - \frac{z}{g + \ddot{z}}\ddot{y} \quad (4.12)$$

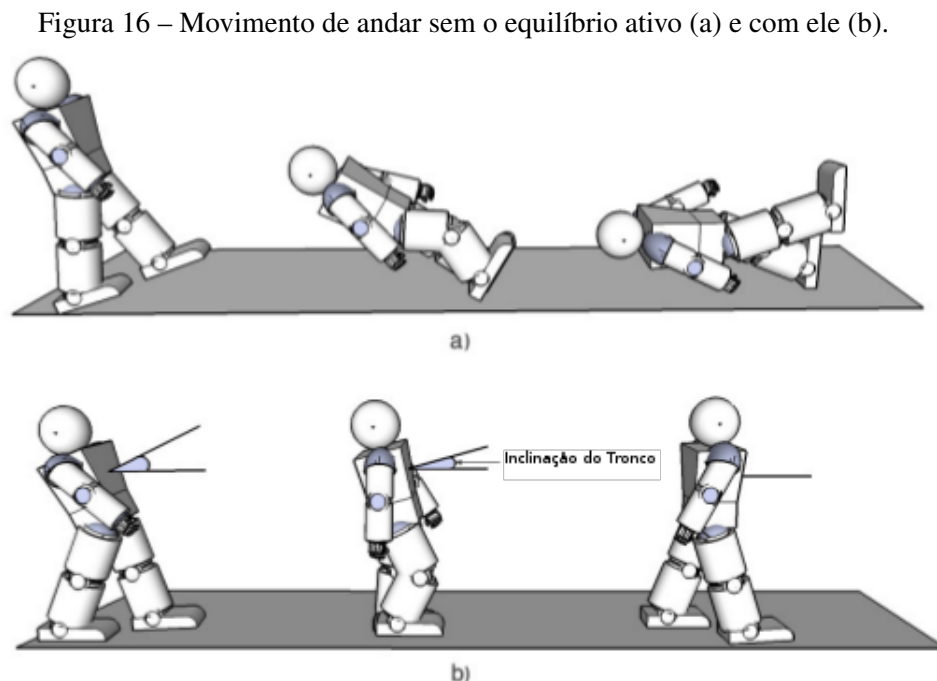
Uma vez calculado o ZMP para o pé de suporte durante o movimento, é necessário calcular a posição do centro de massa com relação ao pé e em seguida, utilizar a cinemática

inversa para definir os valores angulares para cada junta durante o movimento. Para realizar o cálculo da trajetória do CoM, Shafii et al. (2014) a divide em duas etapas, a geração da componente vertical do CoM e a geração da componente horizontal. O cálculo utilizado para a geração do CoM pode ser visto em Shafii et al. (2014).

4.2.3 Equilíbrio ativo com retroalimentação

Uma vez calculado a posição dos pés durante o movimento de andar com relação ao CoM, é necessário realizar os cálculos relativos à orientação do mesmo. Devido às simplificações realizadas pelos modelos matemáticos, e aos erros na execução do movimento, não pode-se garantir que a orientação calculada pelo modelo é suficiente para manter o equilíbrio do robô.

Com isso em mente Shafii (2015) propõe um ciclo de retroalimentação para manter o equilíbrio do agente durante o movimento, corrigindo falhas na orientação do corpo do mesmo. A partir do sensor *GyroRate* do agente, a inclinação de seu corpo é calculada e utilizada no cálculo da orientação do pé, compensando essa inclinação para retornar o corpo à posição perpendicular ao chão. A figura 16 apresenta uma sequência do andar com e sem o equilíbrio ativo.



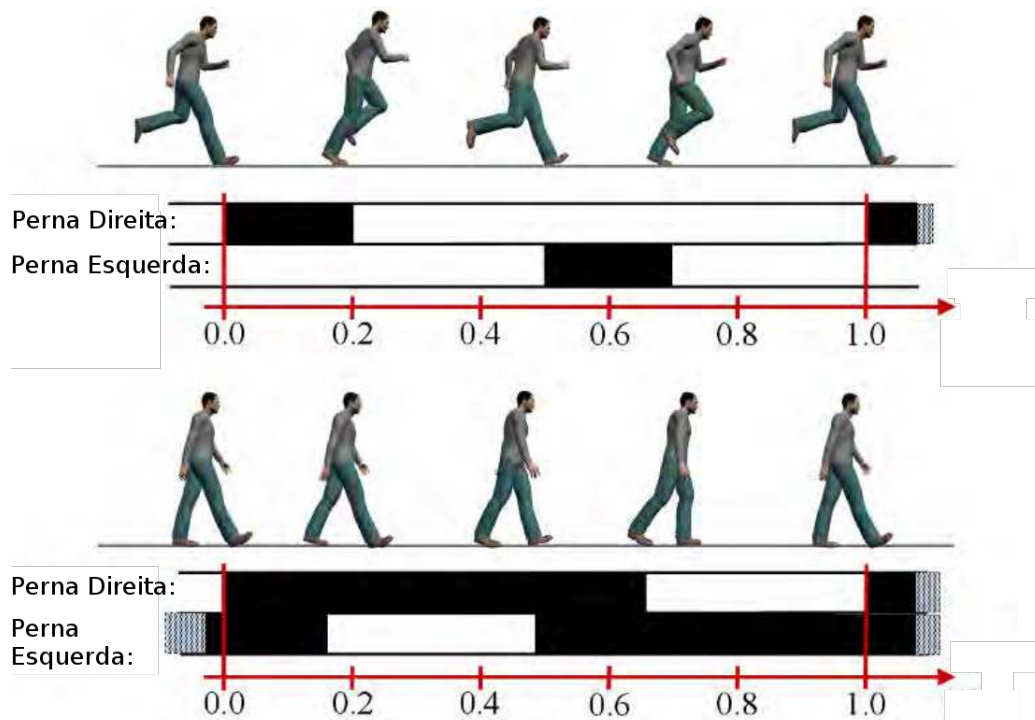
Fonte – Adaptado de (SHAFII, 2015)

5 MODELOS PARA O MOVIMENTO DE CORRER

Considerando que o movimento de caminhada bípede já apresentou diversas soluções dinâmicas com resultados positivos, Kajita et al. (2007) pergunta "Podemos construir um robô capaz de correr?". Como vimos no capítulo 4 robôs humanoides são criados para imitar o ser humano em seus comportamentos, portanto, o movimento de corrida também faz parte desse conjunto.

Assim como o andar, o movimento de corrida se utiliza de simplificações das etapas do movimento humano. Similar à etapa de suporte único da caminhada, a corrida possui uma etapa onde todo o peso do corpo está equilibrado em um pé, conhecida como a fase de postura. Diferente do andar, entretanto, a corrida não possui uma fase similar à etapa de suporte duplo, tendo em seu lugar uma fase de voo. Durante essa fase, nenhuma parte do corpo toca o chão, movimentando-se por pura inércia. A figura 17 apresenta uma comparação entre as etapas dos movimentos de corrida e caminhada. O movimento de corrida pode ser então resumido como uma "locomoção bípede com fase de voo" Kajita et al. (2007).

Figura 17 – Comparação entre as etapas dos movimentos de corrida(acima) e caminhada(abaixo).



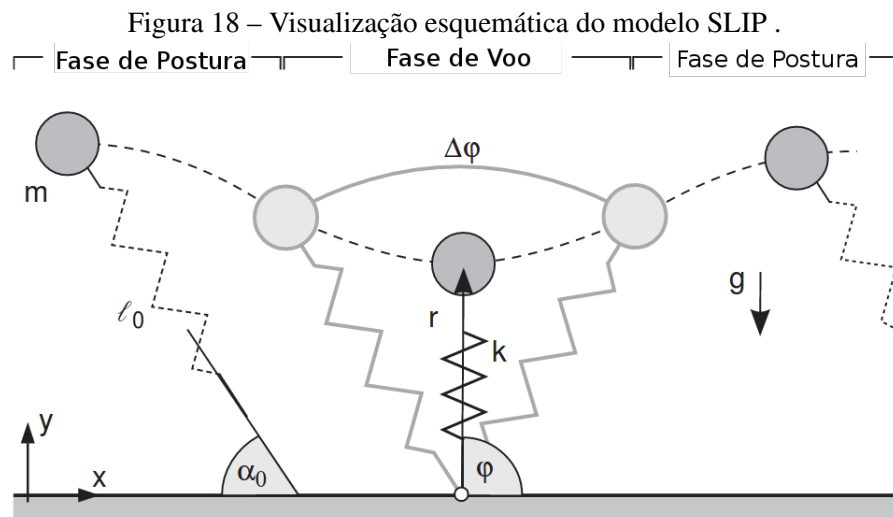
Fonte – Adaptado de (FIHL; MOESLUND, 2010)

Nesse capítulo serão discutidas abstrações do movimento de corrida e modelos para

a execução do mesmo em robôs.

5.1 MODELO *SPRING-MASS*

O principal modelo utilizado para simular o movimento de corrida é o modelo *Spring-Mass* (BLICKHAN, 1989), também conhecido como o modelo do pêndulo invertido montado em mola, *Spring-Loaded Inverted Pendulum* (SLIP). Nesse modelo, o agente é representado por uma massa puntiforme, equivalente ao centro de massa do robô, equilibrado sobre uma mola sem massa. Durante a fase de postura, quando a mola toca o chão, ela se contrai passivamente gerando energia potencial, que é liberada propulsionando a massa e iniciando a etapa de voo. Durante a etapa de voo, o movimento do agente se assemelha a uma trajetória de voo de um projétil. A figura 18 apresenta uma visualização esquemática do modelo.



A massa é representada por m enquanto a mola apresenta seu comprimento inicial l_0 e coeficiente k . Durante a fase de postura, quando a mola é comprimida, seu valor na instância de tempo t é representada por r . Também durante essa etapa, o ângulo entre a mola e o chão é representado por φ , com sua variação $\Delta\varphi$. No momento de transição entre a etapa de voo e de postura, quando a mola faz contato com o chão, o seu ângulo formado é representado por α_0 .

5.1.1 Solução aproximada para CoM

Geyer et al. (2005) propõe uma solução aproximada para o sistema de massa e mola, partindo de algumas aproximações, para simplificar seus cálculos. A partir dessas aproximações

é possível obter uma solução dinâmica com taxa de erro baixa dentro das condições assumidas pelas aproximações. Na figura 18 pode-se ver a notação dos valores utilizados em seus cálculos. Utilizando as coordenadas polares (r, φ) , a radial e o ângulo polar respectivamente, as mecânicas de Lagrange podem ser aplicadas como:

$$L = \frac{m}{2}(\dot{r}^2 + r^2\dot{\varphi}^2) - \frac{k}{2}(\ell_0 - r)^2 - mgr \cos(\varphi) \quad (5.1)$$

A partir dela, pode-se obter as equações de movimento da fase de postura como (YU et al., 2012):

$$m\ddot{r} + k(r - \ell_0) + mg \cos(\varphi) - mr\dot{\varphi}^2 = 0 \quad (5.2)$$

$$\frac{d}{dt}(mr^2\dot{\varphi}) - mgr \sin(\varphi) = 0 \quad (5.3)$$

Com essas formulas como base, Geyer et al. (2005) faz a primeira suposição para a sua solução. Para variações angulares $\Delta\varphi$ suficientemente pequenas, o ângulo φ pode ser assumido como 0, resultando nas aproximações $\sin(\varphi) \approx 0$ e $\cos(\varphi) \approx 1$, simplificando as equações 5.2 e 5.3 como:

$$m\ddot{r} = k(\ell_0 - r) + mr\dot{\varphi}^2 - mg \quad (5.4)$$

$$\frac{d}{dt}(mr^2\dot{\varphi}) = 0 \quad (5.5)$$

Como resultado dessa aproximação, a energia E e o momento angular P do movimento são conservados. Com $P = mr^2\dot{\varphi}$ e:

$$E = \frac{m}{2}\dot{r}^2 + \frac{P^2}{2mr^2} + \frac{k}{2}(\ell_0 - r)^2 + mgr \quad (5.6)$$

Aplicando as substituições $\varepsilon = \frac{2E}{m\ell_0^2}$, $\omega = \frac{P}{m\ell_0^2}$, $\rho = \frac{r-\ell_0}{\ell_0} \leq 0$ e $\omega_0 = \sqrt{\frac{k}{m}}$ obtém-se:

$$\varepsilon = \dot{\rho}^2 + \frac{\omega^2}{(1+\rho)^2} + \omega_0^2 \rho^2 + \frac{2g}{\ell_0}(1+\rho) \quad (5.7)$$

Aqui é aplicada a segunda suposição. Em seu modelo, Geyer et al. (2005) supõe que ρ , que representa a porcentagem de compressão da mola, nunca chegará a um valor muito alto. Com $|\rho| \ll 1$, o termo $\frac{1}{(1+\rho)^2}$ pode ser representado como uma expansão de Taylor próxima a zero como:

$$\frac{1}{(1+\rho)^2} \Big|_{\rho=0} = 1 - 2\rho + 3\rho^2 - O(\rho^3) \quad (5.8)$$

