



UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”
Instituto de Ciência e Tecnologia
Câmpus de Sorocaba

GUILHERME DOUBEK LOPES

Desenvolvimento de um ambiente de cosimulação
Adams-Simulink para o controle da locomoção de um robô
quadrúpede

Sorocaba
2023

GUILHERME DOUBEK LOPES

DESENVOLVIMENTO DE UM AMBIENTE DE
COSIMULAÇÃO ADAMS-SIMULINK PARA O CONTROLE
DA LOCOMOÇÃO DE UM ROBÔ QUADRÚPEDE

Trabalho de Conclusão de Curso apresentado
ao Instituto de Ciência e Tecnologia de Sorocaba,
Universidade Estadual Paulista (UNESP),
como parte dos requisitos para obtenção do
grau de Bacharel em Engenharia de Controle e
Automação.

Orientador: Prof. Dr. Mauricio Becerra Vargas

Sorocaba

2023

L864d

Lopes, Guilherme Doubek

Desenvolvimento de um ambiente de cosimulação Adams-Simulink para o controle da locomoção de um robô quadrúpede / Guilherme Doubek Lopes. -- Sorocaba, 2023

78 p. : il., tabs., fotos

Trabalho de conclusão de curso (Bacharelado - Engenharia de Controle e Automação) - Universidade Estadual Paulista (Unesp), Instituto de Ciência e Tecnologia, Sorocaba

Orientador: Mauricio Becerra Vargas

1. Robótica. 2. Locomoção. 3. Sistemas não lineares. 4. Engenharia auxiliada por computador. 5. Simulink. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Instituto de Ciência e Tecnologia, Sorocaba. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.



**DESENVOLVIMENTO DE UM AMBIENTE DE
COSIMULAÇÃO ADAMS-SIMULINK PARA O CONTROLE
DA LOCOMOÇÃO DE UM ROBÔ QUADRÚPEDE**

GUILHERME DOUBEK LOPES

**ESTE PROJETO FINAL DE CURSO FOI JULGADO ADEQUADO
COMO PARTE DO REQUISITO PARA A OBTENÇÃO DO GRAU DE
BACHAREL EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO**

Prof. Dr. Everson Martins
Coordenador

BANCA EXAMINADORA:

Prof. Dr. MAURICIO BECERRA VARGAS
Orientador/UNESP – Campus de Sorocaba

Prof. Dr. FELIX MAURICIO ESCALANTE ORTEGA
UNESP – Campus de Sorocaba

Eng. DANIEL TEIXEIRA DE PAULA
UNESP – Campus de Sorocaba

Dezembro de 2023

AGRADECIMENTOS

Agradeço a toda minha família, principalmente aos meus pais, Fábio e Giani, por todo suporte, paciência, incentivo e confiança que depositaram em mim desde sempre e sem os quais não teria chegado onde cheguei.

Agradeço ao Prof. Dr. Mauricio Becerra Vargas por toda a orientação fornecida e pela incessante colaboração durante toda a execução do projeto.

Agradeço aos meus amigos Giovanni Dotta, Lucas Lima e Matheus Nunes pela parceria nos momentos de luta e de glória que a graduação nos proporcionou, inclusive na execução deste projeto final de curso.

Agradeço ao Daniel Teixeira de Paula e ao Dimitri Margutti Cezar por toda a contribuição que tiveram na confecção e no entendimento de conceitos referentes à execução este trabalho.

Por fim, expresso minha gratidão a todo o corpo docente, funcionários e prestadores de serviço da UNESP, cujas contribuições não apenas moldaram minha formação, mas também foram e são fundamentais para manter a excelência desta instituição acadêmica.

LOPES, G. D. **Desenvolvimento de um ambiente de cosimulação Adams-Simulink para o controle da locomoção de um robô quadrúpede.** 2023. 78 p. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Controle e Automação) – Instituto de Ciência e Tecnologia, Universidade Estadual Paulista, Sorocaba, 2023.

RESUMO

O presente trabalho propõe a criação de um ambiente virtual para a implementação de técnicas de controle da locomoção em um protótipo virtual de robô quadrúpede com pernas contendo dois graus de liberdade e com movimento restrito ao plano sagital, modelado com base nas dimensões do *Mini Cheetah*. As etapas do trabalho incluem a construção do protótipo virtual do robô quadrúpede, a implementação do processo de geração de trajetórias cinemáticas e a cosimulação via Adams-Simulink da execução da marcha trote pelo robô por meio de uma malha de controle cinemático de posição com acionamento por velocidade utilizando um controlador proporcional. Realizou-se a prototipagem virtual no Adams, onde foram feitas a importação e manipulação do desenho tridimensional do quadrúpede, inserção de pontos de construção para definição das posições das juntas e validação dos comprimentos das patas do robô com base nos comprimentos das patas do *Mini Cheetah*, definição das propriedades de massa, inserção de juntas e criação de contato entre as pernas do robô e o solo. Implementou-se o processo de geração de trajetórias cinemáticas no MATLAB, o qual consistiu na implantação do modelo matemático do *Central Pattern Generator*, criação das curvas de Bézier e resolução da cinemática inversa do robô quadrúpede. O protótipo e as trajetórias cinemáticas foram integradas em uma cosimulação por meio de uma malha de controle cinemático de posição com acionamento por velocidade construída no Simulink com um controlador proporcional. Os resultados obtidos foram analisados e comparados em gráficos e englobaram a geração das trajetórias cinemáticas, a atuação do controlador implementado, os torques nas juntas motorizadas e a estabilidade do robô durante a execução da marcha trote. Tais resultados demonstraram a estabilidade do robô durante a movimentação, a execução de trajetórias cinemáticas coerentes, compassadas e sem esforços excessivos nas juntas e, além disso, a facilidade com que os parâmetros da cosimulação podem ser alterados, permitindo a utilização do ambiente virtual construído para a implementação simplificada de técnicas de controle da locomoção.

Palavras-chave: prototipagem virtual; Central Pattern Generator; curvas de Bézier; cinemática inversa; controle da locomoção.

LOPES, G. D. **Development of an Adams-Simulink cosimulation environment to control the locomotion of a quadruped robot.** 2023. 78 p. Final Paper (Bachelor's degree in Control and Automation Engineering) – Instituto de Ciência e Tecnologia, Universidade Estadual Paulista, Sorocaba, 2023.

ABSTRACT

This paper proposes the creation of a virtual environment for implementing locomotion control techniques in a virtual prototype of a quadrupedal robot with legs containing two degrees of freedom and with movement restricted to the sagittal plane, modeled based on the dimensions of the Mini Cheetah. The stages of the project include the construction of the quadrupedal robot virtual prototype, the implementation of the process of generating kinematic trajectories, and the cosimulation via Adams-Simulink of the execution of the trot gait by the robot through a kinematic position control loop activated by velocity using a proportional controller. Virtual prototyping was carried out in Adams, where the three-dimensional design of the quadruped was imported and manipulated, construction points were inserted to define the positions of the joints and to validate the lengths of the robot's legs based on the lengths of the Mini Cheetah legs, mass properties were defined, joints were inserted and contact was created between the robot's legs and the ground. The process of generating kinematic trajectories was implemented in MATLAB, which consisted of implementing the Central Pattern Generator mathematical model, creating Bézier curves and solving the inverse kinematics of the quadruped robot. The prototype and kinematic trajectories were integrated into a cosimulation through a kinematic position control loop with velocity actuation built in Simulink with a proportional controller. The results obtained were analyzed and compared in graphs and included the generation of kinematic trajectories, the performance of the implemented controller, the torques on the motorized joints and the stability of the robot during the execution of the trot gait. These results demonstrated the stability of the robot during movement, the execution of coherent and paced kinematic trajectories without excessive effort on the joints and, in addition, the ease with which the cosimulation parameters can be changed, allowing the use of the virtual environment built for the simplified implementation of locomotion control techniques.

Keywords: virtual prototyping; Central Pattern Generator; Bézier curves; inverse kinematics; locomotion control.

LISTA DE FIGURAS

Figura 1 – Representação das etapas pertinentes ao controle de um robô com destaque para o controle da locomoção, dividido em geração e controle do movimento	13
Figura 2 – Exemplos de diagramas da marcha galope para um cavalo (<i>horse</i>) e para um guepardo (<i>cheetah</i>)	15
Figura 3 – Campo de vetores (<i>vector field</i>) plotado em um plano de fase (<i>phase plane</i>) .	20
Figura 4 – Trajetórias formadas pelo campo de vetores (<i>vector field</i>) em um plano de fase (<i>phase plane</i>) constituindo um diagrama ou retrato de fase (<i>phase diagram</i> ou <i>phase portrait</i>)	21
Figura 5 – Exemplos de ciclos-limite estável, instável e parcialmente estável	22
Figura 6 – Ponto $P_0(t)$ parametrizado em função de t e localizado no segmento de reta que interpola os pontos $P_A(x_A, y_A)$ e $P_B(x_B, y_B)$, sendo a posição de $P_0(0)$ correspondente com a do ponto $P_A(x_A, y_A)$ e a posição de $P_0(1)$ correspondente com a do ponto $P_B(x_B, y_B)$	26
Figura 7 – Interpolações lineares em um sistema de coordenadas bidimensional criadas a partir dos pontos $P_A(x_A, y_A)$, $P_B(x_B, y_B)$ e $P_C(x_C, y_C)$	27
Figura 8 – Construção de uma curva de Bézier definida por três pontos de controle a partir da variação do parâmetro t entre 0 e 1	27
Figura 9 – Cadeia cinemática aberta com dois graus de liberdade e com os ângulos θ_1 e θ_2 destacados	29
Figura 10 – Medidas de ângulo e de comprimento feitas na cadeia cinemática aberta com 2 DOF para o equacionamento de θ_1	29
Figura 11 – Medidas de ângulo e de comprimento feitas na cadeia cinemática aberta com 2 DOF para o equacionamento de θ_2	31
Figura 12 – Representação das duas posições que a cadeia cinemática aberta de 2 DOF pode assumir para os mesmos dados de posição e de orientação do sistema de coordenadas com origem no TCP	32
Figura 13 – Correspondência entre as etapas da prototipagem virtual e os módulos do Adams que podem ser utilizados para executá-las	34
Figura 14 – Modelo tridimensional original do <i>Mini Cheetah</i> , à esquerda, em comparação com o desenho do robô quadrúpede retirado do <i>GrabCAD</i> , à direita	36
Figura 15 – Partes constituintes de cada uma das patas do <i>Mini Cheetah</i>	36

Figura 16 – Ponto de construção (em conjunto com um <i>marker</i>) inserido no centro de massa da engrenagem simétrica constituinte do <i>lower link</i> (e que intersecta o <i>link</i> superior), à esquerda, e pontos coplanares (em conjunto com <i>markers</i>) associados ao motor de joelho (<i>knee actuator</i>) e ao pé de borracha (<i>foot</i>), à direita	37
Figura 17 – Modelo tridimensional do robô quadrúpede com junta planar e elementos das juntas motorizadas ativas (juntas de revolução e <i>Rotational Joint Motions</i>) destacados	41
Figura 18 – Estrutura externa do pé de borracha do modelo de robô quadrúpede do Adams/View, à esquerda, e esfera criada para contato com o solo encoberta pela estrutura externa do pé de borracha, à direita	41
Figura 19 – Correspondência entre os osciladores 1, 2, 3 e 4 e as pernas LF, RF, RH e LH associadas	44
Figura 20 – Malha de controle cinemático de posição com acionamento por velocidade que integra os <i>softwares</i> Adams, MATLAB e Simulink na cosimulação . . .	47
Figura 21 – Ângulos θ_1 e θ_2 medidos no protótipo virtual construído no Adams e utilizados na malha de controle cinemático do Simulink	47
Figura 22 – Parâmetros definidos para a exportação das oito variáveis de entrada (<i>INPUT</i>) e das oito variáveis de saída (<i>OUTPUT</i>) em uma única planta de controle por meio do Adams/Controls	49
Figura 23 – Parâmetros definidos para a configuração do bloco representativo do protótipo virtual no Simulink	49
Figura 24 – Controle cinemático de posição com acionamento por velocidade implementado no Simulink para a perna dianteira esquerda (LF) do robô quadrúpede .	50
Figura 25 – Variáveis de estado x_{LF} e y_{LF} do oscilador 1, associado à perna dianteira esquerda (LF) do robô quadrúpede, plotadas ao longo do tempo	51
Figura 26 – Variáveis de estado x_{LF} , x_{RF} , x_{RH} e x_{LH} plotadas ao longo do tempo com condições iniciais nulas	52
Figura 27 – Diagrama de fase do oscilador 1, associado à perna LF do robô quadrúpede, contendo ciclo-limite de amplitude $\sqrt{\mu}$ unitária	53
Figura 28 – Variáveis de estado x_{LF} , x_{RF} , x_{RH} e x_{LH} plotadas ao longo do tempo com condições iniciais iguais a 2,0, 0,5, -0,5 e -1,5, respectivamente	53
Figura 29 – Curva de Bézier projetada para a perna LF, com pontos extremos destacados, plotada com base nas variáveis de estado resultantes do oscilador 1 em um sistema de coordenadas cartesianas	55
Figura 30 – Curvas de Bézier projetadas para as quatro pernas (LF, RF, RH e LH) do robô quadrúpede e implementadas no MATLAB	56
Figura 31 – Ângulos θ_1 e θ_2 das juntas do robô quadrúpede calculados para a perna LF e plotados ao longo do tempo	57

Figura 32 – Ângulos $\theta_{1(LF)}$, $\theta_{1(RF)}$, $\theta_{1(RH)}$ e $\theta_{1(LH)}$ plotados ao longo do tempo	58
Figura 33 – Ângulos $\theta_{2(LF)}$, $\theta_{2(RF)}$, $\theta_{2(RH)}$ e $\theta_{2(LH)}$ plotados ao longo do tempo	58
Figura 34 – Gráfico superior contendo comparação entre os ângulos $\theta_{1(LF)}$ desejados e medidos ao longo do tempo e gráfico inferior contendo erro de posição angular resultante da comparação entre os ângulos $\theta_{1(LF)}$ desejados e medidos ao longo do tempo	59
Figura 35 – Gráfico superior contendo comparação entre os ângulos $\theta_{2(LF)}$ desejados e medidos ao longo do tempo e gráfico inferior contendo erro de posição angular resultante da comparação entre os ângulos $\theta_{2(LF)}$ desejados e medidos ao longo do tempo	59
Figura 36 – Trajetória do pé de borracha da perna LF durante a cosimulação	60
Figura 37 – Trajetórias dos pés de borracha das pernas LF, RF, RH e LH durante a cosimulação	61
Figura 38 – Medição da altura (apenas no eixo y) de um <i>marker</i> localizado no centro de massa do corpo central do protótipo virtual do Adams ao longo do tempo de cosimulação	62
Figura 39 – Medição do ângulo de rotação em torno do eixo z de um <i>marker</i> localizado no centro de massa do corpo central do protótipo virtual do Adams ao longo do tempo de cosimulação	62
Figura 40 – Medição de torque na junta 1 da perna LF do protótipo virtual durante a execução da marcha trote na cosimulação via Adams-Simulink	63
Figura 41 – Medição de torque na junta 2 da perna LF do protótipo virtual durante a execução da marcha trote na cosimulação via Adams-Simulink	64

LISTA DE TABELAS

Tabela 1 – Fases relativas características das marchas <i>walk</i> , <i>trot</i> , <i>transverse gallop</i> e <i>pronk</i>	16
Tabela 2 – Comparação entre os comprimentos das pernas do modelo tridimensional de robô quadrúpede do Adams e de uma das patas do <i>Mini Cheetah</i>	38
Tabela 3 – Valores de massa, volume e densidade de cada elemento constituinte do modelo tridimensional do robô quadrúpede baseado no <i>Mini Cheetah</i>	38
Tabela 4 – Valores de massa, volume e densidade dos elementos agrupados no modelo tridimensional do robô quadrúpede baseado no <i>Mini Cheetah</i>	39
Tabela 5 – Juntas de revolução definidas para o modelo tridimensional de robô quadrúpede no Adams	40
Tabela 6 – Novos valores de massa, volume e densidade definidos para a estrutura composta por <i>lower link</i> e <i>foot</i> do modelo tridimensional de robô quadrúpede baseado no <i>Mini Cheetah</i>	42
Tabela 7 – Juntas de revolução definidas para o modelo tridimensional do robô quadrúpede no Adams	43
Tabela 8 – Valores dos parâmetros utilizados para a implementação do modelo matemático do <i>Central Pattern Generator</i> no MATLAB	43
Tabela 9 – Valores referentes às diferenças de fases entre os osciladores (θ_{ji}) do protótipo virtual do robô quadrúpede para a marcha trote (<i>trot gait</i>)	44
Tabela 10 – Condições iniciais definidas para as variáveis de estado constituintes do modelo matemático do CPG implementado no MATLAB	51

SUMÁRIO

1	INTRODUÇÃO	13
2	REVISÃO DE LITERATURA	15
2.1	Princípios da locomoção	15
2.2	<i>Central Pattern Generator</i> (CPG)	16
3	FUNDAMENTAÇÃO TEÓRICA	19
3.1	Modelo matemático do <i>Central Pattern Generator</i>	19
3.1.1	<i>Abordagem geométrica para a resolução de problemas</i>	19
3.1.2	<i>Ciclo-limite</i>	21
3.1.3	<i>Bifurcação de Hopf</i>	22
3.1.4	<i>Osciladores modificados de Hopf</i>	24
3.2	Curvas de Bézier	26
3.3	Cinemática inversa de uma cadeia cinemática aberta com dois graus de liberdade (<i>Degrees of Freedom</i> ou DOF)	28
3.4	Prototipagem virtual	33
4	METODOLOGIA	35
4.1	Prototipagem virtual de um robô quadrúpede no Adams	35
4.1.1	<i>Importação e manipulação do desenho tridimensional de um robô quadrúpede</i>	35
4.1.2	<i>Inserção de pontos de construção para definição das posições das juntas e validação dos comprimentos das patas do robô</i>	36
4.1.3	<i>Definição das propriedades de massa e inserção de juntas motorizadas no robô quadrúpede</i>	38
4.1.4	<i>Inserção de juntas no robô quadrúpede</i>	39
4.1.5	<i>Criação de contato entre o robô quadrúpede e o solo</i>	40
4.2	Implementação do processo de geração de trajetórias cinemáticas no MATLAB	43
4.2.1	<i>Implementação do modelo matemático do <i>Central Pattern Generator</i> no MATLAB</i>	43
4.2.2	<i>Implementação das curvas de Bézier no MATLAB</i>	44
4.2.3	<i>Implementação da cinemática inversa no MATLAB</i>	46
4.3	Cosimulação entre Adams e Simulink	46
4.3.1	<i>Integração do protótipo do Adams à malha de controle cinemático</i>	47
4.3.2	<i>Configuração da malha de controle cinemático de posição com acionamento por velocidade no Simulink</i>	48
5	RESULTADOS E DISCUSSÕES	51

5.1	Resultados da implementação do modelo matemático do CPG no MATLAB .	51
5.2	Resultados da implementação das curvas de Bézier no MATLAB	54
5.3	Resultados da implementação da cinemática inversa no MATLAB	56
5.4	Resultados da cosimulação Adams-Simulink	58
5.4.1	Ângulos das juntas do protótipo virtual do Adams	58
5.4.2	Trajetórias cinemáticas desenvolvidas pelas pernas do protótipo virtual do Adams	60
5.5	Estabilidade da locomoção do robô quadrúpede	61
5.6	Torque nas juntas das pernas do robô quadrúpede	63
6	CONCLUSÃO	65
	REFERÊNCIAS	66
	APÊNDICE A - Variáveis de estado x e y dos osciladores associados às pernas LF, RF, RH e LH do robô quadrúpede plotadas ao longo do tempo	69
	APÊNDICE B - Diagramas de fase dos osciladores associados às pernas LF, RF, RH e LH do robô quadrúpede contendo ciclos-limite de amplitude unitária . . .	70
	APÊNDICE C - Curvas de Bézier projetadas para as pernas LF, RF, RH e LH no MATLAB e plotadas em um sistema de coordenadas cartesianas	71
	APÊNDICE D - Comparação entre os ângulos $\theta_{1(RF)}$ desejados e medidos ao longo do tempo (gráfico superior) em conjunto com o erro de posição angular correspondente (gráfico inferior)	72
	APÊNDICE E - Comparação entre os ângulos $\theta_{2(RF)}$ desejados e medidos ao longo do tempo (gráfico superior) em conjunto com o erro de posição angular correspondente (gráfico inferior)	73
	APÊNDICE F - Comparação entre os ângulos $\theta_{1(RH)}$ desejados e medidos ao longo do tempo (gráfico superior) em conjunto com o erro de posição angular correspondente (gráfico inferior)	74
	APÊNDICE G - Comparação entre os ângulos $\theta_{2(RH)}$ desejados e medidos ao longo do tempo (gráfico superior) em conjunto com o erro de posição angular correspondente (gráfico inferior)	75
	APÊNDICE H - Comparação entre os ângulos $\theta_{1(LH)}$ desejados e medidos ao longo do tempo (gráfico superior) em conjunto com o erro de posição angular correspondente (gráfico inferior)	76

APÊNDICE I - Comparação entre os ângulos $\theta_{2(LH)}$ desejados e medidos ao longo do tempo (gráfico superior) em conjunto com o erro de posição angular correspondente (gráfico inferior)	77
APÊNDICE J - Curvas de Bézier executadas pelas pernas LF, RF, RH e LH durante a cosimulação Adams-Simulink	78

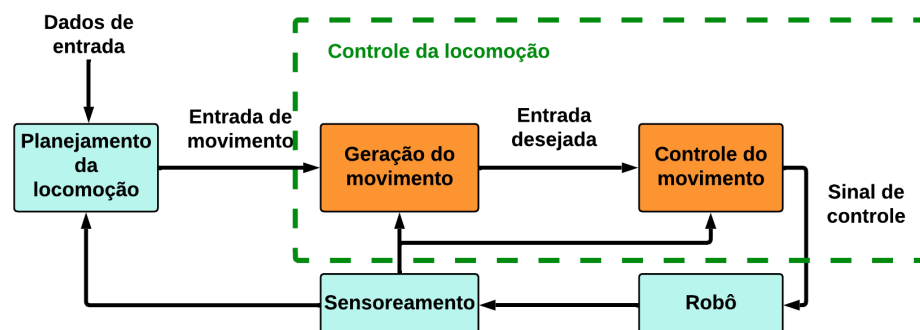
1 INTRODUÇÃO

A natureza sempre foi fonte de inspiração para o homem, seja para a melhoria da aerodinâmica de um trem-bala pautada em atributos físicos de determinadas aves, seja para a manutenção da temperatura constante em um prédio a partir de um *design* arquitetônico baseado em cupinzeiros (BBC, 2017; SEBRAE, 2018). A área multidisciplinar do conhecimento que estuda os meios pelos quais a natureza supera determinados obstáculos e que procura replicá-los em aplicações humanas, como os exemplos supracitados, é denominada biomimética (SINGULARIDADES, 2020). Dentre as diversas vertentes englobadas pela mesma, uma das que se destacam é a locomoção por pernas, já que possibilita o desenvolvimento de robôs com forte apelo midiático, como o *Atlas*, capaz de executar manobras de *parkour*, e o *Mini Cheetah*, apto a realizar *back flips* (DEVITO, 2021; GALLAGHER, 2021).

Além do destaque na mídia, robôs com pernas, mais especificamente os quadrúpedes, podem ser usados na exploração de ambientes perigosos, inspeção em ambientes industriais de difícil acesso e em pesquisa e desenvolvimento (DYNAMICS, 2023). Esta última aplicação tem se tornado bastante útil para o estudo de estruturas biológicas relacionadas à movimentação e implementação de algoritmos de aprendizado de máquina e de técnicas de controle da locomoção (IJSPEERT, 2008).

O controle da locomoção faz parte do contexto geral de controle de um robô e é destacado no diagrama ilustrado na Figura 1, onde é possível observar que o mesmo é dividido em duas partes: geração do movimento, que consiste no processo de criação das trajetórias cinemáticas, e controle do movimento, que viabiliza a execução de tais trajetórias de acordo com a entrada desejada.

Figura 1 – Representação das etapas pertinentes ao controle de um robô com destaque para o controle da locomoção, dividido em geração e controle do movimento



Fonte: Autoria própria.

O controle da locomoção é essencial, contudo, se demonstrou um problema implementá-lo no *Cheetah 3* do MIT (Massachusetts Institute of Technology), por exemplo, já que, por

conta de sua estrutura grande e pesada, falhas de implementação poderiam danificar o robô e machucar pessoas ao redor do mesmo, por isso, para facilitar este tipo de aplicação, foi criado o *Mini Cheetah*, com uma estrutura compacta, leve e robusta. (KATZ, 2018).

No entanto, apesar da importância do tema e da dificuldade de implementação apresentada, não há ampla difusão de uma plataforma virtual que permita a aplicação de estratégias de controle da locomoção em robôs quadrúpedes. Plataformas como o Webots fornecem um ambiente para a simulação robótica, o Adams permite a construção de robôs com pernas e o MATLAB permite a geração de trajetórias cinemáticas, no entanto, tais *softwares* carecem de uma aplicação focada que permita apenas a inserção de poucos parâmetros para a execução de simulações e de análises.

Por isso, o objetivo deste trabalho é construir um ambiente virtual para a implementação de técnicas de controle da locomoção em um protótipo virtual de robô quadrúpede por meio das seguintes etapas: construção de um protótipo virtual de robô quadrúpede baseado nas dimensões do *Mini Cheetah* no Adams com pernas contendo dois graus de liberdade e restrição do movimento ao plano sagital, implementação do processo de geração de trajetórias cinemáticas no MATLAB e cosimulação via Adams-Simulink do protótipo virtual executando o trote, que consiste em uma marcha simples e simétrica, por meio de uma malha de controle cinemático de posição com acionamento por velocidade utilizando um controlador proporcional.

2 REVISÃO DE LITERATURA

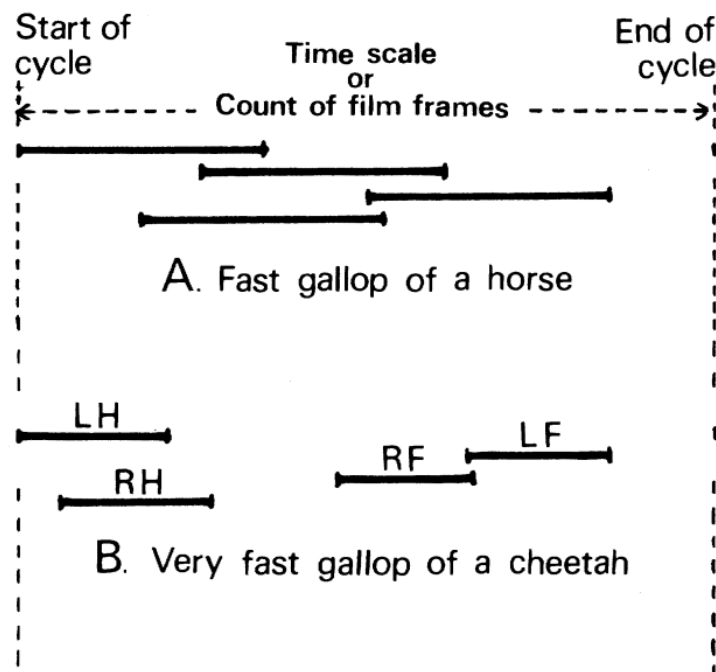
Nesta seção faz-se o levantamento da literatura relevante para a apresentação de conceitos norteadores deste trabalho e que consistem em dois tópicos principais: princípios da locomoção e *Central Pattern Generator* (CPGs).

2.1 Princípios da locomoção

A locomoção por pernas é a abordagem geralmente encontrada na natureza para o deslocamento de um local ao outro (SHARBAFI; SEYFARTH, 2017). A mesma ocorre por meio da interação do animal com o solo, a qual, por sua vez, gera forças de reação capazes de propulsioná-lo ou cessar sua movimentação.

Este tipo de locomoção é classificado a partir de marchas (*gaits*), isto é, a sequência com que as pernas entram contato com o solo ao longo do tempo (SHARBAFI; SEYFARTH, 2017). Os *gaits* podem ser representados por diagramas de marchas, conforme mostra o exemplo da Figura 2.

Figura 2 – Exemplos de diagramas da marcha galope para um cavalo (*horse*) e para um guepardo (*cheetah*)



Fonte: Hildebrand (1977).

O diagrama de fase apresentado se refere a um animal quadrúpede cujas pernas dianteira esquerda, dianteira direita, traseira esquerda e traseira direita são representadas por LF (*left fore leg*), RF (*right fore leg*), LH (*left hind leg*) e RH (*right hind leg*).

Além disso, os traços contidos na Figura 2 indicam o período em que cada perna permanece em contato com o solo durante a locomoção, o qual é denominado *stance phase* (fase de contato), enquanto o período em que a perna não está em contato com o chão é chamado de *swing phase* (fase de balanço). O ciclo de locomoção como um todo é constituído pelas duas fases e o período de duração do mesmo é denominado *stride period* (SHARBAFI; SEYFARTH, 2017).

A relação entre *stance phase* (T_{st}) e *stride period* ($T_{st} + T_{sw}$) é denominada *duty factor* e expressa pela equação (1).

$$\beta = \frac{T_{st}}{T_{st} + T_{sw}} \quad (1)$$

O *duty factor* representa a porcentagem do período de locomoção em que determinada perna está em contato com o solo e pode ser utilizado para classificar os tipos de marcha entre caminhadas ou corridas: marchas com valores de β maiores que 0,5 são classificadas como caminhadas, enquanto as que apresentam *duty factor* menor que 0,5 são classificadas como corridas (ALEXANDER, 1984).

Além desta classificação, pode-se utilizar o espaçamento temporal, ou fase relativa ϕ , com relação a determinada perna, para classificar as marchas em simétricas ou assimétricas. Marchas simétricas são aquelas em que pernas sagitalmente opostas possuem o mesmo *duty factor* e fase relativa igual a, no máximo, 0,5, enquanto marchas assimétricas são aquelas que não possuem estas características (ALEXANDER, 1984).

Exemplos de marchas simétricas consistem em caminhada (*walk*) e trote (*trot*), enquanto de marchas assimétricas consistem em galope (*transverse gallop*) e salto (*pronk*). Ademais, algumas fases relativas típicas, medidas com base na perna LF, são exibidas na Tabela 1.

Tabela 1 – Fases relativas características das marchas *walk*, *trot*, *transverse gallop* e *pronk*

Tipo de marcha	ϕ_{LF}	ϕ_{RF}	ϕ_{RH}	ϕ_{LH}
<i>Walk</i>	0	0,5	0,25	0,75
<i>Trot</i>	0	0,5	0	0,5
<i>Transverse gallop</i>	0	0,1	0,6	0,5
<i>Pronk</i>	0	0	0	0

Fonte: Autoria própria.

2.2 Central Pattern Generator (CPG)

A geração dos padrões de locomoção explicitados na subseção anterior ocorre por meio de uma estrutura biológica conhecida como *Central Pattern Generator* (CPG), um tipo de circuito neural encontrado em animais vertebrados e invertebrados que produz padrões coordenados de

sinais de saída elétricos, usados para ações como caminhar, nadar ou voar (IJSPEERT, 2008).

Por ser uma estrutura que não necessita de impulsos externos para funcionar, é replicada com frequência na área da robótica, onde permite a criação de robôs mais eficientes e independentes de *feedback* sensorial constante. Segundo Ijspeert (2008), modelos de CPGs podem ser implementados seguindo diferentes vertentes, de modo que algumas das principais dizem respeito ao uso de redes neurais, modelo que foca no comportamento biológico da estrutura geradora de padrões de locomoção, e de osciladores não-lineares acoplados, que são representações puramente matemáticas dos CPGs, permitindo maior possibilidade de variação de parâmetros quando comparada com a primeira vertente apresentada.

Diversas estratégias envolvendo osciladores têm sido implementadas para destacar aspectos específicos de interesse. Um exemplo é o modelo proposto por Golubitsky *et al.* (1998), que utiliza osciladores não-lineares acoplados para animais com $2n$ pernas. No contexto desse modelo, é sugerido o uso de uma rede de, no mínimo, $4n$ células para gerar todas as relações de fase observadas durante as marchas. O autor destaca a importância de considerar a topologia da rede de osciladores para identificar simetrias, o que pode simplificar o acoplamento entre as células interligadas.

Já Righetti e Ijspeert (2006) reforçam e equacionam outra característica importante em sua pesquisa, a qual foi integrada ao *RobotCub*, robô humanoide de 54 graus de liberdade (SANDINI; METTA; VERNON, 2004). No âmbito desse trabalho, foi estabelecida uma equação fundamentada na constante elástica de uma mola para a definição de uma frequência variável em consonância com a fase do oscilador. Esse enfoque revelou-se de grande valia para representar as distintas frequências associadas às fases de balanço e de contato, proporcionando uma abordagem mais precisa e dinâmica à modelagem desses padrões de movimento.

A progressão deste estudo é conduzida tanto por Righetti e Ijspeert (2008), que incorporam osciladores de Hopf com frequência dependente da fase e com ciclo-limite de topologia independente da frequência, quanto por Wu *et al.* (2013), que recalibra os parâmetros da frequência variável para permitir uma transição mais fluida entre diferentes padrões de marcha.

O crescente aumento no poder computacional e o aprimoramento de técnicas de aprendizado de máquinas dos últimos anos têm possibilitado a utilização de modelos de controle preditivo em robôs com pernas, com otimização efetuada a partir de técnicas de aprendizado de máquina. Lele *et al.* (2020), por exemplo, faz o uso de um CPG baseado em redes neurais pulsantes (*Spiking Neural Networks*) em conjunto com técnicas de aprendizado de máquina por reforço para ensinar um robô hexápode a se locomover sem conhecimento prévio do padrão de disparo correto das células constituintes da rede neural ou das trajetórias a serem efetuadas pelas pernas.

Os modelos de replicação do CPG apresentados podem ser simulados em robôs com pernas para diversas aplicações, seja para testes de performance de um modelo constituído por

osciladores de Hopf em diferentes terrenos, conforme explicitam Ma, Liang e Tian (2020), seja para estudar o controle da troca de marchas em robôs quadrúpedes, conforme descrevem Lipeng e Bing (2023). Embora muitas das implementações sejam feitas por meio de cosimulações envolvendo o MATLAB e o Adams (AN; HENG; YONGZHENG, 2013; XIE *et al.*, 2019; MORAIS; SINGH; ACHARYA, 2021), como é o caso dos estudos supraindicados, não se observa preocupação com a construção de uma plataforma que permita outras implementações além das descritas nos trabalhos, levando à necessidade de se construir um ambiente virtual focado a cada nova aplicação a ser efetuada.

3 FUNDAMENTAÇÃO TEÓRICA

Os conceitos teóricos abordados nesta seção estão relacionados aos seguintes processos que viabilizam a implementação da locomoção de um robô quadrúpede: geração dos padrões de marcha, geração das trajetórias realizadas pelas patas do robô, obtenção dos ângulos das juntas correspondentes às trajetórias almejadas e prototipagem virtual.

A geração dos padrões de marcha é feita pelo modelo matemático do *Central Pattern Generator* (CPG) construído com base em osciladores de Hopf modificados, a geração das trajetórias das patas do robô se baseia na utilização de curvas paramétricas denominadas curvas de Bézier, os ângulos das juntas que correspondem às trajetórias desejadas são encontrados por meio da aplicação da cinemática inversa ao robô e, por fim, a prototipagem permite a simulação da locomoção do robô bem como a validação da coerência da mesma.

3.1 Modelo matemático do *Central Pattern Generator*

A geração dos padrões de marchas nada mais é do que a criação e repetição de determinada sequência em que as patas de um robô entram em contato com o solo, por isso, o equacionamento de um modelo matemático para o CPG pode ser feito com osciladores. Para o caso de um robô quadrúpede, um oscilador é acoplado a cada pata para que as mesmas possam realizar o movimento almejado enquanto o acoplamento entre os osciladores permite que a marcha desejada seja executada.

O modelo matemático do *Central Pattern Generator* não pode depender das condições iniciais, isto porque, independentemente da posição ou da ordem em que, por exemplo, as patas de um robô quadrúpede começam a se movimentar, a marcha, em regime permanente, deve ser executada corretamente, por isso, o sistema equacionado precisa oscilar conforme uma estrutura não-linear específica denominada ciclo-limite, explicada em detalhes no decorrer desta subseção.

3.1.1 Abordagem geométrica para a resolução de problemas

Em um sistema linear descrito por equações diferenciais, as variáveis dependentes e todas as suas derivadas são do primeiro grau (ZILL; CULLEN, 2008). Tal característica permite a separação de um problema linear em partes menores, permitindo que as mesmas sejam resolvidas de forma analítica separadamente e recombinações em uma solução final (STROGATZ, 2018).

A abordagem analítica nem sempre é possível em um problema não-linear ou mesmo interessante, já que pode não refletir com clareza seu funcionamento. Por isso, interpretar um sistema não-linear geometricamente é a abordagem geralmente adotada, tendo em vista que permite a compreensão deste sem a necessidade de resolvê-lo analiticamente.

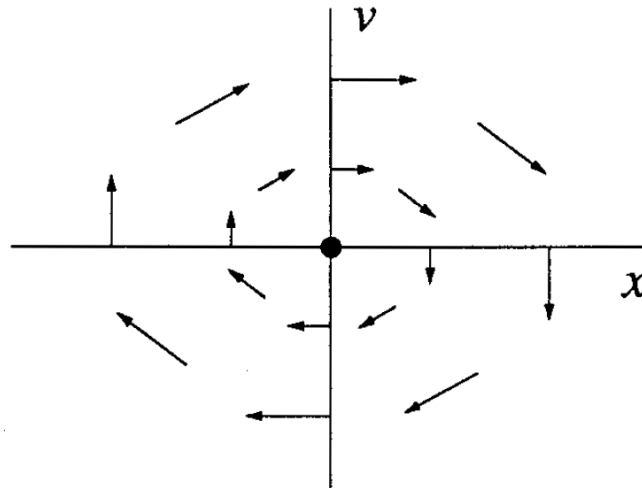
Os recursos utilizados para a interpretação geométrica de determinado problema podem ser compreendidos a partir da análise de um sistema massa-mola ideal (também chamado de oscilador harmônico linear simples) constituído por um corpo de massa m e uma mola com constante elástica k .

O estado do sistema, isto é, o conjunto mínimo de variáveis que o descrevem no instante inicial $t = t_0$ e que determinam o comportamento do mesmo para $t \geq t_0$ (OGATA, 2011), é constituído pelo deslocamento x da mola e pela velocidade v da partícula. O sistema pode ser descrito por meio de tais variáveis de estado conforme expressa (2).

$$\begin{cases} \dot{x} = v \\ \dot{v} = -\frac{k}{m}x \end{cases} \quad (2)$$

Tratando-se as variáveis de estado do oscilador harmônico simples definidas em (2) como eixos de um sistema de coordenadas bidimensional, pode-se associar a cada ponto (x, v) um vetor $(\dot{x}, \dot{v}) = (v, -\frac{k}{m}x)$ (STROGATZ, 2018). Plotando-se alguns vetores no plano, obtém-se o gráfico apresentado na Figura 3.