Utilizando essa suposição e a equação 5.7 pode ser escrita uma integral como:

$$t = \int \frac{d\rho}{\sqrt{\lambda\rho^2 + \mu\rho + \nu}} \quad (5.9)$$

Com $\lambda = -(3\omega^2 + \omega_0^2)$, $\mu = 2(\omega^2 - g/\ell_0)$ e $\nu = (\varepsilon - \omega^2 - 2g/\ell_0)$. Essa integral pode ser resolvida como:

$$\int \frac{d\rho}{\sqrt{\lambda\rho^2 + \mu\rho + \nu}} = -\frac{1}{\sqrt{-\lambda}} \arcsin\left(\frac{2\lambda\rho + \mu}{\sqrt{\mu^2 - 4\lambda\nu}}\right) \quad (5.10)$$

Desde que duas condições sejam atendidas, λ seja negativo e $\mu^2 - 4\lambda\nu$ seja positivo. Por definição, pode-se ver que a primeira condição é atendida, λ é sempre negativo. Desse modo a segunda condição passa a ser atendida quando ν é positiva, sendo uma constante, basta que isso seja checado no momento em que a mola fizer contato com o chão. Com essa solução para a integral, a equação geral para a radial é obtida como:

$$r(t) = \ell_0(1 + a + b \sin(\hat{\omega}_0 t)) \quad (5.11)$$

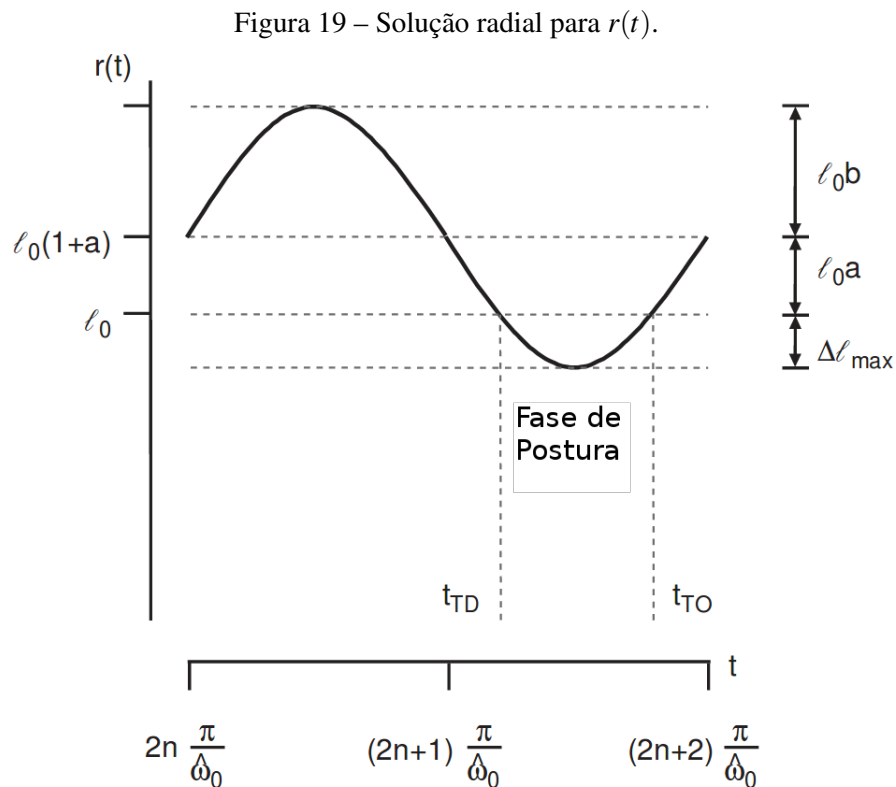
Onde:

$$\hat{\omega}_0 = \sqrt{\omega_0^2 + 3\omega^2} \quad (5.12)$$

$$a = \frac{\omega^2 - g/\ell_0}{\hat{\omega}_0^2} \quad (5.13)$$

$$b = \sqrt{a^2 + \frac{\varepsilon - \omega^2 - 2g/\ell_0}{\hat{\omega}_0^2}} \quad (5.14)$$

A figura 19 apresenta a movimentação da radial em função do tempo. Os valores de tempo t_{TD} e t_{TO} representam os momentos de transição entre a etapa de voo para a de postura e de volta para a de voo, respectivamente (*Touch-Down* e *Take-Off*). As equações 5.15 e 5.16 apresentam essas instâncias de tempo, calculadas com base na equação da radial, com n como um inteiro arbitrário.



Fonte – Adaptado de (GEYER et al., 2005)

$$t_{TD} = \frac{1}{\hat{\omega}_0} \left\{ \left(2n + \frac{3}{2} \right) \pi - \left[\frac{\pi}{2} + \arcsin \left(-\frac{a}{b} \right) \right] \right\} \quad (5.15)$$

$$t_{TO} = \frac{1}{\hat{\omega}_0} \left\{ \left(2n + \frac{3}{2} \right) \pi + \left[\frac{\pi}{2} + \arcsin \left(-\frac{a}{b} \right) \right] \right\} \quad (5.16)$$

Além de encontrar a radial, para o sistema de coordenadas polares do movimento é necessário também resolver a equação do movimento angular. A movimentação angular pode ser obtida com base na radial $r(t)$, uma vez que devido às aproximações, o momento angular é conservado (ARSLAN et al., 2009). Desse modo, a velocidade angular pode ser obtida como:

$$\dot{\varphi} = \frac{\omega}{(1+\rho)^2} \quad (5.17)$$

Novamente, é utilizada a expansão de Taylor para aproximar linearmente o termo $\frac{1}{(1+\rho)^2}$, resultando em $\dot{\varphi} = \omega(1 - 2\rho)$. Substituindo ρ por r , pode-se obter a variação angular durante a fase de postura como:

$$\Delta\varphi = \int_{t_{TD}}^{t_{TO}} \omega[(1 - 2a) - 2b \sin(\hat{\omega}_0 t)] dt \quad (5.18)$$

Que pode ser resolvida como:

$$\Delta\varphi = 2 \frac{\omega}{\hat{\omega}_0} [(1 - 2a) \arccos(a/b) + 2\sqrt{b^2 - a^2}] \quad (5.19)$$

Devido às suposições e aproximações feitas até o momento, a energia mecânica e momento angular são conservados, a variação angular $\Delta\varphi$ pode ser completamente determinada pelos parâmetros do movimento no momento de *Touch-Down* ($\ell_0, \dot{r}_{TD}, \varphi_{TD}, \dot{\varphi}_{TD}$) e os parâmetros do sistema (k, m, g). O ângulo φ_{TD} pode ser obtido como $\varphi_{TD} = \pi - \alpha_0$.

Definindo t_{TD} como o tempo 0 do movimento, as fórmulas da radial 5.11 e angular 5.19 podem ser reescritas como:

$$r(t) = \ell_0 + \ell_0[a(1 - \cos(\hat{\omega}_0 t)) - \sqrt{b^2 - a^2} \sin(\hat{\omega}_0 t)] \quad (5.20)$$

$$\varphi(t) = \varphi_{TD} + (1 - a)\omega t + \frac{2\omega}{\hat{\omega}_0}[a \sin(\hat{\omega}_0 t) + \sqrt{b^2 - a^2}(1 - \cos(\hat{\omega}_0 t))] \quad (5.21)$$

Substituindo os valores predefinidos a , b , ε , ω e ω_0 , o sistema para obter a trajetória do centro de massa é resolvido como:

$$r(t) = \ell_0 - \frac{|\dot{r}_{TD}|}{\hat{\omega}_0} \sin(\hat{\omega}_0 t) + \frac{\dot{\varphi}_{TD}^2 \ell_0 - g}{\hat{\omega}_0^2} (1 - \cos(\hat{\omega}_0 t)) \quad (5.22)$$

$$\varphi(t) = \varphi_{TD} + \left(1 - 2 \frac{\dot{\varphi}_{TD}^2 - g/\ell_0}{\hat{\omega}_0^2}\right) \dot{\varphi}_{TD} t + \frac{2\dot{\varphi}_{TD}}{\hat{\omega}_0} \left[\frac{\dot{\varphi}_{TD}^2 - g/\ell_0}{\hat{\omega}_0^2} \sin(\hat{\omega}_0 t) + \frac{|\dot{r}_{TD}|}{\hat{\omega}_0 \ell_0} (1 - \cos(\hat{\omega}_0 t)) \right] \quad (5.23)$$

Em função de t , que varia de 0 a t_s , definido abaixo, e com a frequência de oscilação definida como $\hat{\omega}_0 = \sqrt{k/m + 3\dot{\varphi}_{TD}^2}$.

$$t_s = \frac{1}{\hat{\omega}_0} \left[\pi + 2 \arctan \left(\frac{g - \ell_0 \dot{\varphi}_{TD}^2}{|\dot{r}_{TD}| \hat{\omega}_0} \right) \right] \quad (5.24)$$

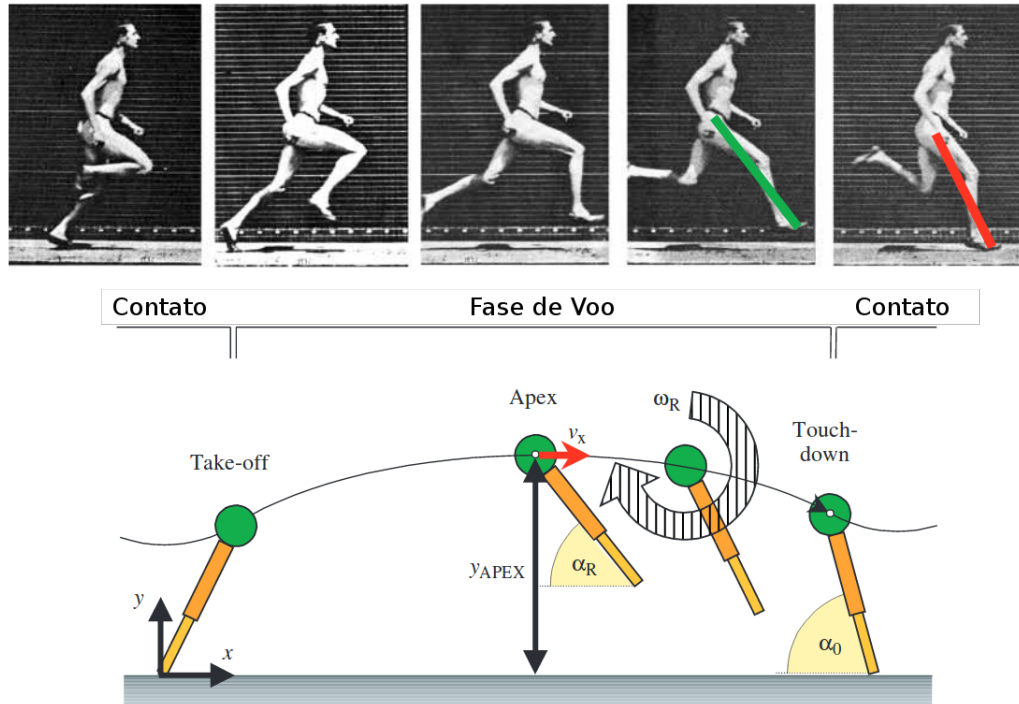
5.1.2 Controle da perna de balanço

Como foi descrito anteriormente, o movimento de corrida é composto pelas etapas de postura e voo. Durante a etapa de voo, a próxima perna a realizar o contato com o chão deve se posicionar para que esse contato tenha um ângulo de ataque α_0 . Em um agente bípede, a perna realizando o movimento é chamada de perna de balanço ou perna de *swing*.

Muitos estudos dentro do modelo de massa e mola, por efeito de simplificação e devido a complexidade do movimento da perna de postura, consideram essa perna de balanço

como estática, fixada no ângulo alvo α_0 e não considerando sua influência no movimento. Entretanto, Seyfarth et al. (2003) propõe o uso de uma perna de retração para estabilizar o movimento, utilizando como inspiração o movimento humano. Na figura 20 pode-se ver o modelo proposto e o movimento humano do qual ele é inspirado.

Figura 20 – Comparação entre o movimento da perna de balanço em um humano e o modelo proposto por Seyfarth et al. (2003).