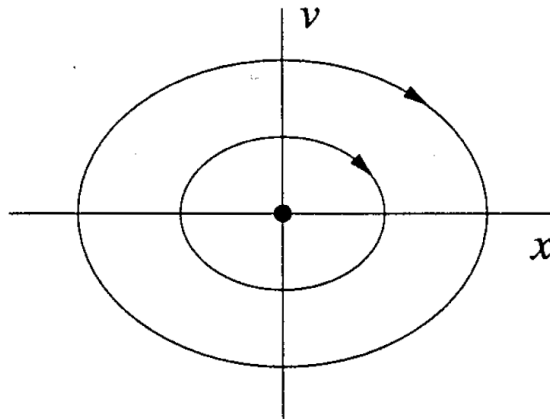
Figura 3 – Campo de vetores (*vector field*) plotado em um plano de fase (*phase plane*)



Fonte: Strogatz (2018).

Cada ponto em que um vetor é associado no gráfico da Figura 3 representa o estado do sistema em determinado instante de tempo e é denominado ponto de fase (*phase point*). Já o sistema de coordenadas bidimensional em que os pontos são plotados é denominado plano de fase (*phase plane*). Nota-se na Figura 3 que o conjunto de vetores plotados, também chamado de campo de vetores (*vector field*) formam trajetórias fechadas, apresentadas de forma explícita na Figura 4, a qual constitui um diagrama ou retrato de fase (*phase diagram* ou *phase portrait*), isto é, o conjunto de todas as trajetórias do sistema plotadas em um plano de fase.

Figura 4 – Trajetórias formadas pelo campo de vetores (*vector field*) em um plano de fase (*phase plane*) constituindo um diagrama ou retrato de fase (*phase diagram* ou *phase portrait*)



Fonte: Strogatz (2018).

Analizando-se apenas o diagrama da Figura 4, pode-se extrair informações importantes pertinentes ao funcionamento do sistema, como o fato de que, para $v = 0$, o deslocamento x assume valores máximos em módulo, e para $x = 0$, a velocidade v assume valores máximos em módulo, indicando que a energia nestes instantes é completamente potencial ou cinética, respectivamente. Percebe-se também que a energia total do sistema se mantém constante, já que as trajetórias são fechadas, indicando que o sistema retorna ao seu estado inicial depois de um ciclo.

Além disso, para valores distintos de x e de v , o sistema apresenta órbitas fechadas distintas, apontando a linearidade do mesmo, uma vez que, se $(\dot{x}, \dot{v}) = (v, -\frac{k}{m}x)$ é uma solução do sistema, todas as combinações lineares da mesma também são, resultando em infinitas trajetórias possíveis. Uma órbita só não é criada em $(x, y) = (0, 0)$, pois este ponto, chamado de ponto de equilíbrio ou ponto fixo, representa o oscilador harmônico parado, sem distensão da mola e sem velocidade, ou seja, em equilíbrio estático.

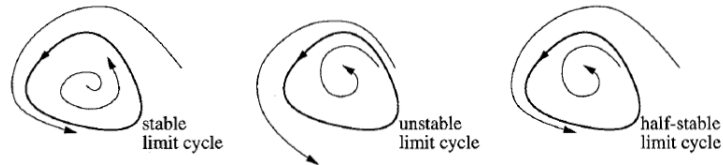
Apesar de ser possível obter um comportamento periódico com o sistema usado como exemplo, a linearidade do mesmo impede que este seja independente das condições iniciais. Por isso, para o atendimento dos requisitos do modelo matemático do CPG, o ciclo-limite precisa ser utilizado.

3.1.2 Ciclo-limite

Ciclo-limite é uma trajetória fechada e isolada que representa a variação periódica do estado de um sistema (STROGATZ, 2018). Enquanto a periodicidade da trajetória é indicada pelo fato da mesma ser fechada, seu isolamento indica que todas as trajetórias suficientemente próximas convergem para o ciclo-limite ou são repelidas do mesmo, isto é, não é possível a

existência de outra trajetória fechada nas redondezas desta estrutura. A Figura 5 demonstra exemplos de ciclos-limite classificados quanto à estabilidade. Se todas as trajetórias vizinhas convergem para o ciclo-limite, o mesmo é estável, caso contrário, se as trajetórias são total ou parcialmente repelidas, o ciclo-limite pode ser classificado como instável ou parcialmente estável, respectivamente (STROGATZ, 2018).

Figura 5 – Exemplos de ciclos-limite estável, instável e parcialmente estável



Fonte: Strogatz (2018).

Ciclos-limite estáveis são de suma importância para a modelagem de sistemas que exibem oscilações auto-sustentadas, como é o caso do CPG. Por isso, o modelo matemático que simula o comportamento de tal estrutura biológica precisa de um tipo específico de artifício matemático capaz de gerar ciclos-limite: a bifurcação de Hopf.

3.1.3 Bifurcação de Hopf

Bifurcação é definida como uma mudança qualitativa na dinâmica de um sistema a partir da variação de determinado parâmetro, alterando, por conseguinte, a estrutura topológica de um *phase diagram*. Bifurcações que geram ciclos-limite ou outras soluções periódicas são denominadas bifurcações de Hopf (JORDAN; SMITH, 2007).

Esta classe de bifurcações surge a partir da desestabilização de um ponto fixo de um sistema com duas variáveis de estado, a qual ocorre por meio da variação de um parâmetro μ até determinado valor crítico (μ_c). Tal fenômeno é verificado fazendo-se a linearização do espaço de estados em torno de um ponto fixo através de uma matriz Jacobiana $\mathbf{J}_{(0,0)}$ (ou Jacobiano), definida por (3), e, posteriormente, analisando-se se a variação do parâmetro contido nos autovalores da matriz Jacobiana, calculados a partir de (4), faz com que os mesmos cruzem o eixo imaginário do plano complexo da esquerda para a direita (STROGATZ, 2018).

$$\mathbf{J}_{(0,0)} = \begin{bmatrix} \frac{\partial \dot{x}}{\partial x} & \frac{\partial \dot{x}}{\partial y} \\ \frac{\partial \dot{y}}{\partial x} & \frac{\partial \dot{y}}{\partial y} \end{bmatrix} \quad (3)$$

$$\det(\mathbf{J}_{(0,0)} - \lambda I) = 0 \quad (4)$$

Na expressão (3), $\dot{x}$ e $\dot{y}$ correspondem às derivadas parciais das variáveis de estado do sistema, enquanto que, na equação (4), λ representa o conjunto de autovalores do Jacobiano.

Como as bifurcações de Hopf não são delimitadas apenas por um conjunto de equações específico, para a modelagem de um sistema que apresente a mudança qualitativa requerida, pode-se utilizar como base os requisitos definidos por Righetti e Ijspeert (2008), que são a independência da frequência com relação à amplitude e o ciclo-limite com amplitude igual a $\sqrt{\mu}$. Com isso, o sistema de equações expresso em coordenadas polares por (5) pode ser considerado.

$$\begin{cases} \dot{r} = r(\mu - r^2) \\ \dot{\theta} = \omega \end{cases} \quad (5)$$

A variável $\dot{r}$ foi expressa conforme (5) porque, quando equivale a zero, pode indicar a localização de pontos fixos ou de ciclos-limite, por isso, foi dimensionada de modo a contemplar um ponto fixo instável na origem do plano de fase ($r = 0$) e um ciclo-limite com amplitude igual a $\sqrt{\mu}$ ($r = \sqrt{\mu}$).

Ademais, a variável θ expressa em (5) é a fase do sistema, ω é a frequência angular e r é definido por $\sqrt{x^2 + y^2}$, em que x e y são variáveis de estado sem significado físico previamente concebido que são definidas em coordenadas polares como segue:

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

Para a aplicação do Jacobiano, deve-se encontrar os valores de $\dot{x}$ e $\dot{y}$, logo, derivando-se (6), obtém-se:

$$\begin{cases} \dot{x} = \dot{r} \cos \theta - r \dot{\theta} \sin \theta \\ \dot{y} = \dot{r} \sin \theta + r \dot{\theta} \cos \theta \end{cases} \quad (7)$$

Substituindo-se (5) em (7), vem:

$$\begin{cases} \dot{x} = (\mu - r^2)r \cos \theta - \omega r \sin \theta \\ \dot{y} = (\mu - r^2)r \sin \theta + \omega r \cos \theta \end{cases} \quad (8)$$

Substituindo-se (6) em (8), tem-se que:

$$\begin{cases} \dot{x} = (\mu - r^2)x - \omega y \\ \dot{y} = (\mu - r^2)y + \omega x \end{cases} \quad (9)$$

Linearizando-se o espaço de estados (9) com base na matriz jacobiana definida em (3),

obtém-se:

$$\mathbf{J}_{(0,0)} = \begin{bmatrix} \frac{\partial[(\mu-r^2)x-\omega y]}{\partial x} & \frac{\partial[(\mu-r^2)x-\omega y]}{\partial y} \\ \frac{\partial[(\mu-r^2)y+\omega x]}{\partial x} & \frac{\partial[(\mu-r^2)y+\omega x]}{\partial y} \end{bmatrix}$$

$$\mathbf{J}_{(0,0)} = \begin{bmatrix} (\mu-r^2) & -\omega \\ \omega & (\mu-r^2) \end{bmatrix} \quad (10)$$

Considerando-se que a linearização foi feita em torno da origem do plano de fase, isto é, onde $(x, y) = (0, 0)$, e que $r^2 = \sqrt{x^2 + y^2}$, vem:

$$\mathbf{J}_{(0,0)} = \begin{bmatrix} \mu & -\omega \\ \omega & \mu \end{bmatrix} \quad (11)$$

Calculando-se os autovalores definidos em (4) a partir de (11), vem:

$$\lambda = \mu \pm i\omega \quad (12)$$

Variando-se μ de valores negativos para positivos nos autovalores encontrados, é possível perceber que a parte real dos mesmos também varia de valores negativos para positivos, cruzando assim o eixo imaginário do plano complexo da esquerda para a direita e, consequentemente, gerando uma bifurcação de Hopf. Logo, um sistema definido pelo espaço de estados (9) e que mantenha μ com valores positivos pode ser classificado como um oscilador de Hopf. Um oscilador semelhante ao equacionado é utilizado por Righetti e Ijspeert (2008) para a modelagem do CPG, contudo, algumas alterações são feitas de modo a atender certas características almeçadas para o modelo matemático.

3.1.4 Osciladores modificados de Hopf

Com o intuito de controlar a convergência das variáveis de estado de forma independente, Righetti e Ijspeert (2008) definiram um oscilador de Hopf considerando as constantes α e β , conforme mostra o espaço de estados (13).

$$\begin{cases} \dot{x} = \alpha(\mu - r^2)x - \omega y \\ \dot{y} = \beta(\mu - r^2)y + \omega x \end{cases} \quad (13)$$

Além disso, como as frequências da *stance phase* (ω_{stance}) e da *swing phase* (ω_{swing}) podem ser diferentes entre si, com base no modelo de Righetti e Ijspeert (2008), Wu *et al.*

(2013) definiu a frequência angular ω como segue;

$$\omega = \frac{\omega_{stance}}{e^{-bx} + 1} + \frac{\omega_{swing}}{e^{bx} + 1} \quad (14)$$

A dependência da frequência ω com relação à fase, isto é, a posição no ciclo-limite em determinado instante de tempo (representada por x neste contexto), caracteriza o oscilador de Hopf como modificado (IJSPEERT, 2008).

Além disso, na equação (14), b representa uma constante que controla a velocidade com que a frequência ω assume os valores de ω_{swing} ou ω_{stance} . Para isto, b precisa assumir um valor positivo relativamente grande com relação aos demais valores da equação, isto porque, considerando-se os resultados conhecidos e expressos por (15) e (16), quando x é menor que zero, a frequência ω deve ser igual a ω_{swing} , correspondendo à *swing phase*, já quando x é maior que zero, ω deve se tornar igual a ω_{stance} , correspondendo à *stance phase*.

$$\lim_{b \rightarrow \infty} e^b = \infty \quad (15)$$

$$\lim_{b \rightarrow \infty} e^{-b} = 0 \quad (16)$$

Como a execução de determinada marcha depende da interação entre os osciladores constituintes do robô que se deseja controlar, Wu *et al.* (2013) adicionou ao oscilador de Hopf modificado apresentado em (13) um termo de acoplamento, o qual representa a diferença de fase com os demais osciladores. O espaço de estados (17) apresenta o oscilador com a modificação efetuada.

$$\begin{cases} \dot{x}_i = \alpha(\mu - r_i^2)x_i - \omega_i y_i \\ \dot{y}_i = \beta(\mu - r_i^2)y_i + \omega_i x_i + k \sum_j \Delta_{ji} \end{cases} \quad (17)$$

Onde i representa o oscilador que está sendo analisado, j representa outro oscilador do robô e k constitui um parâmetro arbitrário que controla a influência que o acoplamento tem em determinado oscilador. O acoplamento Δ_{ji} pode ser calculado conforme expressa a equação (18).

$$\Delta_{ji} = y_j \cos \theta_{ji} - x_j \sin \theta_{ji} \quad (18)$$

Na equação (18), θ_{ji} representa a diferença de fase entre os osciladores i e j e pode ser calculado como segue:

$$\theta_{ji} = 2\pi(\phi_i - \phi_j) \quad (19)$$

Onde ϕ_i e ϕ_j representam as fases de referência dos osciladores i e j .

3.2 Curvas de Bézier

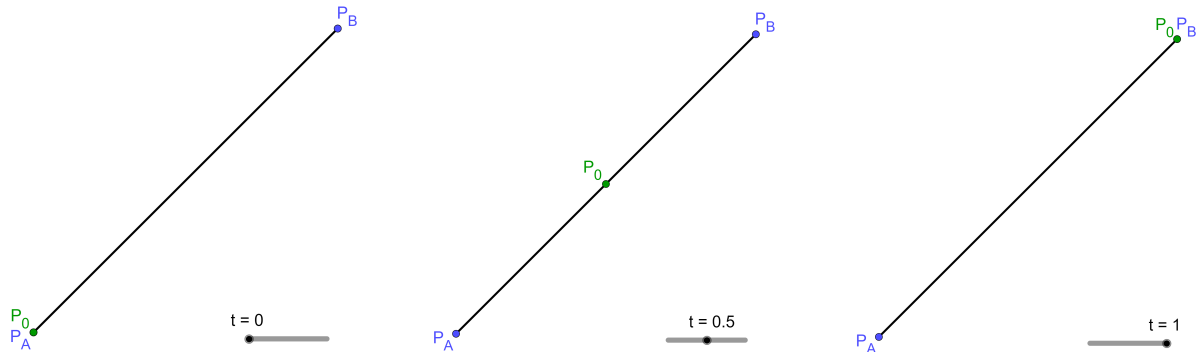
Curvas de Bézier são curvas paramétricas definidas a partir de pontos de controle que permitem uma representação suave, contínua e facilmente ajustável de trajetórias modeladas com base nas variáveis de estado resultantes do modelo matemático do CPG (HYUN *et al.*, 2014). A construção de tais curvas ocorre por meio da aplicação de um método recursivo denominado algoritmo de De Casteljau, o qual pode ser interpretado geometricamente como segue.

Considerando-se os pontos $P_A(x_A, y_A)$ e $P_B(x_B, y_B)$ localizados em um sistema de coordenadas bidimensional, pode-se realizar a interpolação linear entre os mesmos e calcular as coordenadas de um ponto intermediário $P_0(t)$ a partir da equação (20).

$$P_0(t) = (1 - t)P_A + tP_B \quad (20)$$

Analisando-se a equação (20), pode-se perceber que o ponto $P_0(t)$ é parametrizado em função de t , parâmetro que representa a posição do ponto intermediário no segmento de reta e que varia entre 0 e 1, sendo 0 a posição coincidente com a do ponto $P_A(x_A, y_A)$ e 1 a posição coincidente com a do ponto $P_B(x_B, y_B)$, conforme mostra a Figura 6.

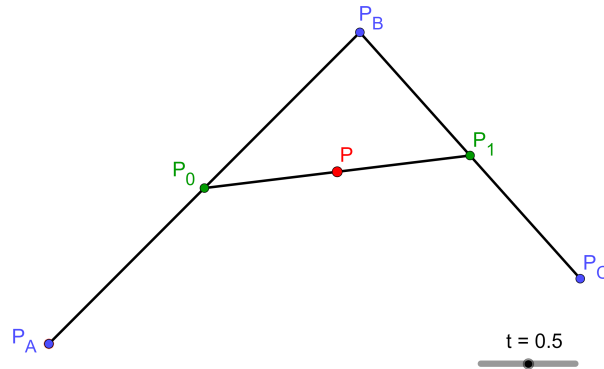
Figura 6 – Ponto $P_0(t)$ parametrizado em função de t e localizado no segmento de reta que interpola os pontos $P_A(x_A, y_A)$ e $P_B(x_B, y_B)$, sendo a posição de $P_0(0)$ correspondente com a do ponto $P_A(x_A, y_A)$ e a posição de $P_0(1)$ correspondente com a do ponto $P_B(x_B, y_B)$



Fonte: Autoria própria.

Adicionando-se um ponto de controle $P_C(x_C, y_C)$ à Figura 6, é possível fazer as interpolações exibidas na Figura 7 e equacionadas em (21).

Figura 7 – Interpolações lineares em um sistema de coordenadas bidimensional criadas a partir dos pontos $P_A(x_A, y_A)$, $P_B(x_B, y_B)$ e $P_C(x_C, y_C)$



Fonte: Autoria própria.

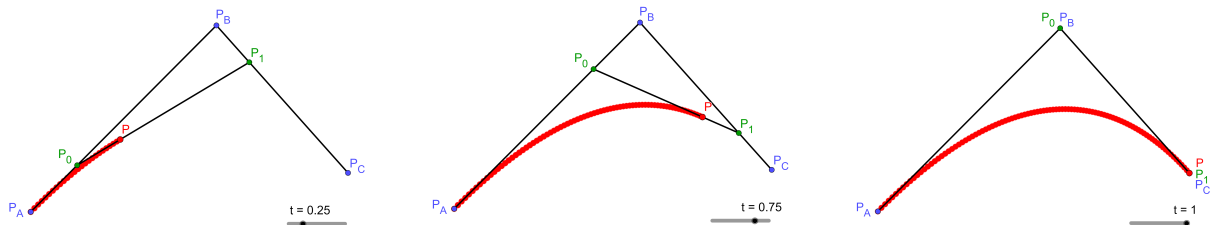
$$\begin{cases} P_0(t) = (1-t)P_A + tP_B \\ P_1(t) = (1-t)P_B + tP_C \\ P(t) = (1-t)P_0 + tP_1 \end{cases} \quad (21)$$

Observa-se que a cada nova interpolação, a anterior é utilizada, caracterizando as etapas definidas por De Casteljaeu como recursivas. Unificando-se as etapas em uma única equação, vem:

$$P(t) = (1-t)^2P_A + 2(1-t)tP_B + t^2P_C \quad (22)$$

Nota-se que todas as interpolações descritas em (21), bem como a equação unificada (22), contam com a parametrização em t , isto é, variando-se t entre 0 e 1, é possível criar uma curva de Bézier, conforme mostra a Figura 8.

Figura 8 – Construção de uma curva de Bézier definida por três pontos de controle a partir da variação do parâmetro t entre 0 e 1



Fonte: Autoria própria.

A morfologia da curva de Bézier pode ser alterada de acordo com a posição dos pontos de controle e plotada a partir de uma quantidade ilimitada de pontos, logo, a sequência de interpolações recursivas que caracterizam o algoritmo de De Casteljaeu e que culminam na equação

(22) pode ser generalizada pela equação (23).

$$P(t) = \sum_{k=1}^n B_{k,n} P_k \quad (23)$$

Onde $P(t)$ representa a equação paramétrica que define a curva de Bézier, n indica o número de pontos de controle, P_k indica o valor do ponto de controle k e o valor de $B_{k,n}(t)$ representa uma estrutura especial conhecida como polinômio de Bernstein, descrito por 24.

$$B_{k,n}(t) = \sum_{k=1}^n \frac{n!}{(n-k)!} (1-t)^{n-k} t^k \quad (24)$$

Logo, a equação que descreve uma curva Bézier de ordem $(n-1)$ definida por n pontos de controle pode ser expressa por (25).

$$P(t) = \sum_{k=1}^n \frac{n!}{(n-k)!} (1-t)^{n-k} t^k P_k \quad (25)$$

3.3 Cinemática inversa de uma cadeia cinemática aberta com dois graus de liberdade (*Degrees of Freedom* ou **DOF**)

A cinemática inversa é o processo de determinação dos ângulos das juntas correspondentes à postura (posição e orientação) de uma cadeia cinemática, definida a partir de um sistema de coordenadas com origem no TCP (*Tool Center Point* ou Ponto Central da Ferramenta) do último *link* do mecanismo (SICILIANO *et al.*, 2010).

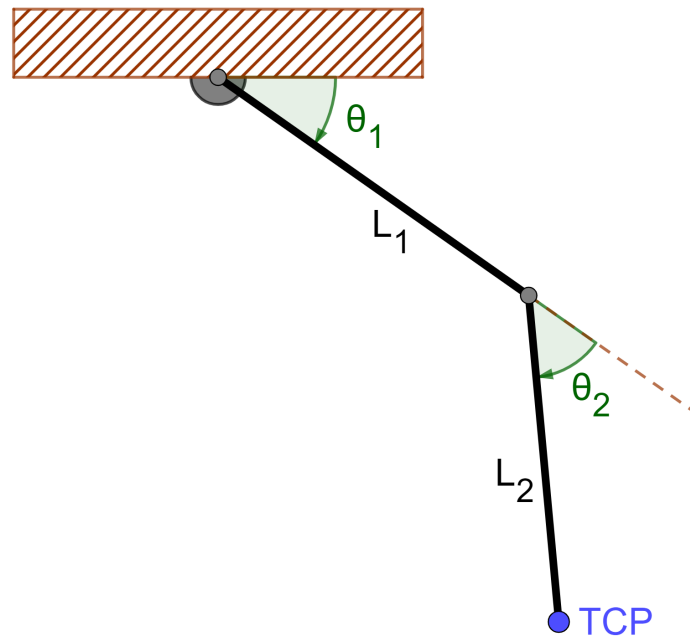
Dentre os vários métodos para a execução da cinemática inversa existentes, o que é descrito a seguir está relacionado à geometria do mecanismo e é aplicado a uma cadeia cinemática aberta com dois graus de liberdade.

A partir da Figura 9, podem ser definidos ângulos e comprimentos estratégicos de modo a viabilizar o equacionamento dos ângulos θ_1 e θ_2 das juntas. É possível equacionar, primeiramente, o ângulo θ_1 a partir das medidas explicitadas na Figura 10.

A partir da geometria da Figura 10, pode-se expressar θ_1 da seguinte maneira:

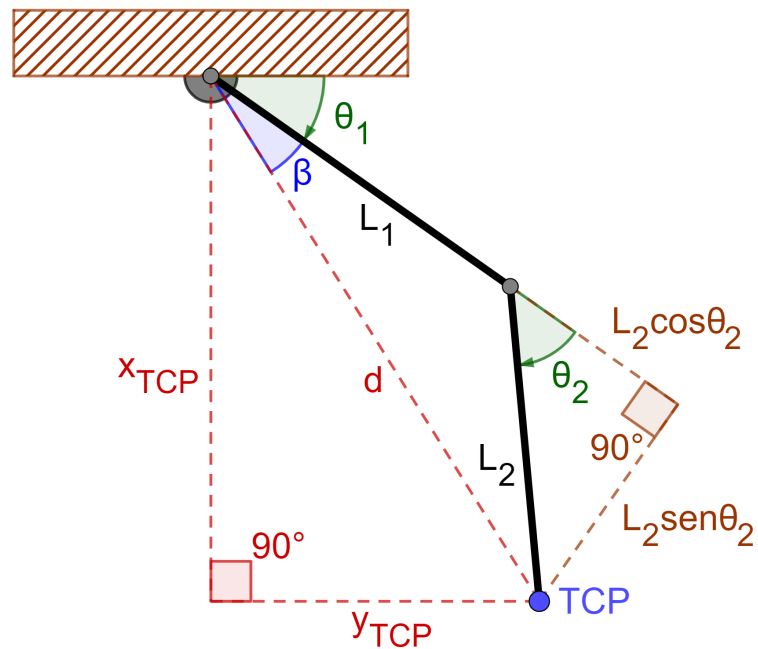
$$\tan(\theta_1 + \beta) = \frac{y_{TCP}}{x_{TCP}} \quad (26)$$

Figura 9 – Cadeia cinemática aberta com dois graus de liberdade e com os ângulos θ_1 e θ_2 destacados



Fonte: Autoria própria.

Figura 10 – Medidas de ângulo e de comprimento feitas na cadeia cinemática aberta com 2 DOF para o equacionamento de θ_1



Fonte: Autoria própria.

Isolando-se θ_1 , vem:

$$\begin{aligned}\tan(\theta_1 + \beta) &= \frac{y_{TCP}}{x_{TCP}} \\ \theta_1 + \beta &= \arctan\left(\frac{y_{TCP}}{x_{TCP}}\right) \\ \theta_1 &= \arctan\left(\frac{y_{TCP}}{x_{TCP}}\right) - \beta\end{aligned}\quad (27)$$

Ainda analisando-se a Figura 10, pode-se expressar β em função dos comprimentos dos *links* da cadeia cinemática, como segue:

$$\tan \beta = \frac{L_2 \sin \theta_2}{L_1 + L_2 \cos \theta_2} \quad (28)$$

Isolando-se β , vem:

$$\beta = \arctan\left(\frac{L_2 \sin \theta_2}{L_1 + L_2 \cos \theta_2}\right) \quad (29)$$

Substituindo-se (29) em (27), obtém-se a equação que descreve θ_1 expressa em (30):

$$\theta_1 = \arctan\left(\frac{y_{TCP}}{x_{TCP}}\right) - \arctan\left(\frac{L_2 \sin \theta_2}{L_1 + L_2 \cos \theta_2}\right) \quad (30)$$

Para o equacionamento de θ_2 , as medições usadas são encontradas na Figura 11.

Pela lei dos cossenos, tem-se que:

$$d^2 = L_1^2 + L_2^2 - 2L_1L_2 \cos \alpha \quad (31)$$

Isolando-se $\cos \alpha$, vem:

$$\cos \alpha = \frac{L_1^2 + L_2^2 - d^2}{2L_1L_2} \quad (32)$$

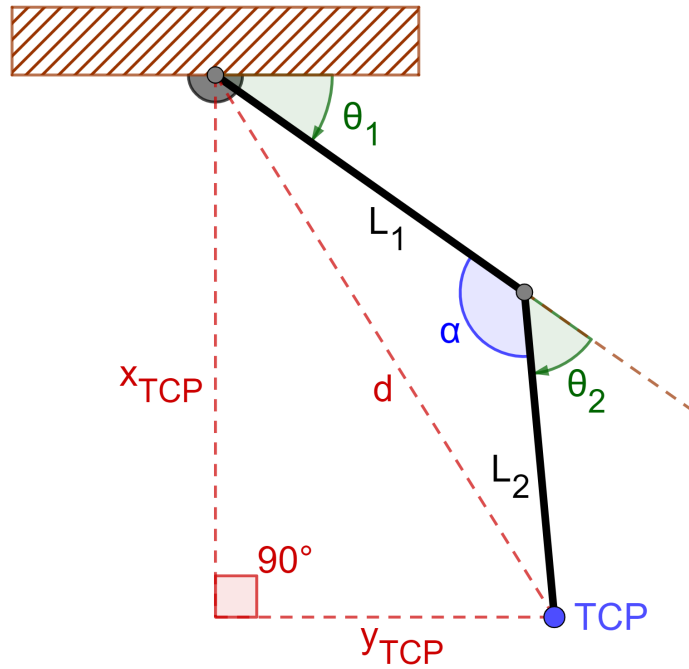
Aplicando-se o teorema de Pitágoras à distância d exibida na Figura 11, tem-se:

$$d^2 = x_{TCP}^2 + y_{TCP}^2 \quad (33)$$

Substituindo-se (33) em (32), obtém-se:

$$\cos \alpha = \frac{L_1^2 + L_2^2 - x_{TCP}^2 - y_{TCP}^2}{2L_1L_2} \quad (34)$$

Figura 11 – Medidas de ângulo e de comprimento feitas na cadeia cinemática aberta com 2 DOF para o equacionamento de θ_2



Fonte: Autoria própria.

Analisando-se a Figura 11, é possível deduzir que θ_2 pode ser expresso por (35).

$$\theta_2 = \pi - \alpha \quad (35)$$

A função cosseno pode ser aplicada aos dois lados da igualdade, como segue:

$$\cos \theta_2 = \cos(\pi - \alpha) \quad (36)$$

Desenvolvendo-se (36) a partir de uma transformação trigonométrica, vem:

$$\cos \theta_2 = \cos(\pi) \cos(\alpha) + \sin(\pi) \sin(\alpha) \quad (37)$$

Substituindo-se $\cos(\pi)$ e $\sin(\pi)$ por seus respectivos valores, obtém-se:

$$\cos \theta_2 = -\cos(\alpha) \quad (38)$$

Substituindo-se (34) em (38), vem:

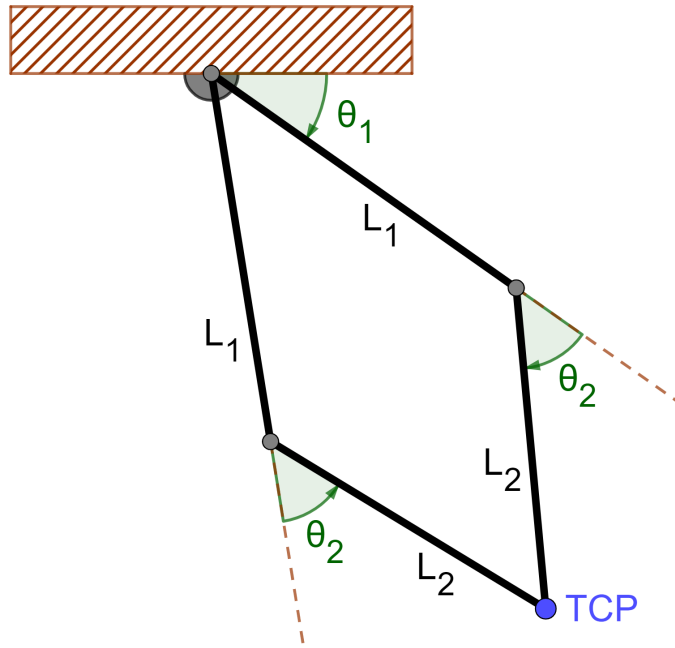
$$\cos \theta_2 = \frac{x_{TCP}^2 + y_{TCP}^2 - L_1^2 - L_2^2}{2L_1L_2} \quad (39)$$

Sabe-se que a função cosseno é par, isto é, $\cos \theta_2$ é igual a $\cos(-\theta_2)$, logo, θ_2 pode ser expresso da seguinte forma:

$$\theta_2 = \pm \arccos\left(\frac{x_{TCP}^2 + y_{TCP}^2 - L_1^2 - L_2^2}{2L_1L_2}\right) \quad (40)$$

As duas soluções para o ângulo θ_2 representam as duas posições que o mecanismo pode assumir para os mesmos dados de posição e orientação do sistema de coordenadas com origem no TCP, conforme explicita a Figura 12.

Figura 12 – Representação das duas posições que a cadeia cinemática aberta de 2 DOF pode assumir para os mesmos dados de posição e de orientação do sistema de coordenadas com origem no TCP



Fonte: Autoria própria.

Com isso, as duas soluções podem ser expressas por (41) e (42), como segue:

$$\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} \arctan\left(\frac{y_{TCP}}{x_{TCP}}\right) - \arctan\left(\frac{L_2 \sin \theta_2}{L_1 + L_2 \cos \theta_2}\right) \\ \arccos\left(\frac{x_{TCP}^2 + y_{TCP}^2 - L_1^2 - L_2^2}{2L_1L_2}\right) \end{bmatrix} \quad (41)$$

$$\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} \arctan\left(\frac{y_{TCP}}{x_{TCP}}\right) - \arctan\left(\frac{L_2 \sin \theta_2}{L_1 + L_2 \cos \theta_2}\right) \\ - \arccos\left(\frac{x_{TCP}^2 + y_{TCP}^2 - L_1^2 - L_2^2}{2L_1L_2}\right) \end{bmatrix} \quad (42)$$

Onde x_{TCP} e y_{TCP} podem ser representados em função dos comprimentos dos *links* L_1 e L_2 por meio das equações (43) e (44), respectivamente.

$$x_{TCP} = L_1 \cos \theta_1 + L_2 \cos (\theta_1 + \theta_2) \quad (43)$$

$$y_{TCP} = L_1 \sin \theta_1 + L_2 \sin (\theta_1 + \theta_2) \quad (44)$$

O ângulo θ_2 é sempre medido com relação ao primeiro *link*, sendo que, se o sentido de rotação com relação a esse referencial for anti-horário, o ângulo é positivo, caso contrário, é negativo. Logo, a solução (41) representa a configuração da cadeia cinemática exibida na Figura 12 com articulação à esquerda, enquanto a solução (42) indica a configuração com a articulação à direita.

3.4 Prototipagem virtual

Um protótipo é a aproximação de um produto, de forma física ou virtual, em uma ou mais características de interesse a fim de testar ideias, compartilhar funcionalidades de forma visual com o time ou com clientes e validar o funcionamento do produto (ULRICH; EPPINGER, 2008).

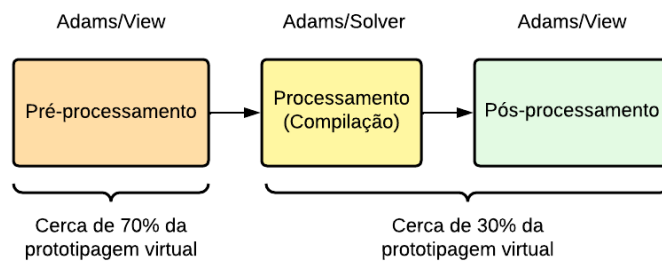
A prototipagem virtual consiste no processo de criação de um protótipo em determinado *software* de engenharia assistido por computador, ou CAE (*Computer Aided-Engineering*), e oferece maior flexibilidade para ajustes, rapidez na confecção e redução de custos quando comparada à prototipagem física (MÜLLER; SAFFARO, 2011). Tal processo não se limita apenas ao uso do *software* e pode ser dividido em três etapas:

- **Pré-processamento:** constitui a maior parte da prototipagem virtual, englobando modelagem 3D, importação do modelo 3D em software CAE, interação com elementos externos e determinação de materiais, massas, fixação e afins (FIGUEIREDO, 2021).
- **Processamento:** compilação do protótipo pelo computador e geração de arquivos com dados que serão analisados na etapa de pós-processamento (FIGUEIREDO, 2021).
- **Pós-processamento:** etapa de testes para validar o funcionamento do protótipo com base nos parâmetros almejados.

Dentre os diversos softwares CAE existentes, para este trabalho, optou-se por utilizar o Adams (acrônimo para *Automatic Dynamic Analysis of Mechanical Systems* ou Análise Dinâmica Automática para Sistemas Mecânicos, em tradução livre), software desenvolvido pela Hexagon que oferece simplicidade de uso e é difundido nos meios industrial e acadêmico (LI; KOTA, 2001).

O Adams possui diversos módulos que viabilizam aplicações específicas como o Adams/View, usado para criação, montagem e configuração de modelos de sistemas mecânicos, o Adams/Solver, utilizado para a realização de cálculos para a resolução de equações de movimento de sistemas mecânicos e o Adams/Controls, que permite análises de desempenho e simulações de sistemas de controle em ambientes dinâmicos. A correspondência entre as etapas da prototipagem virtual e os módulos do Adams que podem ser utilizados para executá-las é apresentada na Figura 13.

Figura 13 – Correspondência entre as etapas da prototipagem virtual e os módulos do Adams que podem ser utilizados para executá-las



Fonte: Autoria própria.

4 METODOLOGIA

Nesta seção são descritas as etapas de prototipagem virtual de um robô quadrúpede no Adams versão 2014, implementação do processo de geração de trajetórias cinemáticas no software MATLAB versão 2022 e a cosimulação entre o Adams e o Simulink, integrando as duas últimas etapas.

4.1 Prototipagem virtual de um robô quadrúpede no Adams

O *design*, as dimensões e as propriedades de massa do protótipo virtual do robô quadrúpede desenvolvido no Adams/View foram baseados nos dados definidos por Katz (2018) e por Katz, Carlo e Kim (2019) para o *Mini Cheetah*, tendo em vista que o mesmo constitui um robô quadrúpede leve, pequeno, robusto e de baixo custo construído justamente para prover maior facilidade e segurança nos testes feitos em inventos similares no MIT (Massachusetts Institute of Technology) (KATZ; CARLO; KIM, 2019).

Os procedimentos descritos nesta subseção fazem parte do pré-processamento da prototipagem virtual e podem divididos nas seguintes etapas:

- Importação e manipulação do desenho tridimensional de um robô quadrúpede;
- Inserção de pontos de construção para definição das posições das juntas e validação dos comprimentos das patas do robô;
- Definição das propriedades de massa do robô quadrúpede;
- Inserção de juntas no robô quadrúpede;
- Criação de contato entre o robô quadrúpede e o solo.

4.1.1 Importação e manipulação do desenho tridimensional de um robô quadrúpede

O modelo tridimensional do robô quadrúpede foi retirado da plataforma *GrabCAD*, comunidade *online* de compartilhamento de desenhos em 3D (GRABCAD, 2017). O *design* de tal modelo, por sua vez, foi construído com base no robô quadrúpede *Mini Cheetah*, conforme mostra a Figura 14.

O *download* do arquivo foi feito no formato *Parasolid* e importado neste mesmo formato no módulo Adams/View. O robô importado apresentava 338 corpos (ou *links*), contudo, este número foi reduzido para 25 removendo-se os componentes internos das peças, como parafusos e enrolamentos dos motores, de forma a manter o quadrúpede apenas com sua carcaça externa.

Figura 14 – Modelo tridimensional original do *Mini Cheetah*, à esquerda, em comparação com o desenho do robô quadrúpede retirado do *GrabCAD*, à direita

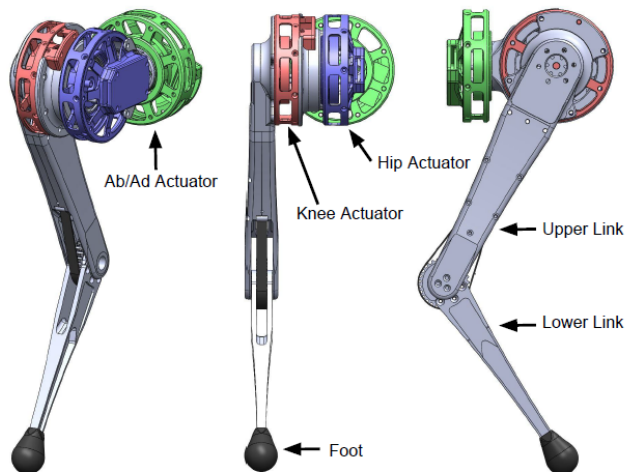


Fonte: Adaptado de Katz (2018).