Fonte – Adaptado de (SEYFARTH et al., 2003)

A altura máxima alcançada pelo agente durante sua fase de voo é representada por y_{APEX} , com o ângulo α_R , sendo menor do que o ângulo α_0 desejado para o momento de *Touch-Down*. Desse modo, ω_R é a velocidade necessária para variar de α_R para α_0 dentre o tempo em que ele se encontra em y_{APEX} , t_{APEX} e o momento do *Touch-Down*. Seguindo o sistema:

$$\alpha(t) = \begin{cases} \text{for } t < t_{APEX}, \alpha_R \\ \text{for } t \geq t_{APEX}, \alpha_R + \omega_R(t - t_{APEX}) \end{cases} \quad (5.25)$$

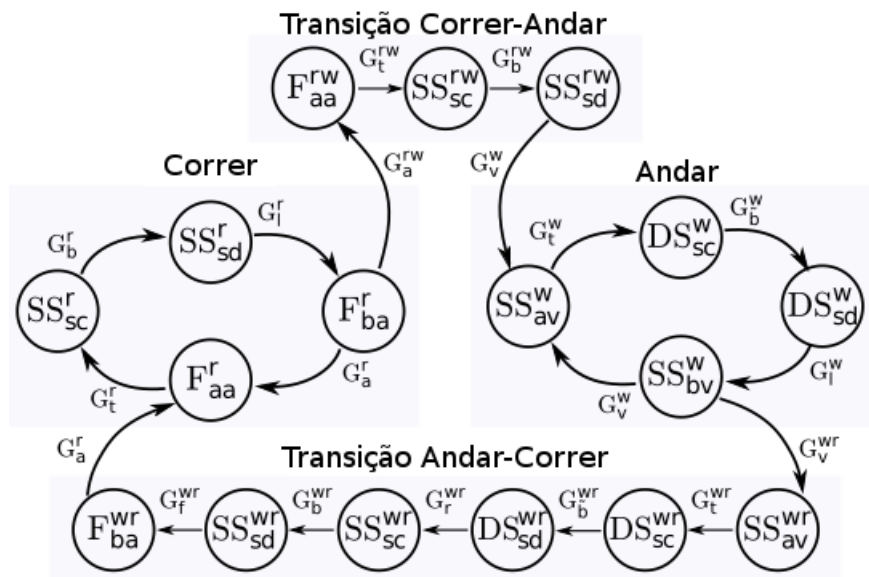
5.2 MODELAGEM UNIFICADA PARA OS MOVIMENTOS DE ANDAR E CORRER

Levando em conta a complexidade do movimento de corrida, um dos desafios para a execução do mesmo é a transição entre o início do movimento e o fim do mesmo, transitando para o movimento de caminhada quando necessário. Shahbazi et al. (2016) propõe um controlador

baseado em eventos para a transição desses movimentos, utilizando como base os modelos SLIP e IP. Propondo o uso de uma versão modificada do SLIP para a fase de suporte duplo do andar.

Para representar seu controlador, Shahbazi et al. (2016) o descreve como o autômato visto na imagem 21. Seguindo as legendas apresentadas na tabela 3, cada nó do modelo representa uma fase do movimento sendo executado, apresentando qual a marcha utilizada e a sub-fase seguindo o modelo $Fase_{Sub-Fase}^{Marcha}$. Já os eventos que desencadeiam o movimento entre os nós seguem a notação G_{Evento}^{Marcha} .

Figura 21 – Autômato híbrido representando os padrões dos movimentos de andar, correr e transições entre eles.



Fonte – Adaptado de (SHAHBAZI et al., 2016)

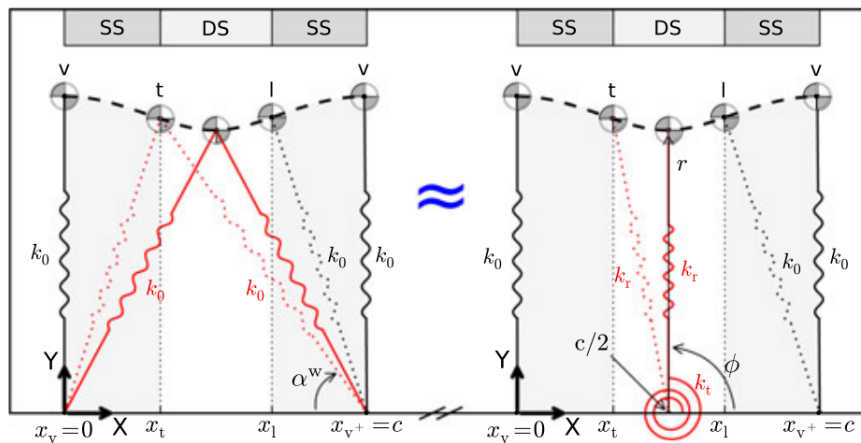
Tabela 3 – Notação utilizada para descrever o autômato de transição de movimentos

Marcha		Fase	
r	correr	F	voe
w	andar	SS	suporte único
rw	transição correr-andar	DS	suporte duplo
wr	transição andar-correr		Evento
	Sub-Fase	a	ápice
		b	inferior
sc	compressão da mola	v	<i>Vertical Leg Orientation</i> (VLO)
sd	descompressão da mola	\tilde{b}	inferior virtual
bv	antes da VLO	t	aterragem
av	após a VLO	l	decolagem
ba	antes do ápice	f	decolagem da perna frontal
aa	após o ápice	r	decolagem da perna traseira

Como pode ser visto pelo autômato, durante a fase de suporte duplo o movimento

apresenta as sub-fases de compressão e decompressão da mola, o que acontece devido ao uso de uma variação do modelo de mola-massa para essa etapa do movimento. As pernas dos agentes, tocando no chão, são abstraídas como uma única perna executando o modelo SLIP. Além dessa abstração, Shahbazi et al. (2016) propõe a incorporação de uma mola axial e de torção, como podem ser vistas na figura 22.

Figura 22 – Modelo SLIP na fase de suporte duplo e abstração do sistema como um SLIP único com mola axial e de torção.



Fonte – Retirado de (SHAHBAZI et al., 2016)

6 TRABALHOS RELACIONADOS

Nesse capítulo será focada a apresentação de trabalhos relacionados ao desenvolvimento de módulos de movimentação completos em robôs humanoides. Para esse capítulo, não serão apresentados trabalhos relacionados especificamente ao movimento de corrida, visto que os principais trabalhos relacionados à essa área já foram trabalhados no capítulo 5.

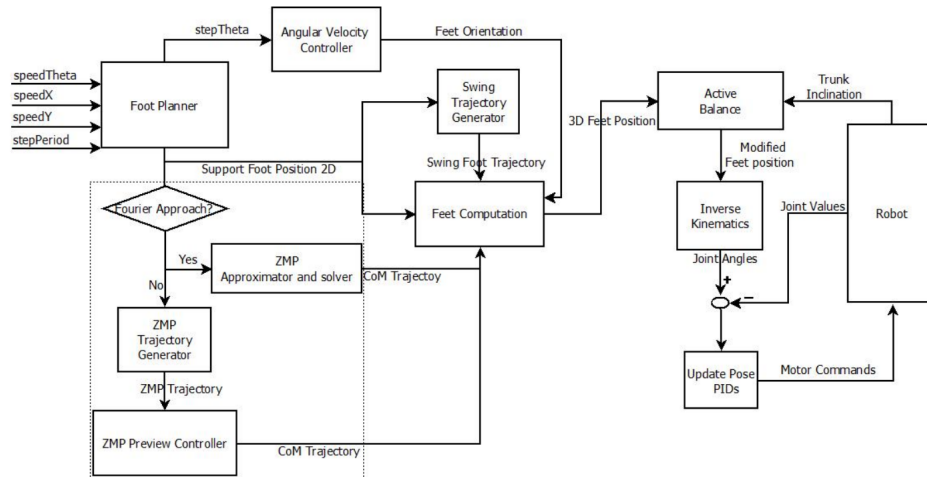
6.1 MÓDULOS DE EXECUÇÃO DE MOVIMENTOS

Para o desenvolvimento de módulos de movimentação complexos, como o proposto nesse trabalho, é necessário definir diversos sub-módulos e como eles interagem entre si para executar o movimento. Considerando esse quesito, um dos trabalhos mais similares ao aqui proposto é o módulo de andar omnidirecional proposto em (SHAFII, 2015).

Shafii (2015) propõe um módulo completo para a execução do movimento de andar omnidirecional, utilizando como base para o movimento o modelo *Cart-Table* que foi apresentado na seção 4.2.1. Como pode ser visto na figura 23, para executar o movimento são necessárias diversas etapas; partindo de um conjunto de velocidades, rotação e período do passo, a posição do pé é planejada. A partir dela, são utilizados ZMP, gerador de CoM, controlador de velocidade angular, e gerador de trajetória da perna de balanço para se chegar à posição final do pé com relação ao centro de massa. Essa posição é então alimentada para a cinemática inversa. O retorno do estado do robô é utilizado para atualizar a posição do seu tronco, para manter o equilíbrio, e para atualizar o controlador *Proportional-Integral-Derivative* (PID) do agente.

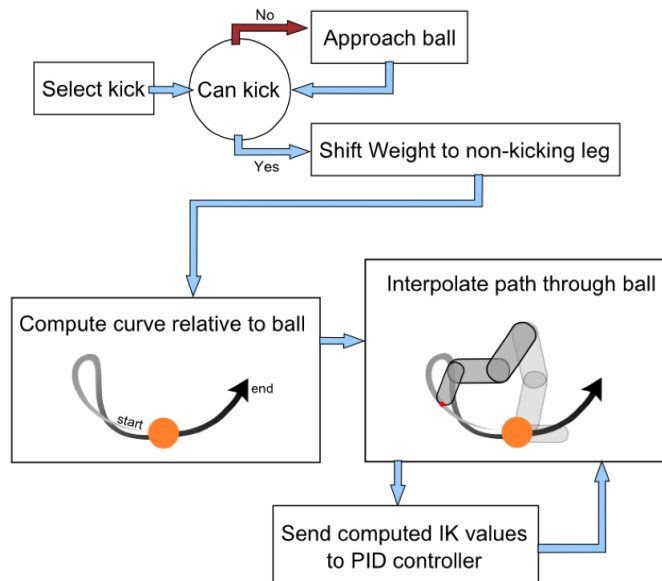
Outro trabalho que apresenta o desenvolvimento de um módulo completo para a execução de um movimento é o desenvolvimento do movimento de chute baseado na cinemática inversa desenvolvido por Lopez-Mobilia (2012). Para a execução do chute, o agente precisa inicialmente equilibrar-se em um pé apenas, para que o outro possa executar o chute. Em seguida ele deve calcular uma curva com relação à bola, determinando a trajetória que o pé irá percorrer para realizar o chute. Essa trajetória é então interpolada pela cinemática inversa, e enviada para o controlador PID. A figura 24 apresenta o fluxo de execução desse movimento.

Figura 23 – Módulo de andar omnidirecional, com todos seus sub-módulos e interações.



Fonte – Retirado de (SHAFII, 2015)

Figura 24 – Fluxo de execução de chute baseado na cinemática inversa.

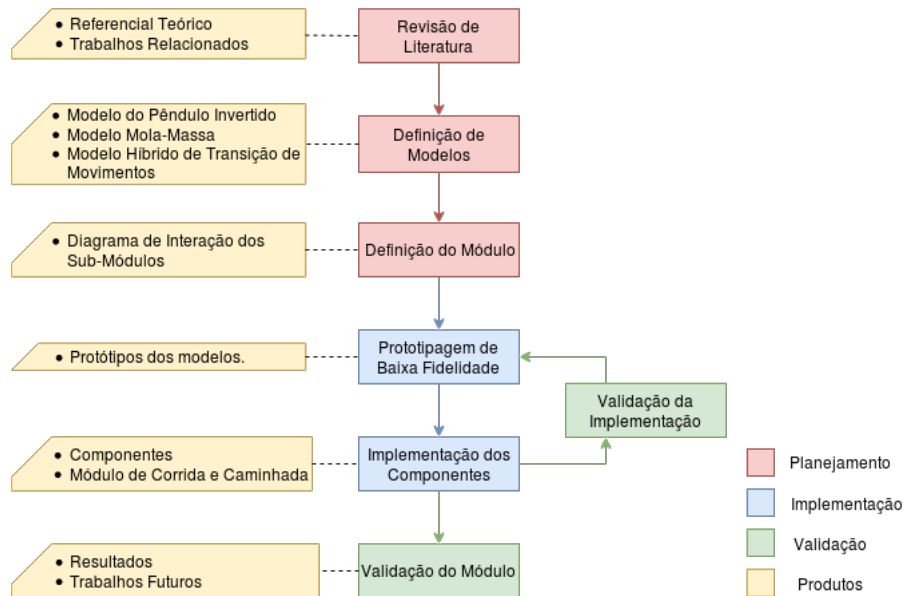


Fonte – Retirado de (LOPEZ-MOBILIA, 2012)

7 MÓDULO DE CORRIDA E CAMINHADA PARA ROBÔS HUMANOIDES

O desenvolvimento desse trabalho foi dividido em sete etapas apresentadas na figura 25.

Figura 25 – Etapas do desenvolvimento do projeto.



Fonte – O autor

Na primeira etapa, de revisão de literatura, foram coletados e filtrados trabalhos relacionados ao tema do desenvolvimento das habilidades de corrida e caminhada em um robô. A partir desses trabalhos foi desenvolvido o referencial teórico e trabalhos relacionados, apresentados nos capítulos 2, 3, 4, 5 e 6.

A segunda etapa foi responsável pela definição dos modelos que seriam utilizados para o desenvolvimento desse projeto, a partir dos quais os componentes seriam criados para o funcionamento do módulo de corrida e caminhada. A partir dessa etapa foram escolhidos o modelo do pêndulo invertido, o modelo mola-massa, e o modelo híbrido de transição de movimentos para serem utilizados pelo módulo. Esses modelos foram apresentados respectivamente nas seções 4.2.2, 5.1 e 5.2.

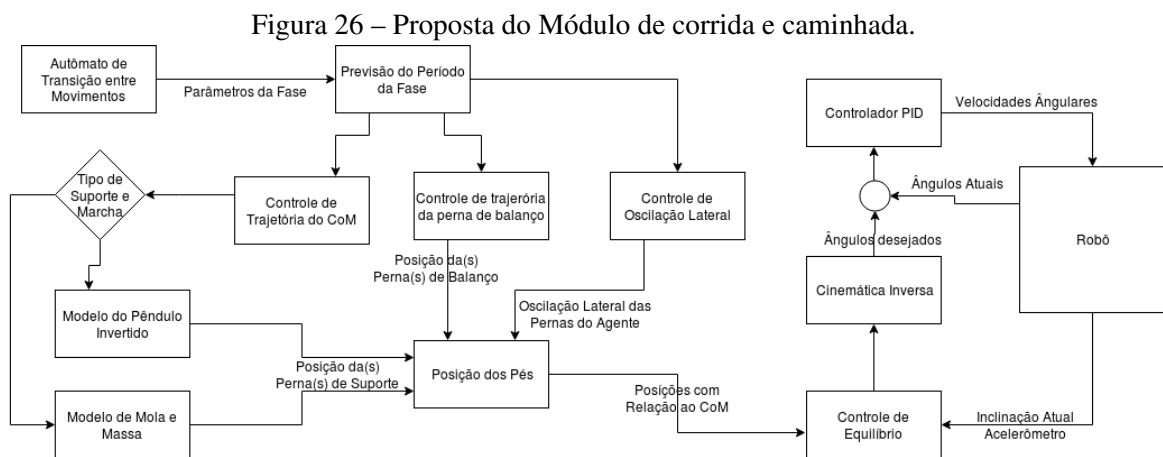
Em seguida, a etapa de definição do módulo foi responsável pela definição dos componentes a fazer parte do módulo de corrida e caminhada, e das relações entre esses componentes. A descrição detalhada do módulo e seus componentes pode ser vista nas seções seguintes.

A etapa seguinte caracterizou o início das etapas de implementação, composta das etapas de prototipagem, implementação de componente e validação da implementação. Para cada componente definido na etapa anterior, foi criado um protótipo de baixa fidelidade na plataforma *Scilab* (ENTERPRISES, 2012). A partir desse protótipo, o componente foi implementado no código do agente, e em seguida validado com relação ao protótipo. O uso dessa etapa inicial de validação permitiu a execução de testes preliminares para ajuste do módulo.

Por fim, na etapa de validação foi definida uma bateria de testes para a avaliação do módulo implementado, a partir dos quais foram colhidos resultados.

7.1 MÓDULO DE CORRIDA E CAMINHADA

Para a execução de movimentos tão complexos como os de andar e correr, é necessário que diversos componentes funcionem em harmonia, calculando diferentes porções ou etapas do movimento, a fim de ter um movimento estável na execução completa do módulo. Utilizando como base o módulo do movimento de andar já implementado no código do agente, e descrito no capítulo 6, foi criada a proposta para o módulo de corrida e caminhada, vista na figura 26.



Fonte – O autor

Nessa seção será descrito em detalhes o funcionamento dos componentes principais desse projeto. Os componentes de controle do equilíbrio, cinemática inversa e o controlador de PID não serão detalhados, uma vez que já faziam parte do código do agente, em seu módulo de andar omnidirecional, e foram apenas integrados nesse módulo. Para o entendimento dos componentes restantes, deve-se levar em consideração os parâmetros utilizados pelas etapas do movimento, descritos na tabela 4.

Tabela 4 – Notação dos principais parâmetros do módulo de correr e andar.

Notação	Descrição
r_0	Comprimento de repouso da perna
r_{LO}	Comprimento da perna na decolagem
φ_0	Ângulo de ataque com relação ao solo
φ_{apex}	Ângulo da perna no ápice da etapa de voo
k	Coefficiente de elasticidade da mola
\dot{r}_{TD}	Velocidade radial inicial
$\dot{\varphi}_{TD}$	Velocidade azimutal inicial
ΔX_{DS}	Tamanho do passo no eixo X
ΔY_{DS}	Separação entre as pernas no eixo Y

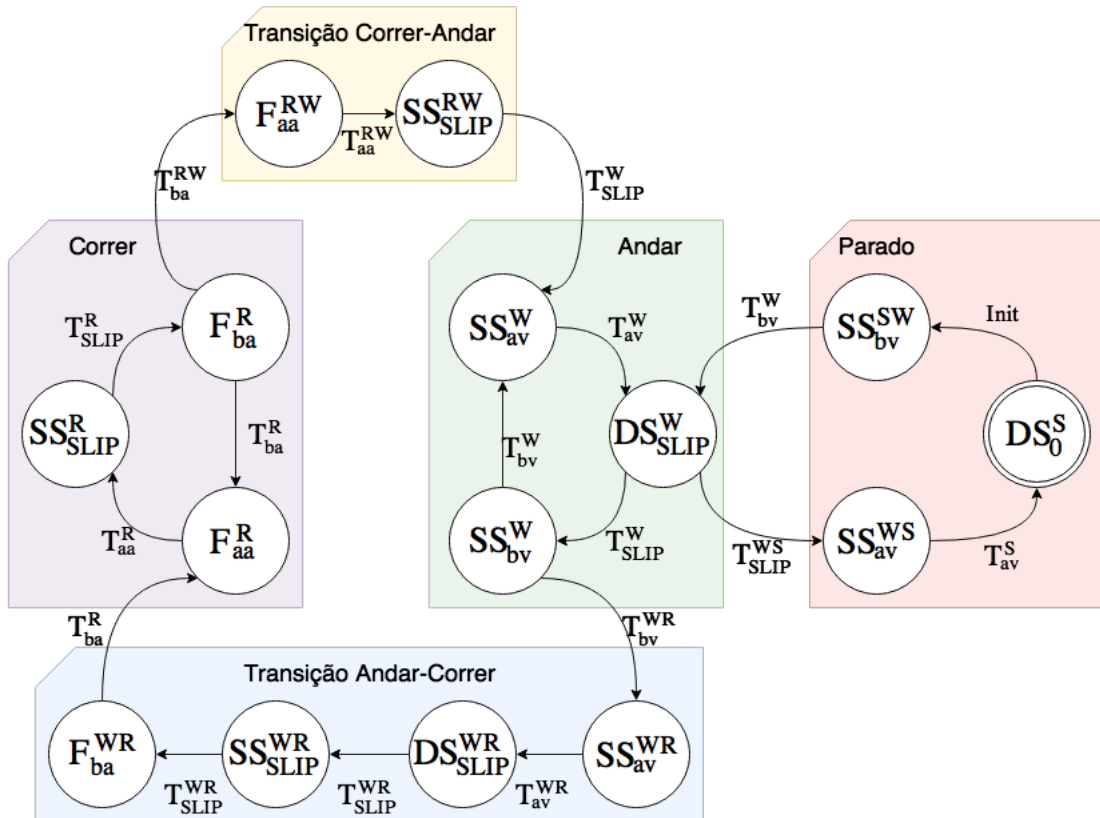
7.1.1 Autômato de transição entre movimentos

Nesse componente foi implementada uma variação do modelo visto na seção 5.2, proposto por Shahbazi et al. (2016). Para a simplificação do modelo, as fases de suporte único e suporte duplo cujas sub-fases fossem de compressão e descompressão foram mescladas, se tornando uma única sub-fase *SLIP*. Foram também propostas duas novas etapas de transição, realizando a transição entre o movimento de andar e o estado parado (*S*), e vice-versa, representadas como *WS* e *SW* respectivamente. O autômato adaptado para esse projeto pode ser visto na figura 27.

No início da execução do módulo, o agente estará no estado parado. A camada de inteligência do agente pode, a qualquer momento da execução do módulo, mandar uma requisição para a execução de um dos três movimentos, andar, correr ou parar. A partir dessa requisição, o autômato irá passar pelos nós necessários para alcançar esse movimento. Para cada marcha, esse módulo carrega um conjunto de parâmetros de movimento, para ser utilizado pelos outros componentes.

Diferente do modelo proposto por Shahbazi et al. (2016), o autômato proposto para esse projeto utiliza métodos de predição de transição em lugar da verificação de eventos. A partir dos modelos de execução utilizados, cada nó do autômato possui associado a ele um método de predição de duração de tempo nessa etapa. Desse modo, a transição de uma etapa para a outra ocorre imediatamente quando o agente passa do tempo previsto para esse nó. No diagrama, essa transição é representada pelo tempo previsto para cada sub-fase $T_{Sub-Fase}^{Marcha}$. O cálculo do tempo previsto para cada tipo de nó é descrito na seção seguinte.

Figura 27 – Automato adaptado para a transição entre movimentos.



Fonte – O autor

7.1.2 Previsão do Período da Fase

Utilizando como base o autômato de transição de movimentos, pode-se ver que são necessários cinco métodos de previsão de tempo diferentes. Dentre eles, os mais similares são os de previsão de tempo antes e depois da posição vertical da perna, VLO, representada na figura 28. Para o primeiro, é necessário utilizar apenas os parâmetros de *Ângulo de Ataque* e *Velocidade Angular Inicial*, visto que o tempo previsto é o tempo para rotacionar entre o ângulo de ataque e VLO, na velocidade angular também definida. O cálculo é semelhante para a previsão após o VLO, apenas alternando a variação angular, que passa a ser do VLO para o ângulo suplementar a φ_0 . Desse modo, as fórmulas para esses métodos são:

$$T_{bv} = \frac{\pi/2 - \varphi_0}{\dot{\varphi}_0} \quad (7.1)$$

$$T_{av} = \frac{(\pi - \varphi_0) - \pi/2}{\dot{\varphi}_0} \quad (7.2)$$

segue o modelo do pêndulo invertido. Para a implementação inicial do movimento, foi utilizada uma versão simplificada do movimento, desconsiderando a gravidade. Durante essa etapa do movimento, o agente mantém sua componente radial da perna estática, alterando apenas o ângulo azimutal segundo a fórmula:

$$\varphi(t) = \varphi_0 + \dot{\varphi} * t \quad (7.5)$$

Já durante as etapas de suporte único do movimento de corrida, e na de suporte duplo do movimento andar, o cálculo da trajetória do centro de massa segue o modelo SLIP apresentado no capítulo 5. Utilizando as equações 5.22 e 5.23 o agente pode calcular a posição do centro de massa no instante atual do tempo.

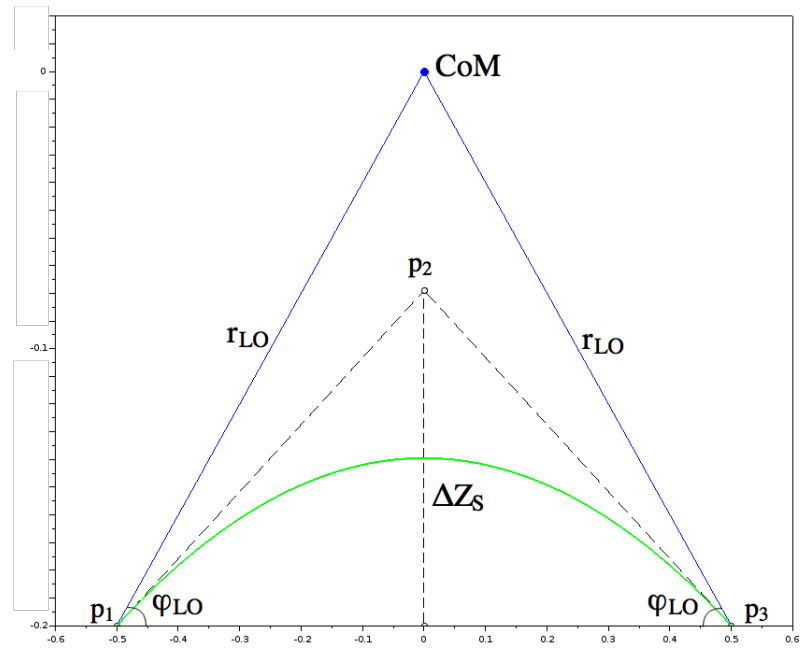
7.1.4 Controle de trajetória da perna de balanço

Além do movimento da perna de suporte que influencia diretamente a posição do CoM, para a execução dos movimentos de andar e correr é necessário realizar também o movimento da perna de balanço, responsável por transicionar o pé do agente da última posição de suporte para a seguinte, além de auxiliar no equilíbrio do agente (SEYFARTH et al., 2003). Durante o movimento de andar, essa transição acontece completamente durante a etapa de suporte único da perna oposta.