Este procedimento foi realizado para simplificar o protótipo, já que a simulação do mesmo não depende das peças excluídas para funcionar e sim de ferramentas inerentes ao próprio *software*.

As partes mantidas foram agrupadas de modo a delimitar o corpo central do robô, onde bateria e componentes eletrônicos são armazenados no *Mini Cheetah*, e as quatro patas (ou pernas) contendo segmentos específicos, conforme ilustra a Figura 15.

Figura 15 – Partes constituintes de cada uma das patas do *Mini Cheetah*



Fonte: Adaptado de Katz (2018).

4.1.2 Inserção de pontos de construção para definição das posições das juntas e validação dos comprimentos das patas do robô

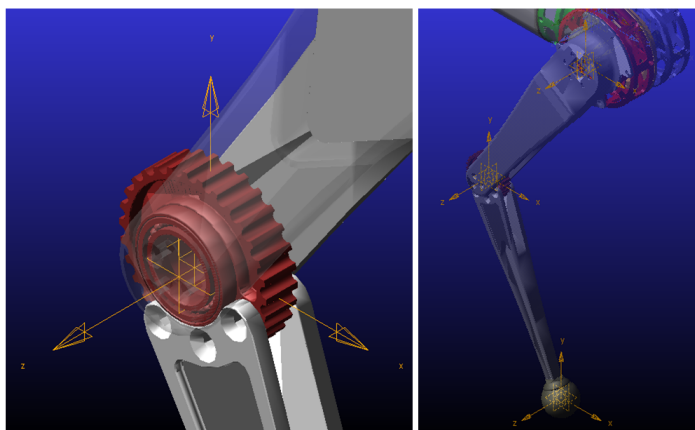
Pontos de construção, no Adams, são entidades geométricas que podem ser associadas ao solo (*ground*) ou aos *links*, servindo como referência para medições e para a inserção de elementos como corpos ou juntas. A construção de projetos com base em pontos deste tipo fornece facilidade de manipulação de parâmetros, já que a localização dos sistemas referenci-

ais (*markers*) associados a tais entidades pode ser alterada apenas com a modificação de suas coordenadas.

As entidades geométricas supracitadas foram postas em posições estratégicas do robô para a inserção de juntas e a realização de medidas. No corpo central do robô, que é simétrico, o ponto de construção foi posicionado em seu centro de massa, servindo de referência para o posicionamento de pontos coplanares nos motores de abdução e adução (*ab/ad actuators*) e motores de quadril (*hip actuators*).

Já para cada um dos *links* inferiores (*lower links*), inseriu-se um ponto de construção no centro de massa da engrenagem simétrica constituinte do *lower link* (e que intersecta o *link* superior) e adicionaram-se pontos coplanares ao mencionado associados com o motor de joelho (*knee actuator*) e o pé de borracha (*foot*), conforme mostra a Figura 16.

Figura 16 – Ponto de construção (em conjunto com um *marker*) inserido no centro de massa da engrenagem simétrica constituinte do *lower link* (e que intersecta o *link* superior), à esquerda, e pontos coplanares (em conjunto com *markers*) associados ao motor de joelho (*knee actuator*) e ao pé de borracha (*foot*), à direita



Fonte: Autoria própria.

Com base nos pontos de construção adicionados, os comprimentos dos *links* inferior e superior das patas foram medidos e comparados com as dimensões correspondentes do *Mini Cheetah* (KATZ, 2018). Como todas as patas do robô quadrúpede do MIT têm dimensões iguais, as medições feitas no modelo tridimensional do Adams foram comparadas apenas com os comprimentos de uma das patas do *Mini Cheetah*, conforme exhibe a Tabela 2.

A similaridade entre as dimensões dos *links* do *Mini Cheetah* e do modelo 3D do Adams permite que outras características do robô desenvolvido no MIT sejam utilizadas para a construção de um protótipo virtual verossímil, como suas propriedades de massa.

Tabela 2 – Comparação entre os comprimentos das pernas do modelo tridimensional de robô quadrúpede do Adams e de uma das patas do *Mini Cheetah*

Perna do robô	Comprimento do link superior (m)	Comprimento do link inferior (m)
Perna do <i>Mini Cheetah</i>	0,21	0,18
Perna dianteira esquerda (LF)	0,22	0,21
Perna dianteira direita (RF)	0,22	0,21
Perna traseira direita (RH)	0,22	0,21
Perna traseira esquerda (LH)	0,22	0,21

Fonte: Autoria própria.

4.1.3 Definição das propriedades de massa e inserção de juntas motorizadas no robô quadrúpede

As massas dos elementos exibidos na Figura 15 foram estipuladas a partir dos dados publicados por Katz (2018) e por Katz, Carlo e Kim (2019). Contudo, no Adams/View, a definição de determinada massa m ocorre de forma indireta a partir da inserção de um valor de densidade ρ , calculado por meio da equação (45).

$$\rho = \frac{m}{v} \quad (45)$$

A variável v representa o volume, o qual pode ser obtido pelo *software* CAE acessando-se as propriedades de cada corpo no Adams/View.

A partir dos valores das massas e dos volumes dos corpos, foi possível obter os valores de densidade, conforme mostra a Tabela 3.

Tabela 3 – Valores de massa, volume e densidade de cada elemento constituinte do modelo tridimensional do robô quadrúpede baseado no *Mini Cheetah*

Elemento do robô	Massa m (kg)	Volume v (mm ³)	Densidade ρ (kg/mm ³)
<i>Abduction/Adduction (Ab/Ad) actuator</i>	0,500	2,32 x 10 ⁴	2,16 x 10 ⁻⁵
<i>Hip actuator</i>	0,500	5,03 x 10 ⁴	9,94 x 10 ⁻⁶
<i>Knee actuator</i>	0,500	5,20 x 10 ⁴	9,62 x 10 ⁻⁶
<i>Upper link</i>	0,092	1,62 x 10 ⁵	5,67 x 10 ⁻⁷
<i>Lower link</i>	0,055	3,34 x 10 ⁴	1,65 x 10 ⁻⁶
<i>Foot</i>	0,025	1,50 x 10 ⁴	1,67 x 10 ⁻⁶
<i>Robot's body</i>	2,312	3,09 x 10 ⁵	7,48 x 10 ⁻⁶

Fonte: Autoria própria.

A massa de cada motor utilizado no *Mini Cheetah* é igual a 0,440 kg (KATZ; CARLO; KIM, 2019), contudo, para englobar as massas dos suportes dos motores e de periféricos removidos, considerou-se arbitrariamente uma massa de 0,500 kg para cada um deles.

Além das massas dos motores, as dos *links* inferior e superior também foram fornecidas por Katz, Carlo e Kim (2019), no entanto, estes dados não foram explicitados para os pés de borracha (*feet*) e para o corpo central do robô (*robot's body*), levando à necessidade de defini-los indiretamente.

Sabendo-se que os pés de borracha do *Mini Cheetah* foram feitos com bolas de *squash* adaptadas com aproximadamente 40 mm de diâmetro (KATZ, 2018), considerou-se o valor de 0,025 kg para a massa de cada um deles com base em um exemplar do artigo esportivo mencionado (GMBH, 2021).

Somando-se as massas de 12 motores, 4 *links* superiores, 4 *links* inferiores e 4 pés de borracha, obtém-se o valor de 6,688 kg. Subtraindo-se este valor da massa total do robô, a qual deve ser igual a do *Mini Cheetah* (equivalente a 9,000 kg) (KATZ; CARLO; KIM, 2019), tem-se a massa do corpo central do robô, igual a 2,312 kg.

Muitos dos elementos supracitados não têm movimento relativo entre si, como ocorre entre o *link* inferior e o pé de borracha, e teriam que ser interligados por juntas fixas. Como uma simulação com diversas juntas fixas pode suscitar problemas de convergência (HONGTAO, 2009), determinados corpos foram unidos no Adams a partir do comando *Merge Bodies* e seus valores de densidade foram modificadas de modo a manter a massa total do robô igual a 9,000 kg. Os novos valores de massa, volume e densidade dos elementos são exibidos na Tabela 4.

Tabela 4 – Valores de massa, volume e densidade dos elementos agrupados no modelo tridimensional do robô quadrúpede baseado no *Mini Cheetah*

Elemento do robô	Massa m (kg)	Volume v (mm³)	Densidade δ (kg/mm³)
<i>Robot's body + 4 Ad/Ab actuators</i>	4,312	$4,02 \times 10^5$	$1,07 \times 10^{-5}$
<i>Hip actuator</i>	0,500	$5,03 \times 10^4$	$9,94 \times 10^{-6}$
<i>Knee actuator + Upper link</i>	0,592	$2,14 \times 10^5$	$2,77 \times 10^{-6}$
<i>Lower link + Foot</i>	0,080	$4,84 \times 10^4$	$1,65 \times 10^{-6}$

Fonte: Autoria própria.

4.1.4 Inserção de juntas no robô quadrúpede

Todos os elementos listados na Tabela 4 foram interligados entre si por meio de juntas de revolução, detalhadas na Tabela 5. Cada junta adicionada teve seu eixo de rotação associado a um ponto de construção e com direção paralela a um dos eixos do sistema referencial global do Adams.

As três juntas listadas na Tabela 5 foram inseridas em cada perna do robô, totalizando doze juntas de revolução no modelo tridimensional. Ademais, as mesmas foram transformadas em juntas motorizadas a partir da atribuição do recurso *Rotational Joint Motion* do Adams/View

a cada uma delas. Tal recurso permite a rotação de juntas de revolução a partir de determinado tipo de entrada (deslocamento, velocidade ou aceleração).

Pode-se fazer uma associação do elemento *Rotational Joint Motion* com um servomotor: apesar da possibilidade de se configurar o tipo de entrada de tal dispositivo como posição, velocidade ou corrente, internamente, há conversão do dado de entrada em grandezas como tensão e torque de modo a produzir o movimento almejado. O mesmo ocorre com o *Rotational Joint Motion*, já que o dado de entrada é convertido em grandezas que possibilitem ao *software* interpretar e executar o deslocamento angular requerido para a junta.

Pretende-se que a junta realize um deslocamento angular ao longo do tempo, contudo, para uma configuração de deslocamento para o dado de entrada em um elemento *Rotational Joint Motion*, o Adams não limita a velocidade com que a junta muda de uma posição à outra, com isso, se um corpo associado a tal junta entrar em contato com outro, pode haver problemas de convergência devido à força excessiva resultante de uma variação angular brusca em um intervalo infinitesimal de tempo. Por isso, configurou-se o tipo de entrada de cada *Rotational Joint Motion* criado no robô quadrúpede como sendo igual à velocidade.

Ademais, restringiu-se o movimento do robô de modo que cada perna apresentasse dois graus de liberdade a partir da inserção de uma entrada de velocidade nula em cada uma das juntas 0, especificadas na Tabela 5. Tal restrição impede o controle de postura do corpo central do robô, por isso uma junta planar foi adicionada no centro de massa de tal corpo, limitando o movimento do robô ao plano sagital. A Figura 17 apresenta as juntas e os *Rotational Joint Motions* destacados no modelo tridimensional do robô quadrúpede.

Tabela 5 – Juntas de revolução definidas para o modelo tridimensional de robô quadrúpede no Adams

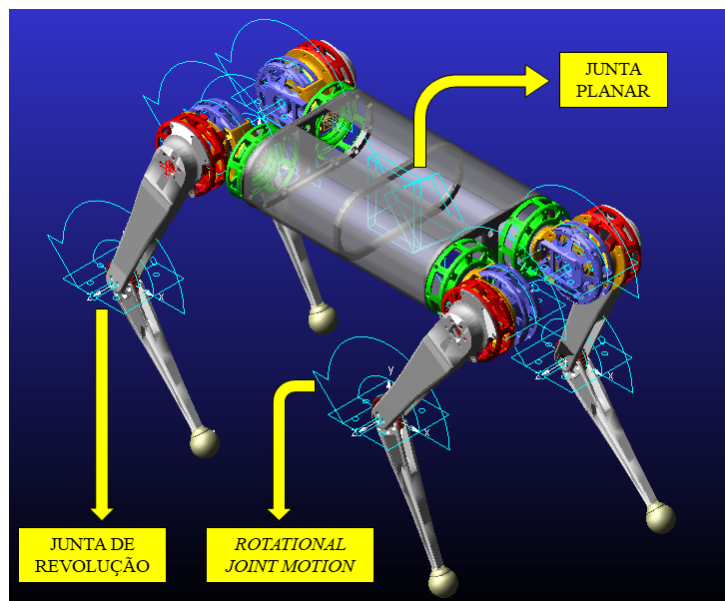
Junta de revolução	Primeiro corpo interligado	Segundo corpo interligado
Junta 0	<i>Robot's body + 4 Ad/Ab actuators</i>	<i>Hip actuator</i>
Junta 1	<i>Hip actuator</i>	<i>Knee actuator + Upper link</i>
Junta 2	<i>Knee actuator + Upper link</i>	<i>Lower link + Foot</i>

Fonte: Autoria própria.

4.1.5 Criação de contato entre o robô quadrúpede e o solo

A utilização de contatos provê ao protótipo um comportamento mais fidedigno ao modelo real de um robô quadrúpede, já que possibilita a análise de sua interação com outros corpos. Para a prototipagem virtual em questão, a interação deve ocorrer entre os pés de borracha e um substrato, os quais, para fins de simplificação da simulação, foram aproximados por esferas e um plano, respectivamente.

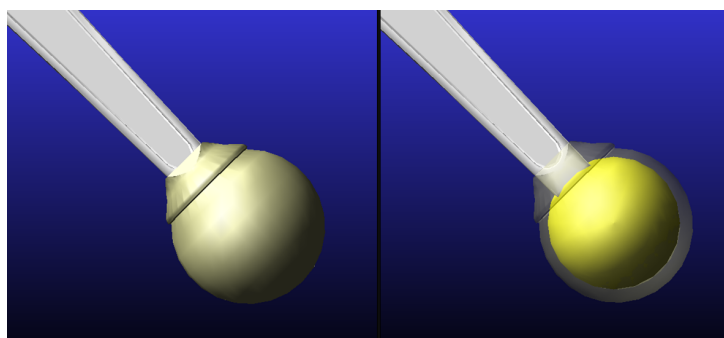
Figura 17 – Modelo tridimensional do robô quadrúpede com junta planar e elementos das juntas motorizadas ativas (juntas de revolução e *Rotational Joint Motions*) destacados



Fonte: Autoria própria.

Como a estrutura externa de cada pé de borracha apresenta uma geometria complexa para a simulação de um contato, criou-se uma esfera de 30 mm de diâmetro internamente a cada um deles de modo que a mesma ficasse encoberta pela estrutura externa do pé de borracha, conforme ilustra a Figura 18.

Figura 18 – Estrutura externa do pé de borracha do modelo de robô quadrúpede do Adams/View, à esquerda, e esfera criada para contato com o solo encoberta pela estrutura externa do pé de borracha, à direita



Fonte: Autoria própria.

As esferas criadas foram associadas aos *links* inferiores por meio do comando *Merge Bodies*, levando à necessidade de se calcular novamente a densidade ρ definida em (45) com o intuito de manter a massa de 0,080 kg para cada conjunto composto por *lower link* e *foot*, considerando que o volume do mesmo mudou com a associação efetuada. Os novos valores definidos para a estrutura são apresentados na Tabela 6.

Tabela 6 – Novos valores de massa, volume e densidade definidos para a estrutura composta por *lower link* e *foot* do modelo tridimensional de robô quadrúpede baseado no *Mini Cheetah*

Elemento do robô	Massa m (kg)	Volume v (mm ³)	Densidade δ (kg/mm ³)
<i>Lower link + Foot</i>	0,080	$6,47 \times 10^4$	$1,24 \times 10^{-6}$

Fonte: Autoria própria.

O plano com o qual os pés de borracha deve entrar em contato é uma entidade geométrica bidimensional desprovida de massa, logo, para a sua associação a um modelo tridimensional, criou-se um sólido retangular, fixou-se o mesmo ao solo por meio de uma junta fixa e associou-se o plano a sua superfície superior.

Entre todas as pernas do robô e o plano, foram criados contatos do tipo *sphere to plane*, nos quais os seguintes parâmetros foram definidos com base no que é explicitado por Hexagon (2021):

- ***Stiffness***: rigidez do material utilizada para o cálculo da força normal no impacto;
- ***Force exponent***: parâmetro que auxilia no cálculo da força normal no impacto;
- ***Damping***: amortecimento do contato definido, geralmente, como sendo equivalente a 1% do coeficiente de rigidez (*stiffness*);
- ***Penetration depth***: penetração em que o amortecimento (*damping*) é máximo;
- ***Static and dynamic coefficients***: coeficientes de atrito estático e dinâmico, respectivamente;
- ***Static transition velocity***: velocidade em que o coeficiente de atrito estático é máximo;
- ***Friction transition velocity***: velocidade em que o valor do coeficiente de atrito se torna igual ao dinâmico (após uma transição gradual do coeficiente estático para o dinâmico).

A direção da força normal do plano com relação aos pés do robô foi configurada de modo a propulsioná-los para frente. Ademais, a adequação das especificações supracitadas e listadas na Tabela 7, as quais foram fornecidas pelo Adams como padrão, foi testada da seguinte forma: simulou-se por um intervalo de tempo total de 1,00 s e com um *step size* de 0,01 s a queda do robô até o alcance da plataforma retangular, de forma a verificar se as condições definidas tornaram o contato muito rígido, levando à projeção do robô a alturas elevadas após o impacto, ou muito suavizado, levando a um primeiro impacto sem oscilações. Como nenhum dos comportamentos extremos foi observado durante o teste, optou-se por manter as condições definidas na Tabela 7.

Tabela 7 – Juntas de revolução definidas para o modelo tridimensional do robô quadrúpede no Adams

Parâmetros da força de impacto	
<i>Stiffness</i>	$1,0 \times 10^5$ N/mm
<i>Damping</i>	10,0 Ns/mm
<i>Exponent</i>	1,2
<i>Dmax</i>	$1,0 \times 10^{-2}$ mm
Atrito de <i>Coulomb</i>	
<i>Static coefficient:</i>	0,9
<i>Dynamic coefficient:</i>	0,8
<i>Stiction transition velocity</i>	100,0 mm/s
<i>Friction transition velocity</i>	1000,0 mm/s

Fonte: Autoria própria.

4.2 Implementação do processo de geração de trajetórias cinemáticas no MATLAB

A implementação do processo de geração de trajetórias cinemáticas consiste na implantação do modelo matemático do *Central Pattern Generator*, criação das curvas de Bézier e resolução da cinemática inversa do robô quadrúpede. Tal procedimento foi executado no MATLAB, software interativo de alta performance amplamente utilizado para a resolução de equações e para a simulação de modelos matemáticos, conforme é descrito nesta subseção.

4.2.1 Implementação do modelo matemático do *Central Pattern Generator* no MATLAB

O modelo matemático do CPG foi implantado no MATLAB com quatro osciladores associados às quatro pernas do robô e expressos pelo sistema de equações (17).

Os valores para os parâmetros de convergência α e β , o parâmetro b de controle da velocidade de transição entre um tipo de frequência e outro (ω_{stance} e ω_{swing}) e a amplitude $\sqrt{\mu}$ foram definidos com base nos valores apresentados por Righetti e Ijspeert (2008), já a constante k foi definida com valor unitário de modo a não influir no aumento ou na diminuição da robustez do acoplamento entre os osciladores. Os parâmetros definidos são exibidos na Tabela 8.