Utilizando o componente de controle de trajetória do CoM, em conjunto com a previsão de período do movimento é possível definir a posição prevista da próxima etapa de suporte. Com essa posição conhecida, a trajetória da perna de balanço para o movimento de andar é calculada utilizando uma simples curva de Bézier quadrática (SEDERBERG, 2012), com os pontos iniciais e finais sendo a posição do pé na última etapa de suporte e a posição seguinte, respectivamente. O terceiro ponto da curva de Bézier é calculado como o ponto médio entre os dois pontos no eixo X , com altura Z definida pelo parâmetro ΔZ_S , como altura do passo. A curva gerada para o movimento de andar pode ser vista na figura 29. Com φ_{LO} :

$$\varphi_{LO} = \pi - \arcsin \left(\frac{\sin(\varphi_0) \frac{\Delta X_{DS}}{2}}{r_{LO}} \right) - \varphi_0 \quad (7.6)$$

Figura 29 – Representação da curva de Bézier gerada pelos pontos p_1 , p_2 e p_3 . Em azul a representação da perna do agente e em verde, a curva que ela deve seguir.



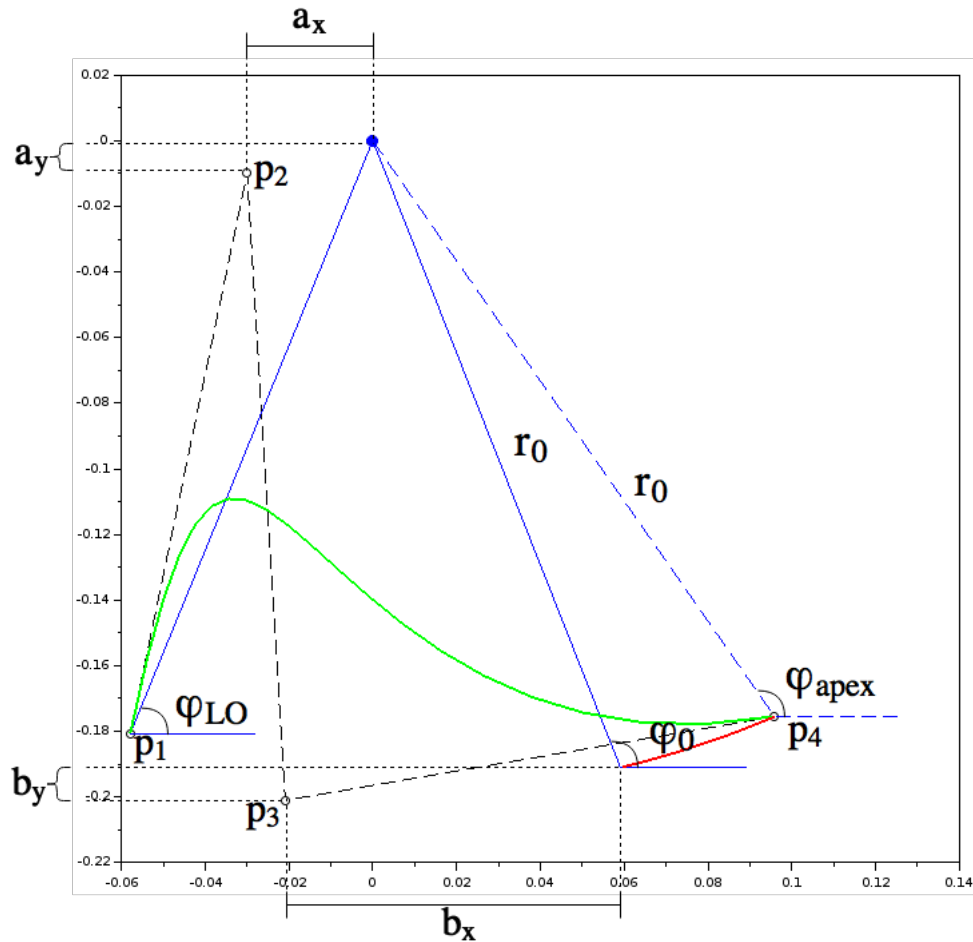
Fonte – O autor

O cálculo da trajetória de balanço durante o movimento de corrida entretanto, possui uma duração maior do que a do andar, iniciando na etapa de voo imediatamente após a etapa de suporte único da própria perna, e terminando após a segunda etapa de voo, que ocorre após o suporte único da perna seguinte. Utilizando o autômato da figura 27 como referência, pode-se dizer que enquanto a etapa de balanço de uma perna no andar realiza uma volta ($DS_{SLIP}^W \rightarrow SS_{bv}^W \rightarrow SS_{av}^W \rightarrow DS_{SLIP}^W$), ela realiza duas durante o correr ($SS_{SLIP}^R \rightarrow F_{ba}^R \rightarrow F_{aa}^R \rightarrow SS_{SLIP}^R \rightarrow F_{ba}^R \rightarrow F_{aa}^R \rightarrow SS_{SLIP}^R$).

Seguindo o modelo proposto por Seyfarth et al. (2003), o controle da trajetória até o ápice da segunda etapa de voo deve ter como ponto alvo posicionar a perna tão esticada quanto ela deve estar no início da próxima etapa de suporte, e com um ângulo ϕ_{apex} maior do que o ângulo de ataque. Durante a sub-fase seguinte, após o ápice, o modelo rotaciona a perna de forma a alcançar o ângulo de ataque.

Até alcançar o ápice, o movimento executado pela perna é novamente uma curva de Bézier. Entretanto, para o movimento de corrida foi proposta a utilização de uma curva de Bézier cúbica, com parâmetros dos dois pontos intermediários calculados com base na posição do CoM, a partir dos valores a_x e a_y , e no ponto final da etapa de balanço, com os valores b_x e b_y , a fim de obter uma curva semelhante ao movimento da perna na corrida do ser humano. O

Figura 30 – Representação da curva de Bézier gerada pelos pontos p_1 , p_2 , p_3 e p_4 . Em azul a representação da perna do agente e em verde, a curva de Bézier que ela deve seguir. Após alcançar o ponto p_4 , no ápice da etapa de voo, a perna segue a curva em vermelho, para se preparar para o pouso.



Fonte – O autor

comportamento da perna de corrida com relação com o CoM durante o movimento de corrida pode ser visto na figura 30.

7.1.5 Controle de oscilação lateral

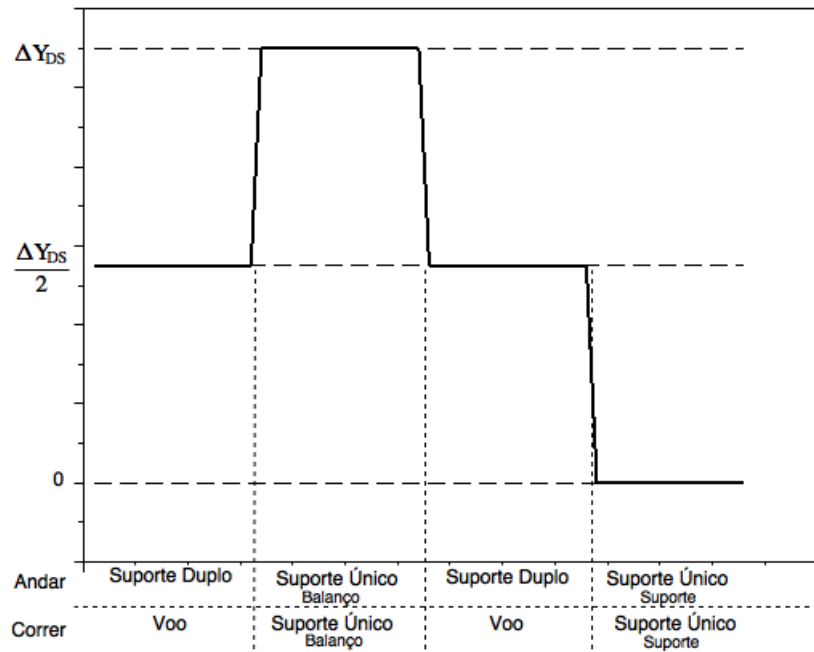
Para manter o agente equilibrado durante todas as etapas do movimento, é necessário realizar a transição da posição do centro de massa de forma que sua projeção no chão coincida com a posição do ZMP. A transição da posição do CoM no eixo frontal X do movimento é realizada pelo componente *controle de trajetória do CoM*. Já a movimentação do mesmo no eixo lateral Y é realizada pelo componente de *controle de oscilação lateral*.

Para definir o cálculo dessa transição é primeiro necessário definir qual o ZMP ideal durante cada fase do movimento. Durante as fases de contato, o ZMP é localizado no ponto

médio dos pontos de contato com o chão, sendo assim a posição do pé de suporte durante a fase de suporte único, e a posição do ponto médio entre os dois pés, durante o suporte duplo.

Uma vez definida a posição ideal do ZMP, e por consequência, do GCoM, é necessário definir a relação desses com a posição calculada para os pés do agente. Considerando o ZMP como a posição do pé de suporte durante a etapa de suporte único, sua posição ideal com relação ao CoM no eixo Y é 0. Já durante a etapa de suporte duplo, a distância no eixo Y de cada pé para o CoM é definida como $\Delta Y_{DS}/2$. Para manter a distância lateral entre os pés semelhante durante todas as etapas do movimento, é proposto que durante etapas de suporte único, a distância do pé para o CoM seja igual a ΔY_{DS} . Por fim, para realizar a transição de um pé para o outro, é proposto que durante a etapa de voo do movimento de correr, a posição dos pés se comporte de maneira similar à fase de suporte duplo. Essas posições podem ser vistas na figura 31.

Figura 31 – Posição ideal do pé com relação ao CoM durante o movimento.



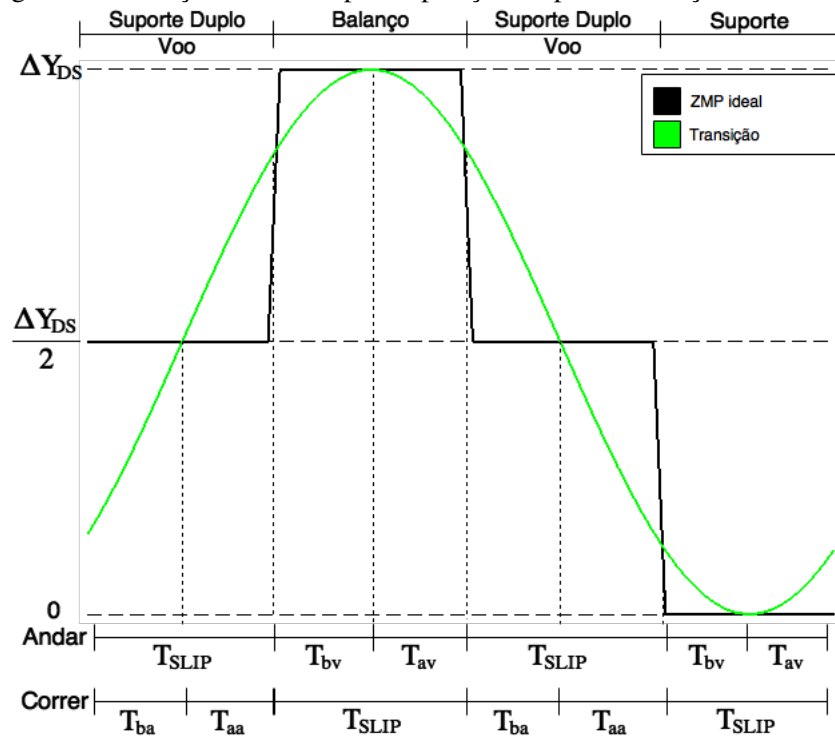
Fonte – O autor

Para realizar essa transição de maneira suave, foi proposto o uso de uma função seno, parametrizada para se aproximar ao ideal e para permitir futuras otimizações. Para calcular a oscilação durante uma sub-fase específica do movimento, a equação da oscilação foi escrita como:

$$Y_{osc} = \Delta Y_{DS} \left(\sin \left(p \frac{T_c}{T_{SF}} + S_{SF} \frac{\pi}{4} \right) + 1 \right) \quad (7.7)$$

Com T_c sendo o tempo atual da sub-fase, p representando a duração da sub-fase na função seno, sendo esta $\pi/4$ para todas as sub-fases com exceção da *SLIP*, que tem uma duração de $\pi/2$, e S_{SF} um contador representando a sub-fase que a função deve executar. O resultado dessa transição pode ser visto na figura 32.

Figura 32 – Posição calculada para a posição do pé com relação ao CoM.



7.1.6 Posição dos pés

Após o cálculo da trajetória do centro de massa com relação ao ponto de suporte do agente, é necessário converter esses resultados em valores cartesianos, para em seguida se calcular a posição do(s) pé(s) com relação ao centro de massa. Essa conversão se dá pela fórmula:

$$x = r \cos(\varphi) \quad (7.8)$$

$$z = r \sin(\varphi) \quad (7.9)$$

Uma vez convertido em coordenadas cartesianas, pode-se encontrar a posição do pé

com relação ao centro de massa nos eixos X e Z . Para as etapas de suporte único, esse cálculo é simples, visto que o início da cadeia cinemática coincide com o ponto do centro de massa, sendo então o ponto $(0,0)$. Assim a posição do pé de suporte durante a etapa de suporte único é obtida multiplicando o resultado da conversão por -1 , invertendo a direção do vetor. Após a conversão para o eixo cartesiano, é aplicada a oscilação lateral calculada pelo componente de controle de oscilação lateral no eixo Y .

Já a posição dos pés de suporte durante a etapa de suporte duplo é calculada a partir do centro de massa com relação ao ponto médio entre os dois pés. Portanto, para obter a posição dos pés com relação ao CoM é necessário primeiro calcular a posição dos pés com relação ao ponto médio. A equação para encontrar esses vetores pode ser vista abaixo, com FFF sendo um multiplicador unitário de valor positivo caso o pé esquerdo esteja à frente, e negativo caso contrário.

$$L_{RelMid} = \begin{bmatrix} FFF \Delta X_{DS}/2 \\ -\Delta Y_{DS}/2 \\ 0 \end{bmatrix} \quad (7.10)$$

$$R_{RelMid} = \begin{bmatrix} FFF \Delta X_{DS}/2 \\ \Delta Y_{DS}/2 \\ 0 \end{bmatrix} \quad (7.11)$$

Em seguida, como temos a posição dos pés e do centro de massa no mesmo plano cartesiano, pode-se obter a posição dos pés com relação ao CoM realizando uma simples subtração de vetores:

$$L_{RelCoM} = L_{RelMid} - CoM_{RelMid} \quad (7.12)$$

$$R_{RelCoM} = R_{RelMid} - CoM_{RelMid} \quad (7.13)$$

Como o componente de controle de trajetória da perna de balanço já realiza seu cálculo com coordenadas cartesianas, calculando a posição do pé com relação ao CoM, o cálculo final de sua posição nesse componente se resume a aplicar a posição Y calculada pelo componente de controle de oscilação lateral. E com isso, é obtida a posição dos dois pés do agente, com relação ao centro de massa, prontas para serem enviadas para o controle de equilíbrio, e por fim, à cinemática inversa.

8 TESTES E RESULTADOS

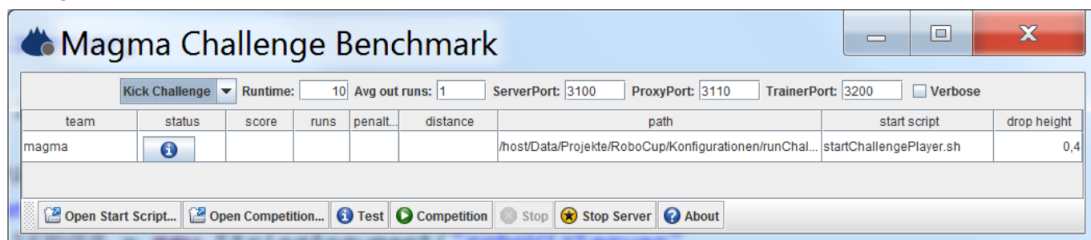
Nesse capítulo serão apresentados os métodos de testes para validação do trabalho proposto. Serão descritos os procedimentos e ferramentas utilizados para a realização dos testes, bem como a justificativa para eles. Em seguida serão apresentados os resultados dos testes e sua avaliação com relação aos objetivos desse projeto.

8.1 PROCEDIMENTO EXPERIMENTAL

Esse trabalho propôs a implementação de um módulo integrado de corrida e caminhada para um agente simulado, com seu foco principal sendo o desenvolvimento da habilidade de corrida para o agente. Para validar a execução desse movimento, será comparado o movimento executado pelo agente com o movimento planejado pelo modelo matemático.

Para realizar os testes de maneira automática e coletar dados do mesmo, foi utilizada uma versão modificada da ferramenta de testes *Magma Challenge Benchmark* (DORER et al., 2016). Essa ferramenta se conecta ao servidor utilizando a mesma porta e troca de mensagens que o monitor da partida, dessa forma ela consegue controlar a partida e coletar dados exclusivos do monitor. A interface da ferramenta pode ser vista na figura 33.

Figura 33 – Ferramenta *Magma Challenge Benchmark* com desafio de chute selecionado.



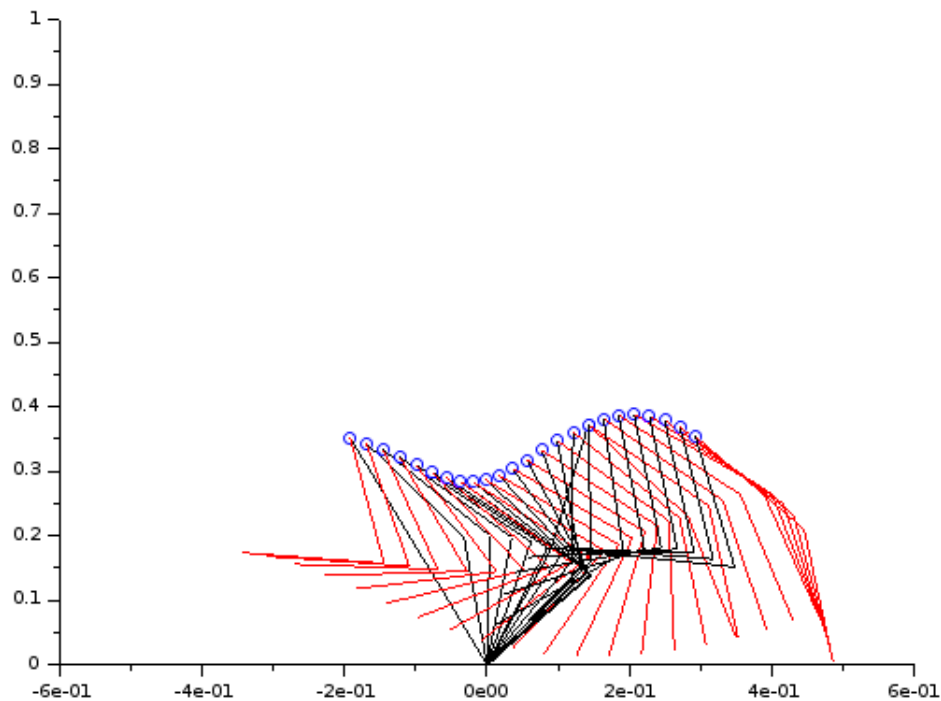
Fonte – Retirado de (DORER et al., 2016)

Utilizando essa ferramenta foi definido um cenário de teste para coletar as informações da movimentação do agente. Uma vez inicializado, o agente deve realizar imediatamente as transições entre os estados *parado*, *andando* e *correndo*. A partir desse ponto, ele deve correr o máximo que conseguir até cair. Nesses testes, só serão considerados aqueles em que o agente caia após iniciar o movimento de corrida, ou seja, testes onde ele consiga se locomover pelo menos 1m. Para coletar a média dos resultados, foram realizados 100 testes com esse cenário.

Durante a etapa de desenvolvimento do projeto, descrita no capítulo anterior, foram

desenvolvidos protótipos para diferentes componentes do módulo. Dentre esses, foi desenvolvido utilizando a ferramenta Scilab um protótipo de baixa fidelidade do movimento de corrida, desprezando inércia, gravidade e o peso de partes diferentes do robô, apenas simulando o modelo matemático da corrida. Esse protótipo foi utilizado como base para definição de valores de comparação e validação do módulo final. A figura 34 apresenta a execução de um passo pelo protótipo.

Figura 34 – Protótipo do modelo de corrida. Em vermelho e preto, as pernas esquerda e direita do agente, respectivamente e em azul, o centro de massa do mesmo.



Fonte – O autor

Para realizar os movimentos de andar e correr, é necessário definir os parâmetros de execução descritos no capítulo 7. Esses parâmetros são responsáveis por descrever como o movimento será executado, influenciando diretamente na velocidade do mesmo. Para realizar os testes do movimento esses parâmetros foram definidos de maneira empírica, utilizando o protótipo como base para testar os valores. Os valores definidos para a execução do andar e correr estão descritos na tabela 5.

Como descrito no capítulo anterior, o modelo matemático utilizado calcula as posições dos pés do agente com relação ao seu centro de massa, e o movimento é executado a partir

Tabela 5 – Parâmetros utilizados nos testes de validação do movimento.

Notação	Descrição	Andar	Correr
r_0	Comprimento de repouso da perna (m)	0.00	0.22
r_{LO}	Comprimento da perna na decolagem (m)	0.22	0.00
φ_0	Ângulo de ataque com relação ao solo (rad)	1.87	1.87
φ_{apex}	Ângulo da perna no ápice da etapa de voo (rad)	0.00	2.07
k	Coefficiente de elasticidade da mola	2500	2500
\dot{r}_{TD}	Velocidade radial inicial (m/s)	0.10	0.50
$\dot{\varphi}_{TD}$	Velocidade azimutal inicial (rad/s)	-2.00	-3.00
ΔX_{DS}	Tamanho do passo no eixo X (m)	0.10	0.00
ΔY_{DS}	Separação entre as pernas no eixo Y (m)	0.11	0.11

desses cálculos. O agente deve então alcançar essas posições utilizando os componentes de PID e Cinemática Inversa já existentes no agente. Como descrito no capítulo 2, o agente recebe do servidor as posições angulares de todas as suas juntas, e a partir destas, é possível utilizar o cálculo de cinemática direta descrito no capítulo 3 para obter a posição real dos pés com relação ao CoM. Com esses dados, é possível visualizar a diferença entre o movimento planejado pelo módulo de corrida e o movimento executado.

Ao descrever o movimento de corrida no capítulo 5 foi destacado que ele é caracterizado pela presença de duas etapas principais, a de postura e de voo. Essas etapas, caracterizadas pelo suporte do corpo do robô com o chão, são essenciais para classificar o movimento executado como um de corrida. Para validar a presença dessas duas etapas, bem como a duração de cada uma delas por passo executado, foi escolhida a análise do sensor FRP. A partir da força de reação aplicada no pé do agente, ou da ausência dela, é possível observar os momentos em que o robô executa as diferentes etapas do movimento. A duração de cada etapa do movimento será também comparada com os resultados previstos pelo protótipo.

8.2 RESULTADOS

Como descrito na seção anterior, parte dos resultados apresentados serão comparados com os resultados apresentados pelo protótipo de baixa fidelidade. Esse protótipo foi executado desconsiderando forças externas como o atrito, executando o movimento de corrida exatamente como planejado pelo modelo de massa e mola. É importante notar também que esse protótipo simula o movimento do agente em apenas duas dimensões, desprezando o movimento lateral das pernas e colisões entre as mesmas. Os resultados previstos pelo protótipo podem ser vistos na tabela abaixo:

Tabela 6 – Valores previstos pelo protótipo para o movimento.

Tempo de Suporte	Tempo de voo	Velocidade média
0.21s	0.09s	0.53m/s

Uma vez executados os testes do movimento de corrida utilizando a ferramenta de testes, os primeiros resultados obtidos foram referentes à velocidade do movimento executado e a distância percorrida. Para obter os dados apenas da corrida, desconsiderando a distância percorrida durante a transição entre o movimento de andar, a coleta de dados só foi iniciada uma vez que o agente se distanciasse 0.5m de sua posição inicial. Uma vez que o agente caísse, seriam salvos a distância percorrida, e o tempo, obtendo assim o resultado da velocidade média.

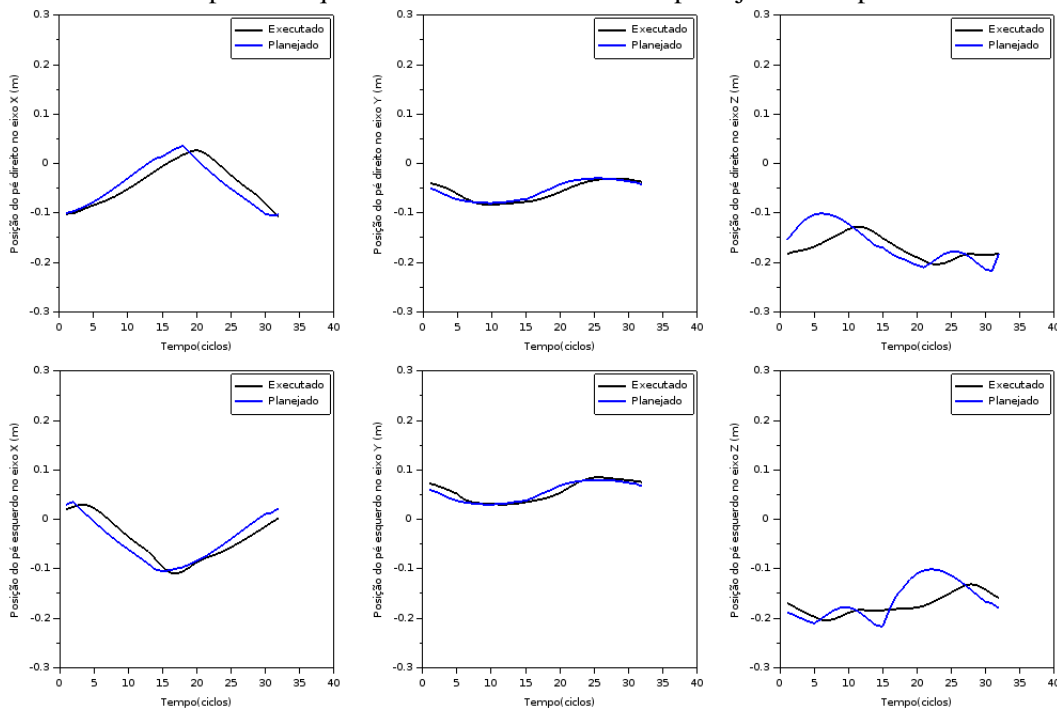
Após os 100 testes realizados, o agente apresentou uma velocidade média de 0.55m/s, percorrendo em média 3.39m antes de cair. Dos 100 testes realizados, 67 conseguiram ultrapassar a distância mínima de 1m em sua primeira tentativa. O restante apresentou uma média de 2.27 quedas até ultrapassar esse limite, resultando em uma média de 0,75 tentativas inválidas por teste.