Tabela 8 – Valores dos parâmetros utilizados para a implementação do modelo matemático do *Central Pattern Generator* no MATLAB

Parâmetro do CPG	Valor utilizado
α	5
β	50
b	1000
k	1
$\sqrt{\mu}$	1

Fonte: Autoria própria.

Configurou-se o CPG para a geração da marcha simétrica trote com *duty factor* (β) igual

a 0,50 e diferenças de fase listadas na Tabela 9.

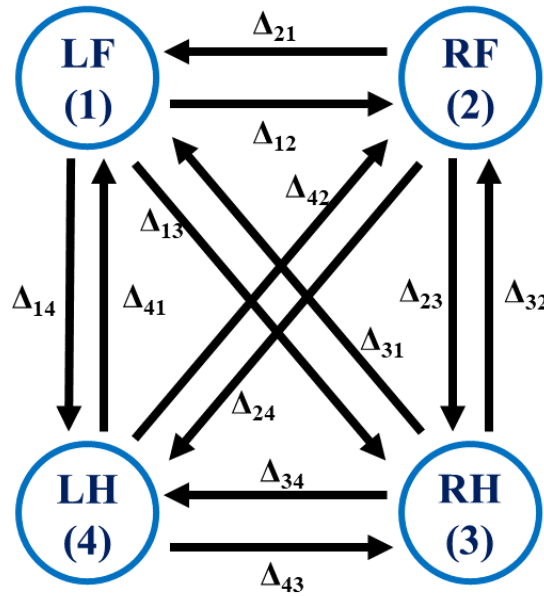
Tabela 9 – Valores referentes às diferenças de fases entre os osciladores (θ_{ji}) do protótipo virtual do robô quadrúpede para a marcha trote (*trot gait*)

Diferença de fase (θ_{ji})	θ_{21}	θ_{31}	θ_{41}	θ_{12}	θ_{32}	θ_{42}	θ_{13}	θ_{23}	θ_{43}	θ_{14}	θ_{24}	θ_{34}
Trot gait	$-\pi$	0	$-\pi$	π	π	0	0	$-\pi$	$-\pi$	π	0	π

Fonte: Autoria própria.

Os osciladores 1, 2, 3 e 4 são associados às pernas LF, RF, RH e LH, respectivamente, conforme ilustra a Figura 19.

Figura 19 – Correspondência entre os osciladores 1, 2, 3 e 4 e as pernas LF, RF, RH e LH associadas



Fonte: Autoria própria.

A resolução das equações pertinentes aos quatro osciladores fornece valores de $\dot{x}$ e de $\dot{y}$, contudo, para se obter valores de x e de y , implementou-se a função *ode45* do MATLAB, que utiliza, por sua vez, o método numérico de *Runge-Kutta* de 4ª ordem, ou método de *Dormand-Prince*, com base em um passo de integração variável (MATHWORKS, 2023a).

4.2.2 Implementação das curvas de Bézier no MATLAB

Para cada perna, foi criada uma curva de Bézier constituída por outras duas criadas separadamente com base na equação (25): uma para a *stance phase* e outra para a *swing phase*.

Os pontos de controle necessários para a criação de cada tipo de curva foram definidos a fim de cumprir os seguintes requisitos: retração suficiente da perna do robô para evitar obstáculos, mas não excessiva de modo a produzir perdas de energia significativas no contato com o

solo, e respeito aos limites físicos do espaço de trabalho de cada pata para a realização da *swing phase* (HYUN *et al.*, 2014).

Com base em tais requisitos, os pontos definidos por PAULA (2021) em um sistema de coordenadas cartesiano em milímetros e apresentados nas tabelas 46 e 47 foram implementados no MATLAB para a construção das curvas referentes à *stance phase* e *swing phase*, respectivamente.

$$P_{st} = \begin{bmatrix} 200,0 & -400,4 \\ 200,0 & -400,4 \\ 200,0 & -400,4 \\ 0,0 & (1-\delta)(-400,4) \\ -200,0 & -400,4 \\ -200,0 & -400,4 \\ -200,0 & -400,4 \end{bmatrix} S_c \quad (46)$$

$$P_{sw} = \begin{bmatrix} -200,0 & -400,4 \\ -200,0 & -400,4 \\ -200,0 & -400,4 \\ -280,5 & -400,4 \\ -300,0 & -261,5 \\ -300,0 & -261,5 \\ -300,0 & -261,5 \\ 0,0 & -261,5 \\ 0,0 & -261,5 \\ 0,0 & -221,8 \\ 303,2 & -221,8 \\ 303,2 & -221,8 \\ 282,6 & -400,4 \\ 200,0 & -400,4 \\ 200,0 & -400,4 \\ 200,0 & -400,4 \end{bmatrix} S_c \quad (47)$$

Onde δ define a profundidade de penetração e S_c representa um fator para o controle de escala (PAULA, 2021). Para a aplicação em questão, definiu-se δ como sendo igual a zero e o fator S_c equivalente a 0,5, permitindo a adequação da curva de Bézier ao espaço de trabalho de cada pata do robô.

4.2.3 Implementação da cinemática inversa no MATLAB

De acordo com o que expõe a seção 3, existem duas soluções resultantes da aplicação da cinemática inversa em uma cadeia cinemática com dois graus de liberdade, as quais podem ser expressas por (41) e por (42) e são correspondentes às diferentes configurações que podem ser assumidas pelo mecanismo. Contudo, no MATLAB, apenas a solução (41) foi implementada, já que é a mais apropriada à disposição inicial do robô quadrúpede (correspondente à configuração apresentada na Figura 12 com a articulação à esquerda).

Todo o processo de geração de trajetórias que culmina na obtenção dos ângulos das juntas por meio da cinemática inversa ocorreu de forma *offline*, isto é, os ângulos θ_1 e θ_2 obtidos para cada perna do robô foram armazenados individualmente em oito arquivos *MAT-file* (extensão ".mat"), permitindo assim a utilização dos dados obtidos em outras aplicações no MATLAB (MATHWORKS, 2023c).

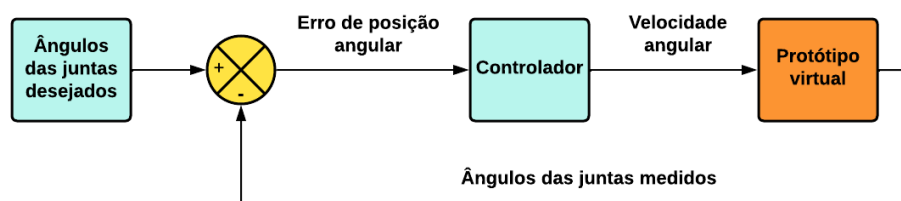
4.3 Cosimulação entre Adams e Simulink

Os ângulos θ_1 e θ_2 das juntas motorizadas do robô quadrúpede, resultantes do processo de geração de trajetórias cinemáticas implementado no MATLAB, e o protótipo virtual desenvolvido no Adams/View foram integrados por meio de uma cosimulação desenvolvida a partir do Simulink, ferramenta de modelagem, simulação e análise de sistemas dinâmicos integrante do *software* MATLAB (MATHWORKS, 2023b). Tal integração foi feita por meio da implementação de um controle cinemático, isto é, um tipo de técnica de controle em que as entradas do sistema são grandezas cinemáticas, como posição ou velocidade (SICILIANO *et al.*, 2010).

Para a aplicação em questão, o controle cinemático foi concebido de modo que os valores dos ângulos θ_1 e θ_2 das pernas do robô, obtidos de forma *offline* e contidos em arquivos *MAT-file*, fossem comparados com os ângulos correspondentes medidos no protótipo virtual, resultando em um erro de deslocamento angular. Tal erro foi submetido a um controlador responsável por fornecer dados de velocidade angular como dados de entrada para as juntas motorizadas do protótipo virtual do Adams/View. A inserção feita motiva um deslocamento angular das patas do robô, o qual é medido e comparado com os ângulos desejados fornecidos pelo MATLAB, repetindo o ciclo.

Com base na forma em que foi implementado, o controle em questão pode ser caracterizado como controle cinemático de posição com acionamento por velocidade e sua estrutura é apresentada na Figura 20. Ademais, as configurações necessárias para que o protótipo virtual do Adams se comporte como um bloco com entradas e saídas na malha de controle bem como a definição do tipo de controlador a ser implementado são descritas nos tópicos subsequentes.

Figura 20 – Malha de controle cinemático de posição com acionamento por velocidade que integra os *softwares* Adams, MATLAB e Simulink na cosimulação



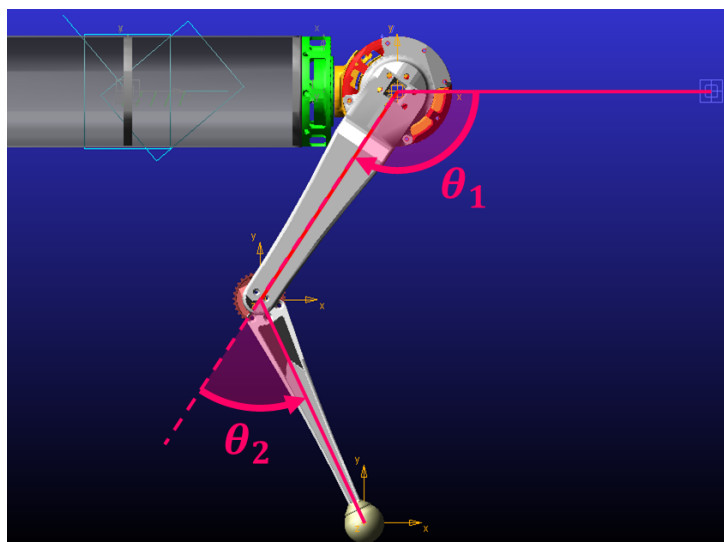
Fonte: Autoria própria.

4.3.1 Integração do protótipo do Adams à malha de controle cinemático

O protótipo virtual foi representado como um bloco na malha de controle cinemático projetada, sendo suas entradas definidas como velocidades angulares das juntas motorizadas e as saídas definidas como deslocamentos angulares das mesmas.

O tipo de entrada das juntas do robô quadrúpede já foi definido como velocidade, conforme explicita a subseção 4.1, já os ângulos de saída θ_1 e θ_2 foram definidos como sendo aqueles medidos entre o *hip actuator* e o *knee actuator/upper link* e entre o corpo *knee actuator/upper link* e o corpo *lower link/foot*, respectivamente. A disposição de tais ângulos é análoga àquela apresentada na subseção 3.3 e pode ser visualizada na Figura 21.

Figura 21 – Ângulos θ_1 e θ_2 medidos no protótipo virtual construído no Adams e utilizados na malha de controle cinemático do Simulink



Fonte: Autoria própria.

A aferição do ângulo θ_1 foi feita entre três pontos de construção coplanares: um primeiro ponto fixo ao *hip actuator*, um ponto central fixo ao *knee actuator* e um ponto final fixo ao *upper link*. Já a medição do ângulo θ_2 foi feita de forma indireta a partir de seu ângulo suple-

mentar $(\pi - \theta_2)$ delimitado pelos seguintes pontos de construção: um primeiro ponto fixo ao *hip actuator*, um ponto central fixo ao *knee actuator* e um ponto final fixo ao *upper link*. Para que as saídas fornecidas à malha de controle do Simulink fossem iguais a θ_2 e não a $(\pi - \theta_2)$, criaram-se funções no Adams/View configuradas com a subtração $[\pi - (\pi - \theta_2)]$.

Com o intuito de armazenar os dados de entrada (velocidade angular) e os de saída (deslocamento angular), foram criadas dezesseis variáveis de estado no Adams/View. Metade das variáveis foram destinadas aos dados de velocidade angular, renomeadas com o prefixo *INPUT* e associadas às entradas das juntas por meio do comando *VARVAL(variável)*, o qual retorna o valor da variável de estado em questão (HEXAGON, 2021).

A outra metade das variáveis foram destinadas aos dados de deslocamento angular (ângulos θ_1 e θ_2 das quatro pernas do robô) e renomeadas com o prefixo *OUTPUT*, sendo que associaram-se as medições de θ_1 das patas do robô a quatro variáveis da quantia que representa o deslocamento angular e as funções que retornam valores de θ_2 às outras quatro.

A comunicação do Adams com o Simulink se dá efetivamente por meio do módulo específico para análises de controle, o Adams/Controls. Para ativá-lo, basta acessar o *Plugin Manager*, localizado em *Tools*, e clicar na opção que o habilita. Com isso, as oito variáveis de estado utilizadas como entradas e as oito usadas como saídas foram exportadas em uma única planta de controle configurada com os parâmetros apresentados na Figura 22.

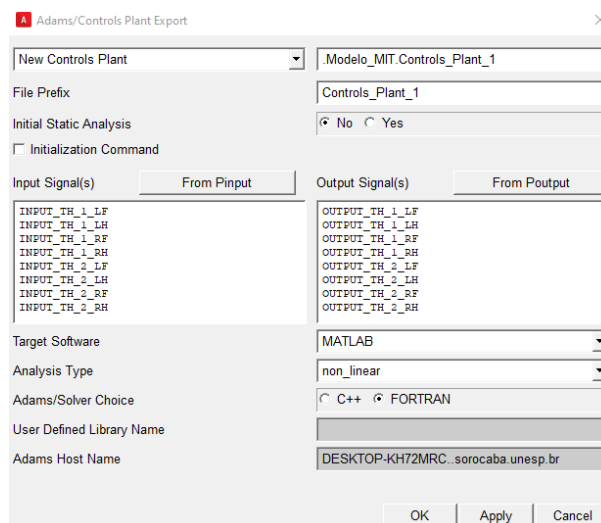
A exportação da planta de controle tem como consequência a geração de vários arquivos que são utilizados direta ou indiretamente na cosimulação, dentre os quais está um arquivo do tipo *MATLAB Code* (extensão ".m") contendo a planta de controle, que pode ser usado no MATLAB para a configuração do protótipo virtual como um bloco na malha de controle construída no Simulink.

4.3.2 Configuração da malha de controle cinemático de posição com acionamento por velocidade no Simulink

Selecionou-se como a pasta de trabalho no MATLAB aquela onde foram exportados os arquivos pertinentes à planta de controle do Adams, importou-se o arquivo do tipo *MATLAB Code* com a planta de controle e executou-o na *Command Window* do MATLAB.

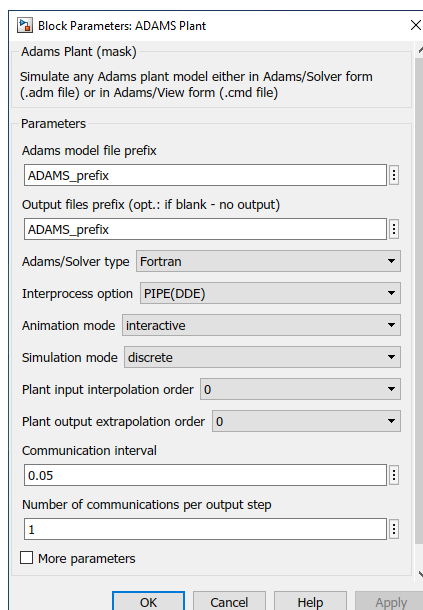
Com isso, ainda na *Command Window*, inseriu-se o comando *adams_sys*, levando à geração automática de um bloco representativo do protótipo virtual do Adams, denominado automaticamente como *adams_sub* em um arquivo do tipo *Simulink Model* (extensão ".slx"). Tal bloco foi copiado e colado em um arquivo do Simulink à parte. Clicando-se duas vezes no *adams_sub* e, posteriormente, no bloco *MSC Software* (em uma janela aberta automaticamente), foram configurados os parâmetros da planta conforme apresenta a Figura 23.

Figura 22 – Parâmetros definidos para a exportação das oito variáveis de entrada (*INPUT*) e das oito variáveis de saída (*OUTPUT*) em uma única planta de controle por meio do Adams/Controls



Fonte: Autoria própria.

Figura 23 – Parâmetros definidos para a configuração do bloco representativo do protótipo virtual no Simulink



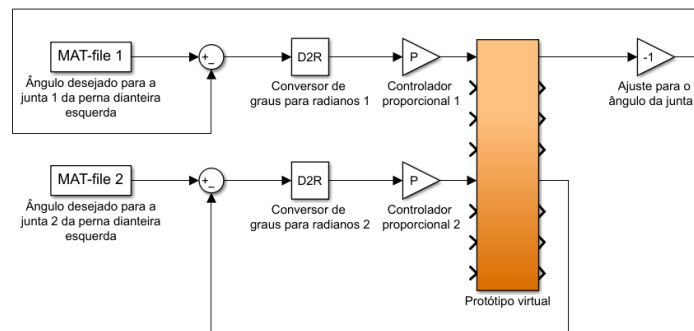
Fonte: Autoria própria.

Retornou-se à janela inicial do arquivo ".slx", onde inseriram-se oito elementos *From File* do Simulink que foram associados aos oito arquivos contendo ângulos das juntas fornecidos de forma *offline* pelo processo de geração de trajetórias cinemáticas implementado no MATLAB e descrito na subseção 4.2. Tal processo foi feito de modo que cada elemento *From File* fosse associado a um único *MAT-file* contendo valores para apenas um tipo de ângulo (θ_1 ou θ_2) de um única perna do robô.

A obtenção dos ângulos desejados das juntas motorizadas bem como o desenvolvimento de um protótipo virtual são de suma importância para a construção de um ambiente que permita a implementação de técnicas de controle, contudo, a validação de ambos é condição indispensável para que isto ocorra. Por isso, além de permitir que a cosimulação ocorra, a integração das duas partes por meio de um controlador viabiliza a validação supracitada, já que permite a interação entre dados das duas implementações e a consequente análise da coerência da mesma. Tendo isto em vista, optou-se por utilizar um controlador proporcional, já que o mesmo possibilita uma integração simples, provendo facilidade em futuras análises.

O controle cinemático de posição com acionamento por velocidade implementado no Simulink com um controlador proporcional é exibido na Figura 24 para apenas uma das pernas do robô, no entanto, o controle associado às juntas foi feito de forma independente, isto é, um controlador foi implementado para cada junta motorizada.

Figura 24 – Controle cinemático de posição com acionamento por velocidade implementado no Simulink para a perna dianteira esquerda (LF) do robô quadrúpede



Fonte: Autoria própria.

Como o processamento do Adams é feito com ângulos em radianos, observa-se na Figura 24 que foram adicionados elementos do Simulink para a conversão do erro de posição angular de graus para radianos.

Além disso, conforme explicado na seção 3.3, se o sentido da rotação da junta que interliga um *link* anterior e um posterior ocorre no sentido anti-horário, o ângulo é positivo, caso contrário, é negativo. Contudo, tal convenção não é adotada pelo Adams, logo, considerando a disposição dos *links* e juntas exibida na Figura 21, é necessário multiplicar o ângulo θ_1 medido por -1 para a correta comparação com o ângulo θ_1 desejado, conforme mostra a Figura 24.

5 RESULTADOS E DISCUSSÕES

Nesta seção são apresentados e discutidos os resultados da implementação do modelo matemático do *Central Pattern Generator*, das curvas de Bézier e da cinemática inversa tanto de forma isolada, no MATLAB, quanto no contexto geral da cosimulação, a partir da análise do comportamento, da estabilidade na execução da marcha e dos torques nas juntas do protótipo virtual desenvolvido no Adams.