Durante a execução dos testes, foi coletada através da cinemática direta, a posição dos pés do agente com relação ao CoM para cada ciclo de execução. Foi também coletada a posição desejada que foi inicialmente enviada para a cinemática inversa. Foram então identificados o início e fim de dois passos completos, e gerados os gráficos da figura 35 comparando os dados.

Após a análise dos gráficos, foi notada a clara discrepância entre as posições planejadas e executadas no eixo Z. Foi realizada uma análise onde concluiu-se que a etapa de postura era a maior responsável pela discrepância. Essa é etapa onde o modelo de massa e mola comprime a última, e depois a expande, para executar o movimento da mola. Como a perna do robô não se trata de uma mola, mas de juntas articuladas simulando o comportamento de uma mola, foi levantada a suspeita de que as velocidades máximas das juntas do agente não eram capazes de executar o movimento calculado para o agente.

Uma vez levantada essa suspeita foi realizado um novo teste do movimento do agente. Nesse teste, foi coletada a velocidade enviada para o servidor de cada uma das juntas do agente, antes do servidor reduzi-las à velocidade máxima. Em seguida, assim como no gráfico anterior, foram identificados dois passos completos do movimento, e gerada a figura 36, representando as velocidades enviadas durante esses passos.

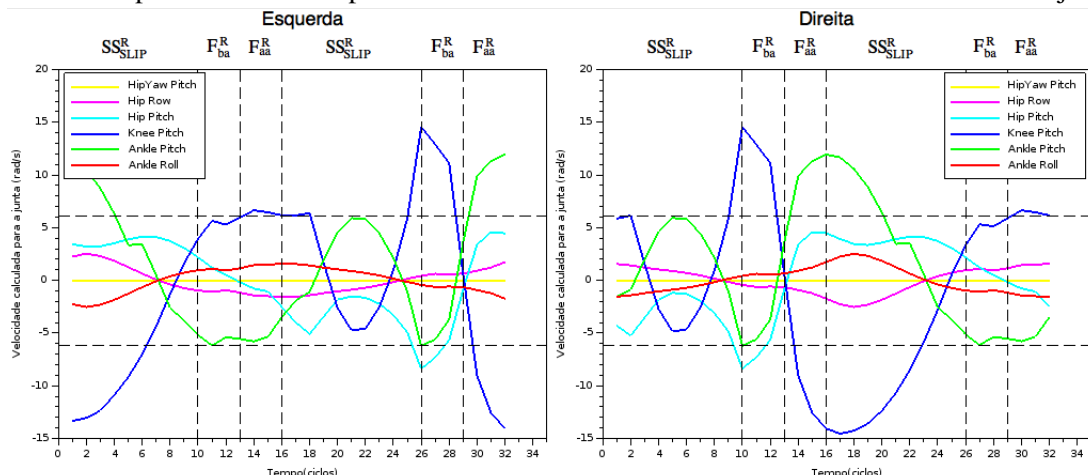
Figura 35 – Comparação da posição dos pés executadas versus desejadas nos eixos X Y e Z respectivamente. Os gráficos de cima representando as posições para o pé direito, e os de baixo para o esquerdo. Em azul o movimento planejado e em preto o executado.



Fonte – O autor

A partir desses gráficos pode-se notar que as velocidades máximas escolhidas para as juntas *Pitch* do quadril, joelho e tornozelo do agente ultrapassam a velocidade máxima determinada para o modelo do robô. Por serem as juntas do tipo *Pitch*, tendo seu eixo de rotação ao redor do eixo *Y*, pode-se concluir que os limites de velocidade das juntas do agente são

Figura 36 – Velocidades enviadas para o servidor para as juntas das pernas. À esquerda, as juntas da perna esquerda, e à direita, as da perna direita. Em amarelo, magenta e ciano as juntas *YawPitch*, *Row* e *Pitch* do quadril, respectivamente. Em azul a junta *Pitch* do joelho. E em verde e vermelho as juntas *Pitch* e *Roll* do tornozelo respectivamente. As linhas tracejadas verticais representando as etapas do movimento e as horizontais a velocidade máxima das juntas.

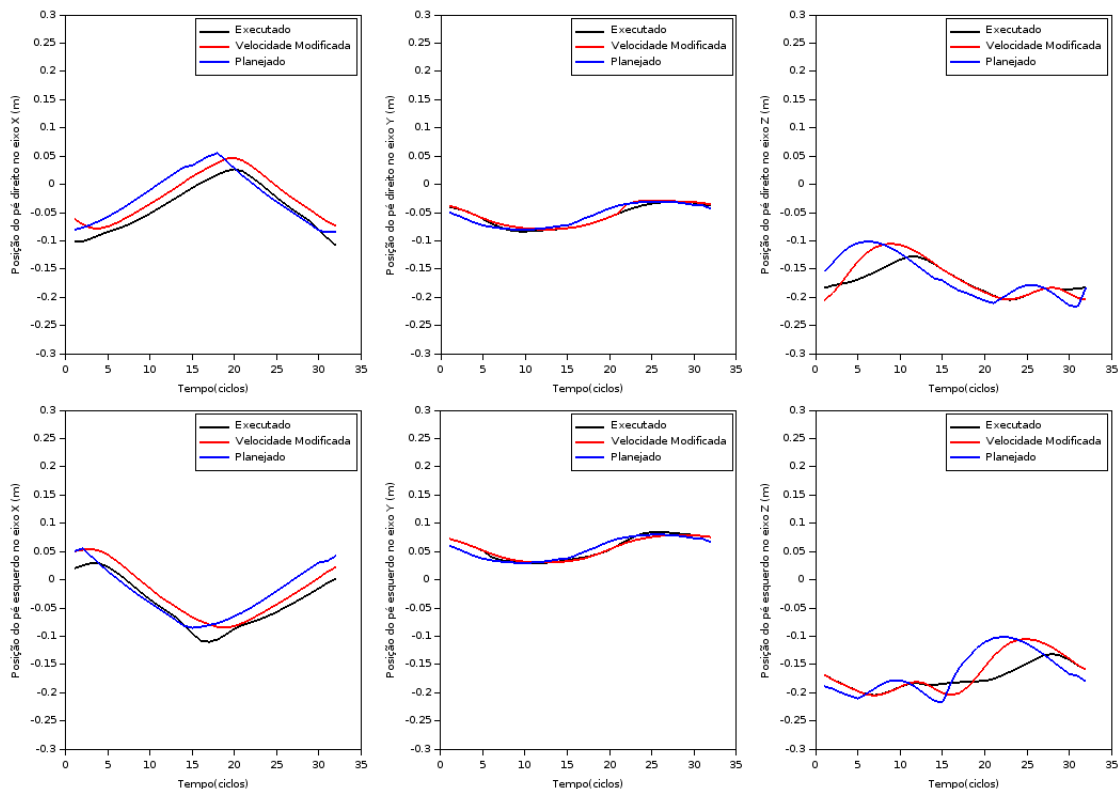


Fonte – O autor

responsáveis pela discrepância nas posições alcançadas no eixo Z.

A fim de verificar se poderia existir alguma outra razão por trás da discrepância além das velocidade máximas, foi alterado o código do servidor para permitir velocidades superiores às enviadas durante o teste anterior. Após ser realizado novamente o teste, e coletado os dados necessários, a figura 37 foi gerada, onde pode-se notar uma clara redução nas discrepâncias. As discrepâncias restantes podem ser atribuídas ao funcionamento do servidor, cujas mensagens com o resultado do movimento só são enviadas no ciclo seguinte, explicando o aparente desvio para a direita das curvas.

Figura 37 – Comparação da posição dos pés executadas versus desejadas nos eixos X Y e Z respectivamente. Os gráficos de cima representando as posições para o pé direito, e os de baixo para o esquerdo. Em azul o movimento planejado e em preto o executado com limites de velocidades nas juntas e em vermelho, o movimento executado sem os limites de velocidade.



Fonte – O autor

A última informação que foi planejada para a coleta de dados foi a análise do sensor FRP com intuito de validar a presença das etapas de voo e postura no movimento. Os valores recebidos pelo sensor no eixo Z foram coletados durante a execução dos testes de corrida, gerando a figura 38. A partir do gráfico, pode-se observar ambas as etapas do movimento sendo representadas, com a etapa de voo durando aproximadamente 4 ciclos e a de postura 12.

Resumindo os coletados e os comparando com os valores previstos inicialmente pelo protótipo obtemos a tabela 7 abaixo.

Figura 38 – Força aplicada na sola dos pés do agente durante o movimento. Em azul o pé direito e em preto o esquerdo. As linhas tracejadas verticais representando a separação entre etapas de postura e voo.

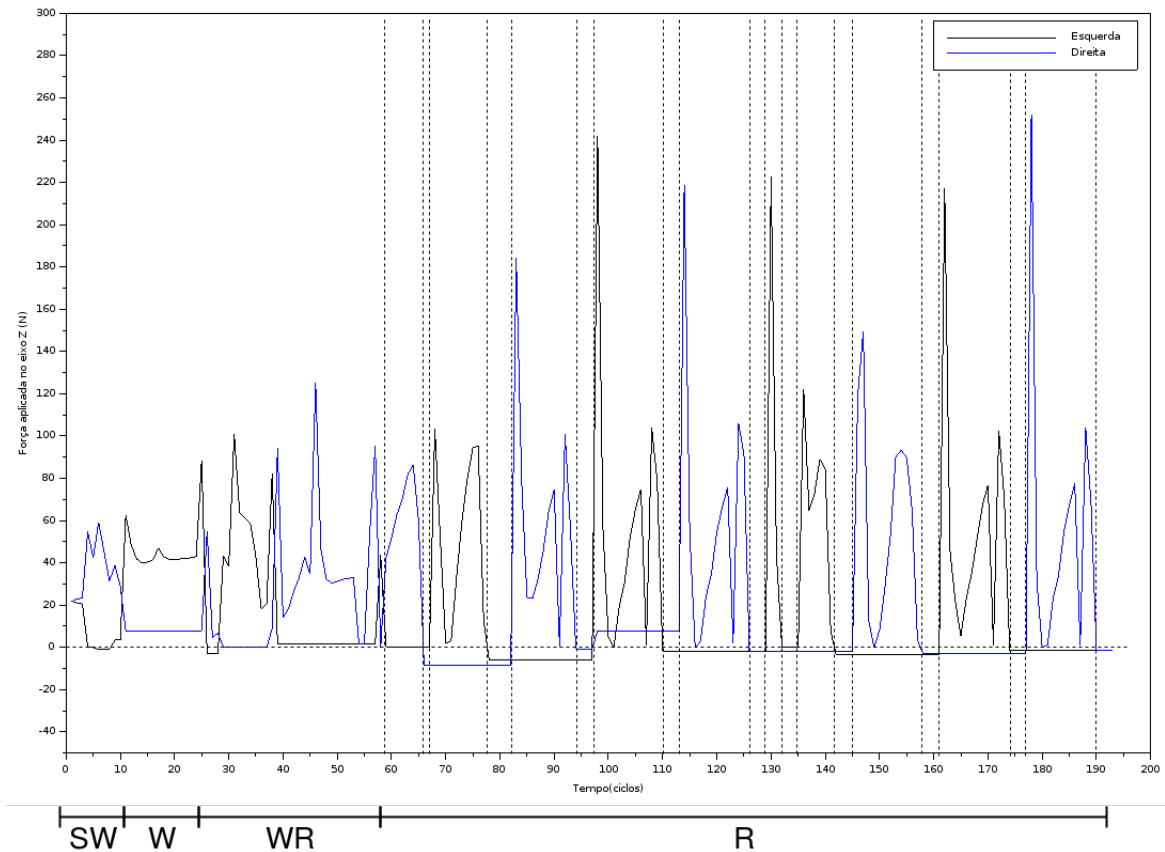


Tabela 7 – Comparação dos resultados previstos pelo protótipo com os coletados após a execução.

	Planejado	Executado
Tempo de Suporte	0.21s	12 ciclos (0.24s)
Tempo de voo	0.09s	4 ciclos (0.08s)
Velocidade média	0.53m/s	0.55m/s
Distância média		3.39m

9 CONSIDERAÇÕES FINAIS

Esse trabalho apresentou o desenvolvimento de um módulo de locomoção para um agente humanoide em ambiente simulado. O módulo de locomoção em questão engloba movimentos de caminhada e corrida, bem como as transições necessárias para executá-los. O principal objetivo desse trabalho foi o desenvolvimento inicial do movimento de corrida para o agente, e para tal, foi estudado e utilizado o modelo de Mola e Massa.

O movimento de corrida foi classificado nesse trabalho como um movimento composto pela etapa de postura e a de voo. Partindo desses critérios foram realizados testes com o módulo desenvolvido, que apresentou a presença de ambas as etapas, permitindo classificar o movimento realizado como um de corrida e cumprindo a proposta inicial do trabalho.

O módulo apresentado também propôs a execução do movimento de caminhada, com intuito de poder realizar as transições necessárias para sair do estado inicial parado do agente e executar a corrida. Tendo esse como objetivo principal, o movimento de caminhada executado alcançou seu objetivo, podendo ser aprimorado futuramente para ser utilizado em outras ocasiões que não apenas como transição para o movimento de corrida.

Para desenvolver o módulo apresentado, este foi dividido em diversos componentes responsáveis por diferentes camadas dos movimentos. Cada um desses componentes pode ser aprimorado individualmente para melhorar o desempenho do agente ao executar o movimento de corrida ou de andar. Também podem ser criados componentes extras para adicionar mais funcionalidade ao movimento.

9.1 TRABALHOS FUTUROS

O desenvolvimento desse projeto foi o início de um novo módulo de locomoção. A partir desse trabalho diversos outros poderão ser realizados para a melhoria das capacidades de movimentação do agente. Alguns dos trabalhos que podem ser realizados a partir desse são:

- Aprimorar o cálculo do movimento de andar do módulo, utilizando fórmulas que levem em consideração gravidade e torque.
- Desenvolvimento de um planejador de movimento capaz de escolher os melhores parâmetros dependendo da velocidade desejada para o movimento.

- Otimização dos parâmetros utilizados para obter movimentos mais eficientes.
- Desenvolvimento de um componente para controlar a rotação dos pés, permitindo ao movimento a realização de curvas.
- Realização de testes de estabilidade, verificando a capacidade do agente de se manter de pé enquanto realizando esses movimentos, mesmo após colisões com outros agentes.
- Desenvolvimento de um componente mais robusto de equilíbrio, para manter o agente equilibrado por mais tempo durante o movimento.

REFERÊNCIAS

- AKIYAMA, H.; DORER, K.; LAU, N. On the progress of soccer simulation leagues. In: SPRINGER. **Robot Soccer World Cup**. [S.l.], 2014. p. 599–610.
- ARSLAN, O.; SARANLI, U.; MORGUL, O. An approximate stance map of the spring mass hopper with gravity correction for nonsymmetric locomotions. In: IEEE. **Robotics and Automation, 2009. ICRA'09. IEEE International Conference on**. [S.l.], 2009. p. 2388–2393.
- ASADA, M.; KITANO, H. The robocup challenge. **Robotics and Autonomous Systems**, Elsevier, v. 29, n. 1, p. 3–12, 1999.
- BLICKHAN, R. The spring-mass model for running and hopping. **Journal of Biomechanics**, v. 22, n. 11, p. 1217 – 1227, 1989. ISSN 0021-9290. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0021929089902248>>.
- BOEDECKER, J.; ASADA, M. Simspark–concepts and application in the robocup 3d soccer simulation league. **Autonomous Robots**, v. 174, p. 181, 2008.
- BURKHARD, H.-D.; DOMANSKA, M. **SimSpark/SoccerServer RCSS as used for RoboNewbie**. 2012. Disponível em: <<https://www2.informatik.hu-berlin.de/~naoth/RoboNewbie/SimSpark-long.pdf>>.
- CRAIG, J. J. **Introduction to robotics: mechanics and control**. [S.l.]: Pearson Prentice Hall Upper Saddle River, 2005. v. 3.
- DENAVID, J.; HARTENBERG, R. S. A kinematic notation for lower-pair mechanisms based on matrices. **ASME J. Appl. Mech.**, 1955.
- DORER, K.; FISCHER, J.; GLASER, S.; NGUYEN, D.; OBRECHT, M.; WEILER, D. The magmaoffenburg 2016 robocup 3d simulation team. In: **Proceedings of RoboCup 2016**. [S.l.: s.n.], 2016.
- ENTERPRISES, S. Scilab: Free and open source software for numerical computation, version 5.41. **Scilab Enterprises, Versailles, France**, <http://www.scilab.org>, 2012.
- FERREIRA, R.; REIS, L. P.; MOREIRA, A. P.; LAU, N. Development of an omnidirectional kick for a nao humanoid robot. In: _____. **Advances in Artificial Intelligence – IBERAMIA 2012: 13th Ibero-American Conference on AI, Cartagena de Indias, Colombia, November 13-16, 2012. Proceedings**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 571–580. ISBN 978-3-642-34654-5. Disponível em: <http://dx.doi.org/10.1007/978-3-642-34654-5_58>.
- FIHL, P.; MOESLUND, T. B. Recognizing human gait types. In: **Robot Vision**. [S.l.]: INTECH, 2010.
- GEYER, H.; SEYFARTH, A.; BLICKHAN, R. Spring-mass running: simple approximate solution and application to gait stability. **Journal of theoretical biology**, Elsevier, v. 232, n. 3, p. 315–328, 2005.
- GLASER, S.; DORER, K. Trunk controlled motion framework. In: **2013 IEEE-RAS International Conference on Humanoid Robots**. [S.l.: s.n.], 2013.

HARADA, K.; KAJITA, S.; KANEKO, K.; HIRUKAWA, H. An analytical method for real-time gait planning for humanoid robots. **International Journal of Humanoid Robotics**, World Scientific, v. 3, n. 01, p. 1–19, 2006.

HARTENBERG, R. S.; DENAVIT, J. **Kinematic synthesis of linkages**. [S.l.]: McGraw-Hill, 1964.

KAGAMI, S.; KITAGAWA, T.; NISHIWAKI, K.; SUGIHARA, T.; INABA, M.; INOUE, H. A fast dynamically equilibrated walking trajectory generation method of humanoid robot. **Autonomous Robots**, Springer, v. 12, n. 1, p. 71–82, 2002.

KAJITA, S.; KANEHIRO, F.; KANEKO, K.; YOKOI, K.; HIRUKAWA, H. The 3d linear inverted pendulum mode: a simple modeling for a biped walking pattern generation. In: **Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)**. [S.l.: s.n.], 2001. v. 1, p. 239–246 vol.1.

KAJITA, S.; KANEHIRO, F.; KANEKO, K.; FUJIWARA, K.; HARADA, K.; YOKOI, K.; HIRUKAWA, H. Biped walking pattern generation by using preview control of zero-moment point. In: IEEE. **Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on**. [S.l.], 2003. v. 2, p. 1620–1626.

KAJITA, S.; NAGASAKI, T.; KANEKO, K.; HIRUKAWA, H. Zmp-based biped running control. **IEEE Robotics & Automation Magazine**, IEEE, v. 14, n. 2, p. 63–72, 2007.

KATAYAMA, T.; OHKI, T.; INOUE, T.; KATO, T. Design of an optimal controller for a discrete-time system subject to previewable demand. **International Journal of Control**, Taylor & Francis, v. 41, n. 3, p. 677–699, 1985.

KOFINAS, N. **Forward and inverse kinematics for the NAO humanoid robot**. Tese (Doutorado) — Technical University of Crete, Greece, 2012.

KUO, A. D.; DONELAN, J. M. Dynamic principles of gait and their clinical implications. **Physical therapy**, American Physical Therapy Association, v. 90, n. 2, p. 157, 2010.

LOPEZ-MOBILIA, A. **Inverse Kinematics Kicking in the Humanoid RoboCup Simulation League**. Tese (Doutorado) — Master's Thesis, University of Texas at Austin, 2012.

MACALPINE, P.; BARRETT, S.; URIELI, D.; VU, V.; STONE, P. Design and optimization of an omnidirectional humanoid walk: A winning approach at the RoboCup 2011 3D simulation competition. In: **Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI)**. [S.l.: s.n.], 2012.

PARK, J.; YOUM, Y. General zmp preview control for bipedal walking. In: IEEE. **Robotics and Automation, 2007 IEEE International Conference on**. [S.l.], 2007. p. 2682–2687.

RÖFER, T.; LAUE, T.; MÜLLER, J.; FABISCH, A.; FELDPAUSCH, F.; GILLMANN, K.; GRAF, C.; HAAS, T. J. de; HÄRTL, A.; HUMANN, A. et al. B-human team report and code release 2011. **Disponível apenas online: [http://www. b-human. de/downloads/bhuman11_coderelease. pdf](http://www.b-human.de/downloads/bhuman11_coderelease.pdf)**, 2011.

SAKKA, S.; HAYOT, C.; LACOUTURE, P. A generalized 3d inverted pendulum model to represent human normal walking. In: IEEE. **Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on**. [S.l.], 2010. p. 486–491.

SEDERBERG, T. **BYU Bézier curves**. [S.l.]: Chapter, 2012.

SEYFARTH, A.; GEYER, H.; HERR, H. Swing-leg retraction: a simple control model for stable running. **Journal of Experimental Biology**, The Company of Biologists Ltd, v. 206, n. 15, p. 2547–2555, 2003.

SHAFII, N. Development of an optimized omnidirectional walk engine for humanoid robots. Citeseer, 2015.

SHAFII, N.; LAU, N.; REIS, L. P. Learning a fast walk based on zmp control and hip height movement. In: IEEE. **Autonomous Robot Systems and Competitions (ICARSC), 2014 IEEE International Conference on**. [S.l.], 2014. p. 181–186.

SHAHBAZI, M.; BABUŠKA, R.; LOPES, G. A. D. Unified modeling and control of walking and running on the spring-loaded inverted pendulum. **IEEE Transactions on Robotics**, v. 32, n. 5, p. 1178–1195, Oct 2016. ISSN 1552-3098.

SIMÕES, M. A. C.; SOUZA, J. R. de; FRIAS, D.; LARANJEIRA, C.; REIS, C. E.; ARGOLLO, E.; JR, S. S. Robôs humanoides em ambiente de simulação 3d: a experiência do acso. In: **Brazilian Humanoid Robot Workshop (BRAHUR)**. São Bernardo do Campo, Brazil: [s.n.], 2016. Disponível em: <<http://fei.edu.br/brahur2016/trab.html>>.

STOECKER, J.; VISSER, U. Roboviz: Programmable visualization for simulated soccer. **RoboCup 2011: Robot Soccer World Cup XV**, Springer, p. 282–293, 2012.

STROM, J.; SLAVOV, G.; CHOWN, E. Omnidirectional walking using zmp and preview control for the nao humanoid robot. In: SPRINGER. **Robot Soccer World Cup**. [S.l.], 2009. p. 378–389.

TAJIMA, R.; HONDA, D.; SUGA, K. Fast running experiments involving a humanoid robot. In: IEEE. **Robotics and Automation, 2009. ICRA'09. IEEE International Conference on**. [S.l.], 2009. p. 1571–1576.

TAKENAKA, T.; MATSUMOTO, T.; YOSHIIKE, T.; SHIROKURA, S. Real time motion generation and control for biped robot -2nd report: Running gait pattern generation-. In: **2009 IEEE/RSJ International Conference on Intelligent Robots and Systems**. [S.l.: s.n.], 2009. p. 1092–1099. ISSN 2153-0858.

VAUGHAN, C. L.; DAVIS, B. L.; O'CONNOR, J. C. **Dynamics of human gait**. [S.l.]: Human Kinetics Publishers Champaign, Illinois, 1992. v. 2.

VUKOBRATOVIĆ, M.; BOROVIAC, B. Zero-moment point—thirty five years of its life. **International Journal of Humanoid Robotics**, World Scientific, v. 1, n. 01, p. 157–173, 2004.

VUKOBRATOVIĆ, M.; BOROVIAC, B.; SURLA, D.; STOKIĆ, D. **Biped locomotion: dynamics, stability, control and application**. [S.l.]: Springer Science & Business Media, 2012. v. 7.

VUKOBRATOVIĆ, M.; JURIĆIĆ, D. Contribution to the synthesis of biped gait. **IEEE Transactions on Biomedical Engineering**, IEEE, n. 1, p. 1–6, 1969.

YU, H.; LI, M.; GUO, W.; CAI, H. Stance control of the slip hopper with adjustable stiffness of leg spring. In: **2012 IEEE International Conference on Mechatronics and Automation**. [S.l.: s.n.], 2012. p. 2007–2012. ISSN 2152-7431.