5.1 Resultados da implementação do modelo matemático do CPG no MATLAB

Com o intuito de evitar elevados impactos entre as patas do robô e o solo, os períodos da *stance phase* (T_{st}) e da *swing phase* (T_{sw}) foram definidos como sendo iguais a 1,0 s, considerando o *duty factor* (β) equivalente a 0,5. Ademais, as condições iniciais para as variáveis de estado foram definidas conforme a Tabela 10.

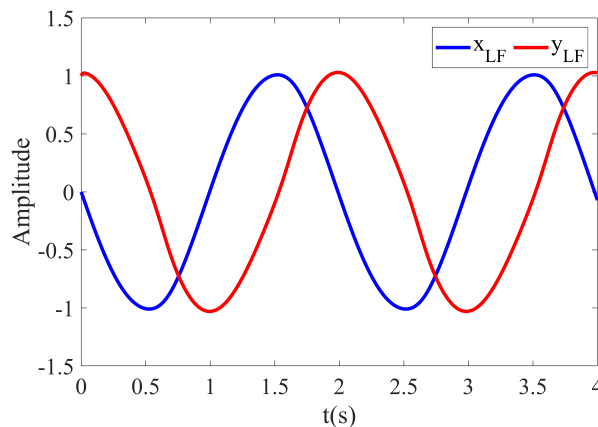
Tabela 10 – Condições iniciais definidas para as variáveis de estado constituintes do modelo matemático do CPG implementado no MATLAB

Variável de estado	x_{LF}	y_{LF}	x_{RF}	y_{RF}	x_{RH}	y_{RH}	x_{LH}	y_{LH}
Condição inicial	0	1	0	-1	0	1	0	-1

Fonte: Autoria própria.

Com os períodos T_{st} e T_{sw} supracitados, as condições da Tabela 10 e os parâmetros definidos na subseção 4.2, foram obtidos valores para as variáveis de estado x e y de cada um dos osciladores associados às pernas do robô quadrúpede. As variáveis x_{LF} e y_{LF} referentes ao oscilador 1, associado à perna dianteira esquerda (LF), foram plotadas ao longo do tempo no gráfico exibido na Figura 25.

Figura 25 – Variáveis de estado x_{LF} e y_{LF} do oscilador 1, associado à perna dianteira esquerda (LF) do robô quadrúpede, plotadas ao longo do tempo

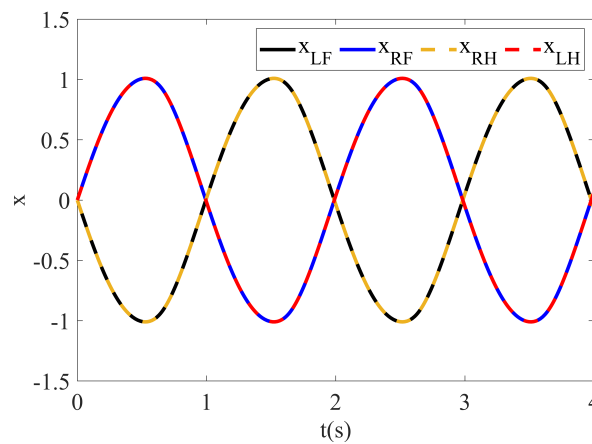


Fonte: Autoria própria.

Analisando-se o gráfico da Figura 25, percebe-se que tanto a amplitude da variável x_{LF} quanto a de y_{LF} iniciam-se nas condições iniciais definidas na Tabela 10 e que a amplitude máxima observada para ambas as curvas é unitária. Além disso, a *swing phase* e a *stance phase*, que são delimitadas por $x_{LF} < 0$ e $x_{LF} > 0$, respectivamente, ocorrem em intervalos de tempo correspondentes a 1,0 s, estando em conformidade com a definição feita no início da presente subseção.

Comparando-se a variável de estado x_{LF} do oscilador 1 (perna LF) com as correspondentes dos osciladores associados às demais patas, obtém-se o gráfico da Figura 26.

Figura 26 – Variáveis de estado x_{LF} , x_{RF} , x_{RH} e x_{LH} plotadas ao longo do tempo com condições iniciais nulas



Fonte: Autoria própria.

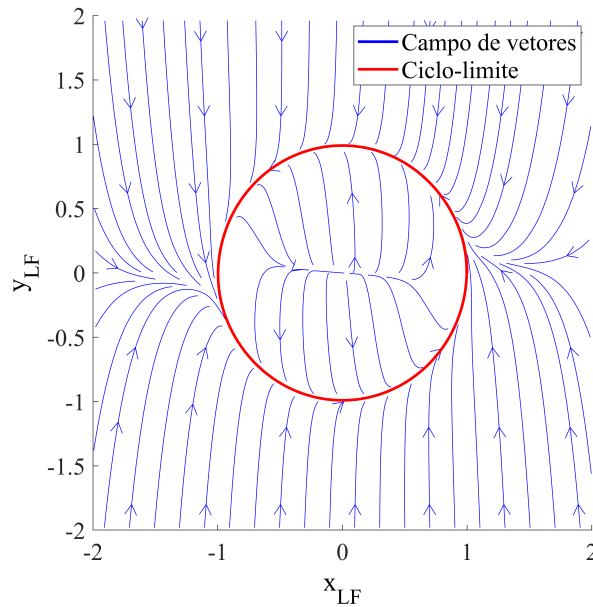
Tanto a superposição entre as curvas dos osciladores de pernas diagonalmente opostas (LF/RH e RF/LH) quanto a defasagem entre as curvas dos osciladores de pernas sagitalmente opostas (LF/RF e RH/LH), apresentadas na Figura 26, caracterizam a marcha trote, haja vista que, quando um par de pernas diagonais está em contato com o solo (*stance phase*), o outro se encontra em *swing phase*, ou seja, pernas opostas diagonalmente devem estar em fase e pernas opostas sagitalmente devem estar defasadas de 180° .

Utilizando-se o oscilador equacionado para a perna LF, plotou-se um diagrama de fase com base nos valores de diversos pontos do plano de fase, inclusive os utilizados como condição inicial para o oscilador de LF ($x_{LF} = 0$ e $y_{LF} = 1$), conforme mostra a Figura 27.

Observa-se na Figura 27 que todos os vetores originários dos pontos de fase do gráfico convergem para uma trajetória circular fechada e isolada que constitui o ciclo-limite, o qual possui raio unitário, consistente com a amplitude $\sqrt{\mu}$ igual a 1 definida na seção 4.2.

Nota-se que as condições iniciais utilizadas para as pernas no robô (definidas na Tabela 10), são pontos que já fazem parte do ciclo-limite, levando à adequação dos valores das variáveis de estado aos parâmetros almejados da marcha trote ($\beta = 0,5$ e $T_{st} = T_{sw} = 1,0$ s) desde o instante inicial de simulação.

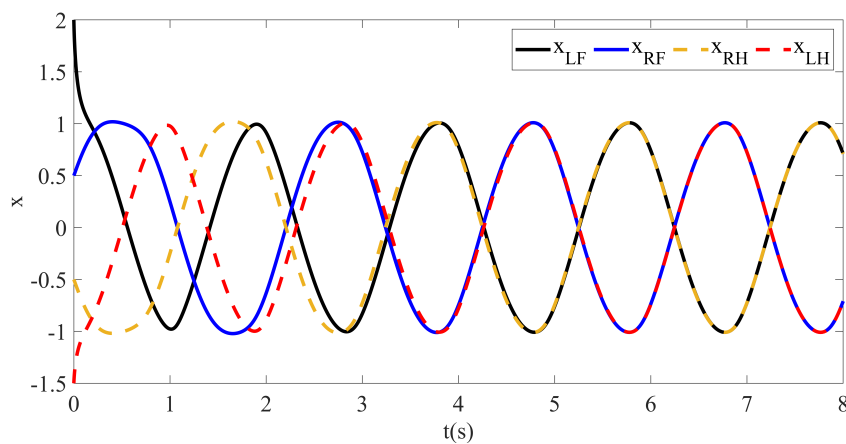
Figura 27 – Diagrama de fase do oscilador 1, associado à perna LF do robô quadrúpede, contendo ciclo-limite de amplitude $\sqrt{\mu}$ unitária



Fonte: Autoria própria.

A definição das variáveis com condições iniciais diferentes das coordenadas do ciclo-limite, mas suficientemente próximas, incutiria em um período transitório até a convergência das trajetórias e a consequente estabilização. Contudo, em tal regime transitório, a marcha executada pelo robô seria descaracterizada, já que as amplitudes, os períodos e as diferenças de fase entre as pernas sofreriam variações, conforme explicita o gráfico das variáveis x_{LF} , x_{RF} , x_{RH} e x_{LH} dos osciladores com relação ao tempo, apresentado na Figura 28.

Figura 28 – Variáveis de estado x_{LF} , x_{RF} , x_{RH} e x_{LH} plotadas ao longo do tempo com condições iniciais iguais a 2,0, 0,5, -0,5 e -1,5, respectivamente



Fonte: Autoria própria.

Observa-se no gráfico apresentado na Figura 28 que condições iniciais fora do ciclo-limite suscitam regimes transitórios com duração proporcional à distância da trajetória circular isolada

e fechada, ou seja, quanto maior é a diferença entre a condição inicial do oscilador e a amplitude do ciclo-limite, maior é o tempo até a estabilização, por isso a curva de x_{LF} é a que mais demora para se estabilizar (levando aproximadamente 5,5 s para tal fim enquanto as demais levaram até 4,5 s para se estabilizar) e, também, a que sofre maiores deformações em sua amplitude e em seu período.

Outrossim, nota-se que as diferenças de fase, principalmente nos três primeiros segundos, encontram-se muito discrepantes daquelas almejadas para a marcha trote e explicitadas anteriormente nesta subseção, já que os acoplamentos entre os osciladores também sofreram o impacto das condições iniciais definidas.

Os gráficos temporais e ciclos-limite individuais dos osciladores 2, 3 e 4 não foram exibidos nesta subseção porque as análises realizadas anteriormente também são válidas para os mesmos, contudo, tais resultados podem ser consultados nos apêndices A e B deste relatório, respectivamente.

5.2 Resultados da implementação das curvas de Bézier no MATLAB

Os resultados obtidos para as variáveis de estado x e y dos quatro osciladores implementados foram utilizados para a parametrização das curvas de Bézier das pernas correspondentes. Para cada perna do robô, foi projetada uma curva de Bézier fechada com duas partes: uma curva feita com os pontos de controle definidos para a *stance phase* (fase de contato com o solo) e outra feita com os pontos definidos para a *swing phase* (fase de balanço). Os valores de x foram usados para a seleção de qual das curvas de Bézier seria implementada em cada perna a cada instante de tempo: para $x < 0$, a curva referente à *swing phase* foi implementada, enquanto para $x > 0$, a curva implementada foi a relativa à *stance phase*.

Já os valores de y foram utilizados para a definição do parâmetro t contido no polinômio (25), expresso na subseção 3.2. Contudo, y pode assumir valores entre -1 e 1, de acordo com a amplitude $\sqrt{\mu}$ unitária definida para o modelo matemático do CPG, enquanto t só pode assumir valores entre 0 e 1, conforme explicita a subseção 3.2. Por isso, as escalas envolvidas no cômputo do parâmetro t foram relacionadas conforme expressam as equações (48) e (49) para a *stance phase* e para a *swing phase*, respectivamente.

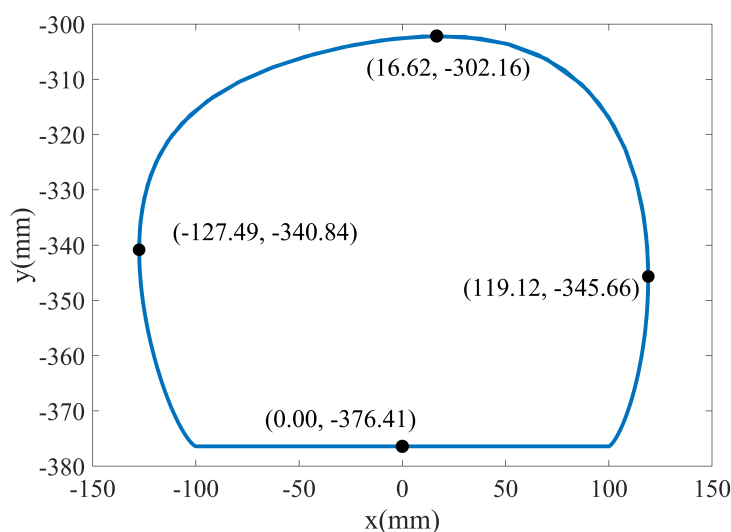
$$t = \frac{1 + y}{2} \quad (48)$$

$$t = \frac{1 - y}{2} \quad (49)$$

Com base no polinômio (25) característico da curva de Bézier, na seleção entre as curvas feita a partir dos valores de x e na parametrização realizada em função dos valores da variável

de estado y , foi possível construir curvas de Bézier para as quatro pernas do robô. Contudo, para a execução das trajetórias cinemáticas a uma altura coerente, de forma que as pernas não ficassem esticadas ou articuladas demais durante a *stance phase*, deslocaram-se as curvas de Bézier linearmente em um sistema de coordenadas cartesianas bidimensional subtraindo das ordenadas das mesmas um valor equivalente a 176,21 mm. A curva de Bézier referente à perna LF e plotada em um sistema de coordenadas cartesianas pode ser visualizada na Figura 29, enquanto as curvas relativas às demais pernas podem ser consultadas no apêndice C.

Figura 29 – Curva de Bézier projetada para a perna LF, com pontos extremos destacados, plotada com base nas variáveis de estado resultantes do oscilador 1 em um sistema de coordenadas cartesianas



Fonte: Autoria própria.

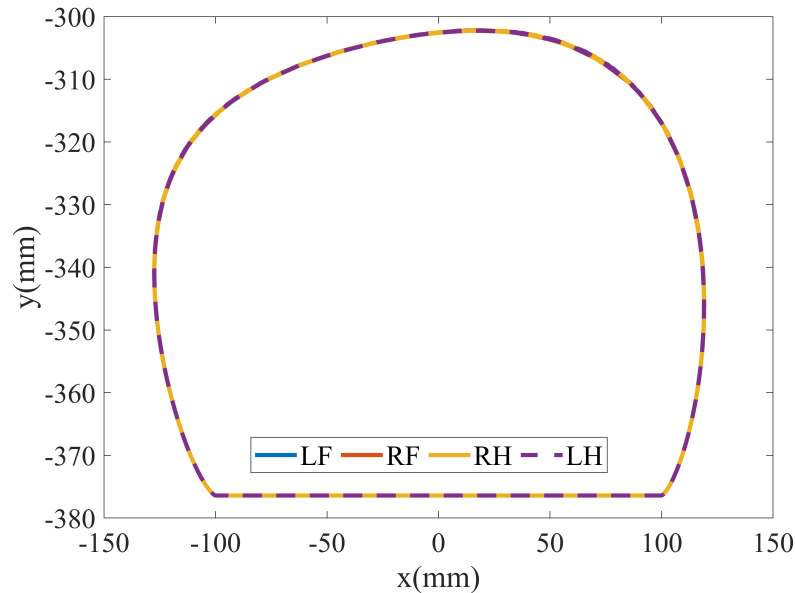
Pode-se perceber que a curva de Bézier apresentada na Figura 29 é fechada, contudo, conforme supraindicado, é composta por duas partes: a curva referente à *stance phase* corresponde àquela com pontos de ordenada igual a -376,41 mm, enquanto a correspondente à *swing phase* engloba todos os demais pontos.

Outrossim, considerando-se que o *links* superior e inferior de cada pata do robô quadrúpede modelado no Adams têm comprimentos iguais a 217,5 mm (0,22 m) e 214,6 mm (0,21 m), pode-se notar a partir dos pontos extremos plotados na Figura 29 que a curva de Bézier construída não ultrapassa os limites físicos do robô, estando compreendida dentro da área de trabalho da pata LF. Ademais, é possível perceber que a retração projetada de 74,25 mm (diferença entre os pontos com ordenadas -302,16 mm e -376,41 mm) é pequena quando comparada com a distância vertical entre o centro de massa do corpo central do robô e o solo (igual a 376,41 mm), contudo, isto não traz empecilhos à locomoção do quadrúpede devido à ausência de obstáculos na cosimulação realizada bem como o intuito de se evitar elevados impactos com o solo (que poderiam ser ocasionados com o aumento da distância vertical entre os pés de borracha e o solo).

O mesmo tipo de curva de Bézier, com amplitude em x próxima de 246 mm e amplitude

em y próxima de 74 mm, foi projetada para cada uma das pernas do robô quadrúpede, conforme mostra a comparação feita na Figura 30, onde a superposição das curvas implementadas no MATLAB ressalta a execução de movimentos idênticos entre si.

Figura 30 – Curvas de Bézier projetadas para as quatro pernas (LF, RF, RH e LH) do robô quadrúpede e implementadas no MATLAB



Fonte: Autoria própria.

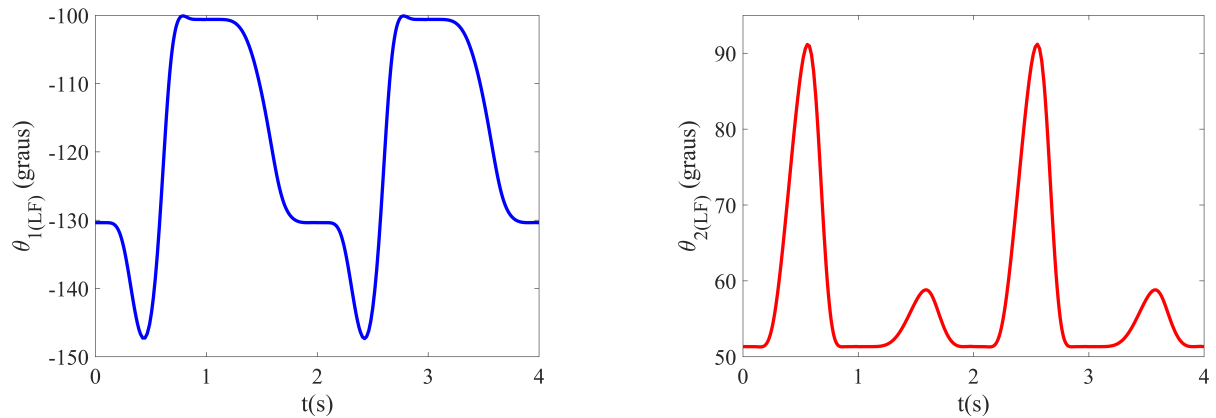
5.3 Resultados da implementação da cinemática inversa no MATLAB

As coordenadas dos pontos que constituem as curvas de Bézier foram utilizadas na implementação da cinemática inversa feita no MATLAB como sendo os valores para as coordenadas x_{TCP} e y_{TCP} , que definem a posição do TCP de cada perna do robô quadrúpede, conforme explicita a equação (41) apresentada na seção 3.3.

Com base na trajetória projetada e na escolha da solução apropriada à condição inicial do robô, foram obtidos os gráficos dos ângulos θ_1 e θ_2 das juntas do robô quadrúpede ao longo do tempo para a perna LF, conforme apresenta a Figura 31.

Analisando-se o gráfico temporal de θ_1 , apresentado na Figura 31, observa-se que no intervalo de tempo compreendido entre 1 s e 2 s, correspondente à *stance phase* da perna LF nas condições iniciais definidas na Tabela 10, o ângulo θ_1 assume valores cada vez maiores em módulo, saindo de -100° e chegando a -130° , o que, no contexto do protótipo virtual, representa o deslocamento angular no sentido anti-horário do *upper link* com relação ao *hip actuator*, que resulta na translação realizada pelo pé de borracha da direita para a esquerda durante a *stance phase*, correspondente à trajetória com ordenada igual -376,41 mm apresentada na curva de Bézier da Figura 29.

Figura 31 – Ângulos θ_1 e θ_2 das juntas do robô quadrúpede calculados para a perna LF e plotados ao longo do tempo



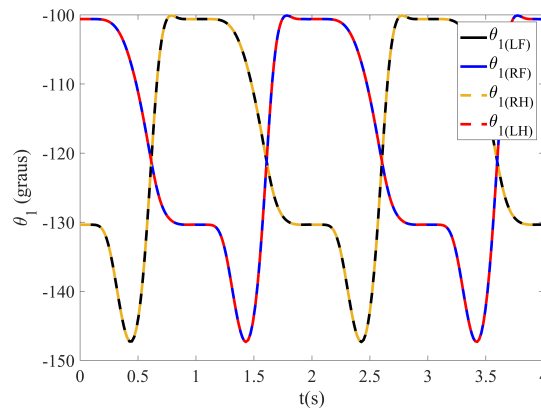
Fonte: Autoria própria

Já entre os instantes 2 s e 3 s, que compreendem a *swing phase*, θ_1 atinge seu valor máximo em módulo, chegando próximo a -150° , o que representa a retração máxima do *upper link*. Posteriormente, o ângulo θ_1 assume valores menores em módulo, retornando ao ângulo inicial de, aproximadamente, -100° e reiniciando o ciclo de locomoção com a execução da *stance phase*.

Analisando-se o gráfico de θ_2 ao longo do tempo exibido na Figura 31, durante a *stance phase* (entre 1 s e 2 s), observa-se um pico de aproximadamente 60° , correspondente à passagem da perna pelo ponto (0,00 mm, -376,41 mm) explicitado na curva de Bézier da Figura 29, já na *swing phase* (entre 2 s e 3 s), observa-se um pico maior, próximo de 90° , correspondendo, da mesma forma que o observado no gráfico temporal de θ_1 , ao ponto em que há a maior retração da perna e por onde o pé de borracha passa pelas coordenadas (16,62 mm, -302,16 mm).

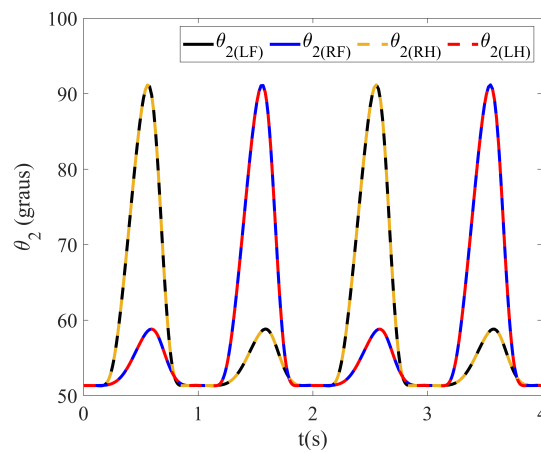
Os ângulos θ_1 de todas as pernas foram comparados entre si e plotados ao longo do tempo na Figura 32, na qual pode-se perceber que há superposição entre curvas referentes aos ângulos θ_1 das pernas LF e RH e entre os ângulos das pernas RF e LH, contudo, verifica-se que entre os ângulos das pernas LF e RF e entre os das patas LH e RH, há uma defasagem de 180° . Tal comportamento é análogo àquele observado para as variáveis de estado comparadas na subseção 5.1 e caracteriza a marcha trote. A sincronia entre pernas opostas diagonalmente e a defasagem entre pernas opostas sagitalmente, características da marcha supracitada, também podem ser observadas comparando-se os ângulos θ_2 referentes às pernas do robô quadrúpede ao longo do tempo, conforme explicita o gráfico da Figura 33.

Figura 32 – Ângulos $\theta_{1(LF)}$, $\theta_{1(RF)}$, $\theta_{1(RH)}$ e $\theta_{1(LH)}$ plotados ao longo do tempo



Fonte: Autoria própria.

Figura 33 – Ângulos $\theta_{2(LF)}$, $\theta_{2(RF)}$, $\theta_{2(RH)}$ e $\theta_{2(LH)}$ plotados ao longo do tempo



Fonte: Autoria própria.

5.4 Resultados da cosimulação Adams-Simulink

5.4.1 Ângulos das juntas do protótipo virtual do Adams

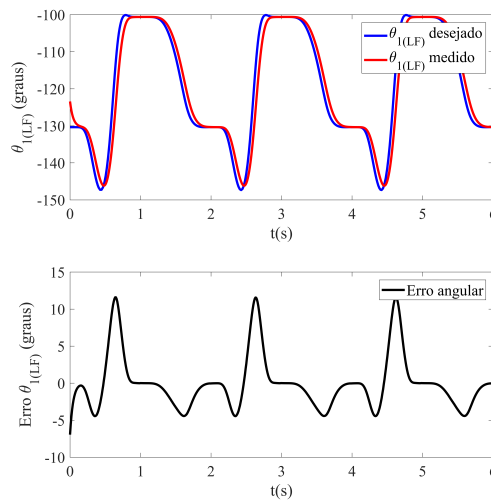
As trajetórias geradas de forma *offline* foram executadas pelo protótipo virtual do Adams por meio da malha de controle de posição com acionamento cinemático por velocidade construída no Simulink para viabilizar a cosimulação, conforme ilustra o diagrama de blocos da Figura 20.

O erro de posição angular, resultante da diferença entre os ângulos das juntas desejados e os ângulos medidos, foi submetido a um controlador cujo ganho proporcional P fora definido empiricamente como sendo igual a 20. Tal definição foi feita de modo a respeitar a sensibilidade do contato entre os pés de borracha do robô e o solo, já que variações angulares abruptas multiplicadas por um ganho muito alto poderiam suscitar elevadas velocidades, ocasionando elevados

impactos contra o solo e produzindo solavancos na trajetória, descaracterizando a marcha trote.

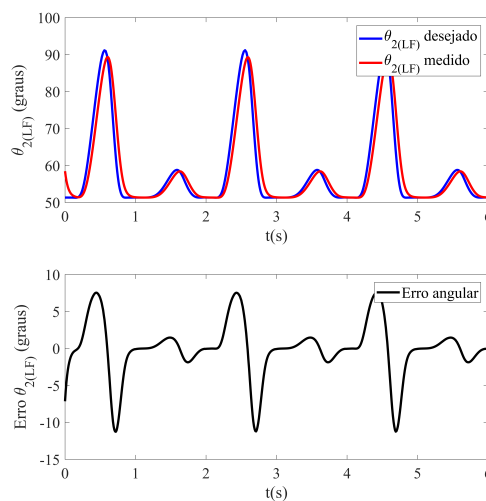
Os gráficos da Figura 34 mostram a comparação entre os ângulos θ_1 desejados e medidos para a perna LF em conjunto com o erro angular correspondente, resultante da aplicação do controlador com ganho proporcional P igual a 20. Os mesmos tipos de gráficos são exibidos para θ_2 na Figura 35.

Figura 34 – Gráfico superior contendo comparação entre os ângulos $\theta_{1(LF)}$ desejados e medidos ao longo do tempo e gráfico inferior contendo erro de posição angular resultante da comparação entre os ângulos $\theta_{1(LF)}$ desejados e medidos ao longo do tempo



Fonte: Autoria própria.

Figura 35 – Gráfico superior contendo comparação entre os ângulos $\theta_{2(LF)}$ desejados e medidos ao longo do tempo e gráfico inferior contendo erro de posição angular resultante da comparação entre os ângulos $\theta_{2(LF)}$ desejados e medidos ao longo do tempo



Fonte: Autoria própria.

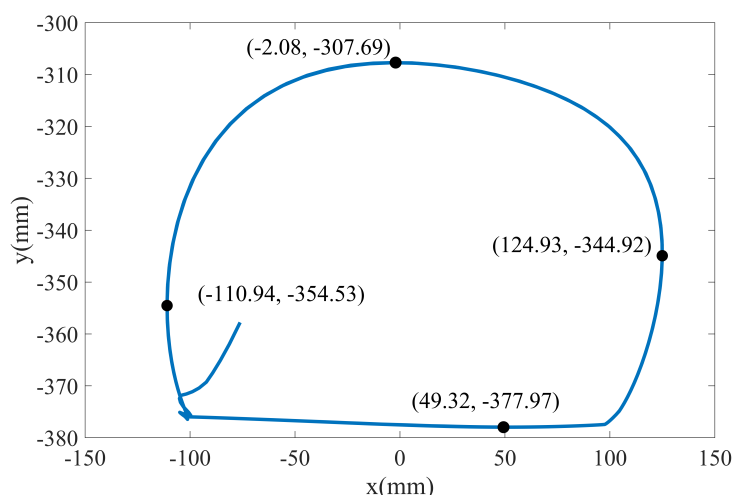
É possível observar tanto no gráfico da Figura 34 quanto no da Figura 35 que ângulos desejados e ângulos medidos apresentam valores muito próximos entre si, sendo que o erro angular máximo registrado assume um valor próximo de 10° durante a *swing phase*. Isto ocorre por conta de um erro de defasagem, isto é, como a variação neste período é muito brusca, o controlador, com o ganho proporcional definido, não consegue acompanhá-la instantaneamente, inculindo em uma defasagem entre ângulo medido e desejado.

Os gráficos de θ_1 e de θ_2 desejados e medidos para as demais pernas com os erros de posição angular correspondentes apresentam comportamento análogo ao da perna LF e podem ser consultados nos apêndices D, E, F, G, H e I.

5.4.2 Trajetórias cinemáticas desenvolvidas pelas pernas do protótipo virtual do Adams

A trajetória do pé de borracha da perna LF foi medida e pode ser vista na Figura 36 enquanto as trajetórias aferidas para as demais pernas podem ser consultadas no apêndice J.

Figura 36 – Trajetória do pé de borracha da perna LF durante a cosimulação



Fonte: Autoria própria.

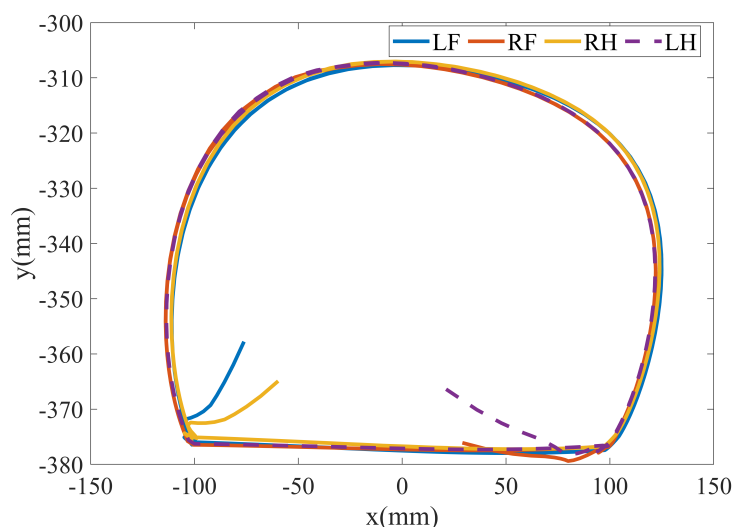
Pode-se observar que a trajetória desenvolvida pelo pé de borracha da perna LF é análoga à curva de Bézier implementada no MATLAB e utilizada para a geração dos ângulos das juntas do protótipo virtual, apresentando amplitude em x próxima de 236 mm e em y próxima de 70 mm, resultados semelhantes às respectivas amplitudes próximas de 246 mm e de 74 mm aferidas a partir do gráfico da Figura 29.

Ademais, nota-se que há um trecho da trajetória que é executado fora dos limites da curva de Bézier. A existência de tal trecho é motivada pela condição inicial da pata no próprio desenho CAD importado para o Adams, que parte de uma região interna aos limites da curva para, em seguida, adequar sua trajetória ao comportamento almejado.

As trajetórias dos quatro pés de borracha referentes às patas LF, RF, RH e LH foram

comparadas no gráfico apresentado na Figura 37.

Figura 37 – Trajetórias dos pés de borracha das pernas LF, RF, RH e LH durante a cosimulação



Fonte: Autoria própria.

Nota-se na Figura 37 que as trajetórias se sobrepõem, evidenciando a similaridade entre as coordenadas das mesmas. Além disso, é nítido que, das condições iniciais do desenho CAD, as pernas se movimentam para posições diferentes compreendidas pelas curvas de Bézier. Isto se deve às condições iniciais definidas na Tabela 10 implementadas no MATLAB e refletidas no protótipo virtual, com isso, o par composto por LF e RH inicia o movimento posicionando os pés de borracha no limiar da *swing phase*, à esquerda da Figura 37 enquanto o par de pernas constituído por RF e LH posiciona os pés no limiar da *stance phase*, à direita da mesma figura.

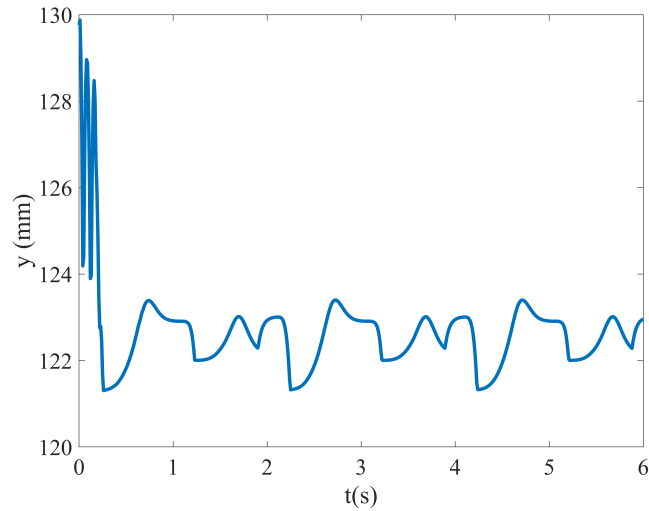
5.5 Estabilidade da locomoção do robô quadrúpede

A estabilidade da locomoção do protótipo virtual foi analisada com base na posição e na orientação do corpo central do robô quadrúpede durante a execução da marcha trote. A partir de um *marker* localizado no corpo central do robô, mediu-se a distância vertical (apenas no eixo y) com relação ao sistema de coordenadas global do Adams ao longo do tempo de cosimulação em que o protótipo se movimentou, gerando o gráfico exibido na Figura 38.

Analisando-se a Figura 38, percebe-se que há um regime transitório que dura menos de 0,5 s, sendo motivado pelo fato de que o robô modelado não inicia a cosimulação em contato com o solo e sim suspenso. Ao iniciar a cosimulação, o mesmo cai e entra em contato com o solo, contudo, por conta do impacto sofrido e da movimentação das patas do robô que já começa quando o mesmo está suspenso, observa-se a oscilação inicial na altura registrada na Figura 38.

Após o transitório, nota-se que a altura assume valores quase constantes, variando entre, aproximadamente, 121 mm e 123 mm ao longo dos 6 s de cosimulação, evidenciando um comportamento estável do robô ao longo de sua locomoção.

Figura 38 – Medição da altura (apenas no eixo y) de um *marker* localizado no centro de massa do corpo central do protótipo virtual do Adams ao longo do tempo de cosimulação

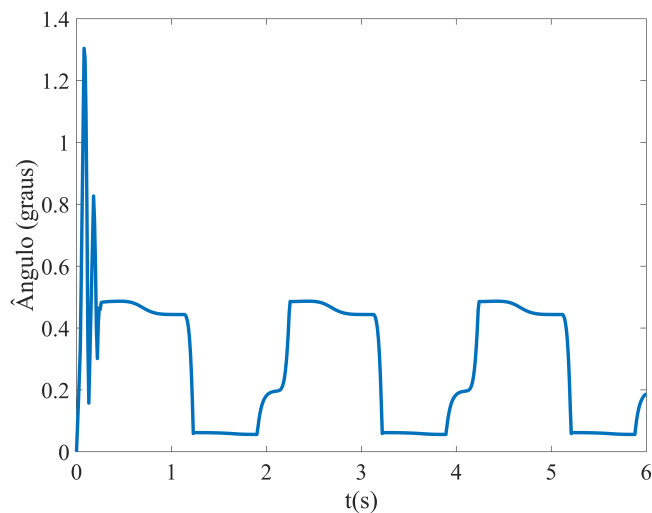


Fonte: Autoria própria.

É válido ressaltar que os valores de y apresentados na Figura 38 não são iguais à altura do robô, já que a origem do sistema de coordenadas global do Adams não está localizada no plano com o qual o protótipo entra em contato e sobre o qual executa a marcha almejada.

O movimento do robô quadrúpede foi restringido ao plano sagital por meio da inserção de uma junta planar, conforme explicita a subseção 4.1, logo, o protótipo só consegue rotacionar em torno de um eixo ortogonal ao plano em questão que, no caso constitui o eixo z. Por isso, para a avaliação da orientação do robô ao longo do trote, mediu-se o ângulo de rotação em torno do eixo z, denominado *pitch* (ψ), a partir do mesmo *marker* localizado no centro de massa do corpo central do robô, resultando no gráfico temporal exibido na Figura 39.

Figura 39 – Medição do ângulo de rotação em torno do eixo z de um *marker* localizado no centro de massa do corpo central do protótipo virtual do Adams ao longo do tempo de cosimulação



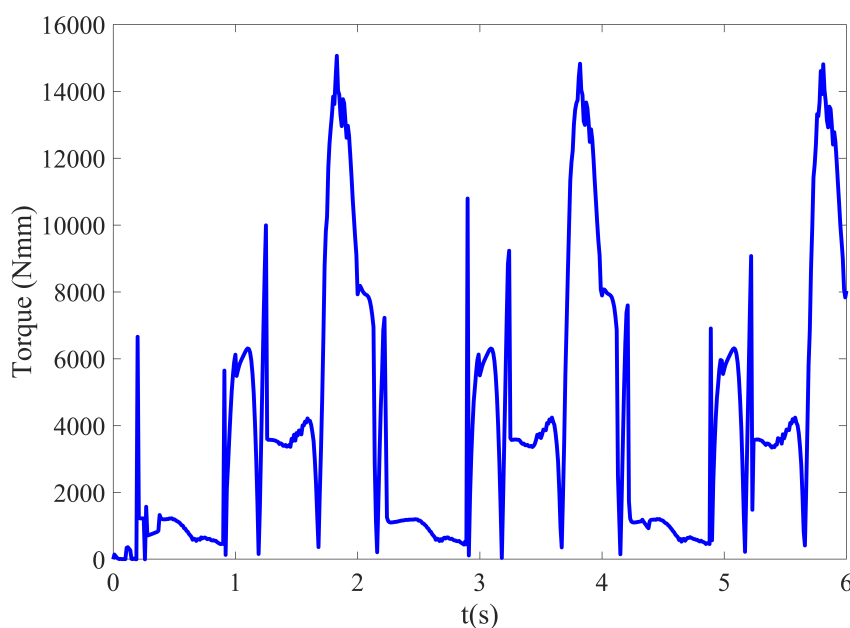
Fonte: Autoria própria.

Nota-se no gráfico da Figura 39 que há um regime transitório que dura menos de 0,5 s, representando a variação angular motivada pelo impacto do robô com o solo no início da cosimulação, assim como o observado na medição apresentada na Figura 38. Após o transitório, observa-se que o ângulo aferido assume valores muito próximos entre si, entre $0,1^\circ$ e $0,5^\circ$, aproximadamente, isto é, mesmo com a movimentação das pernas, a orientação do robô se mantém estável durante a execução da marcha na cosimulação.

5.6 Torque nas juntas das pernas do robô quadrúpede

Os torques nas juntas 1 e 2 da perna LF do protótipo virtual foram aferidos durante a execução da marcha trote na cosimulação, resultando nos gráficos exibidos na Figura 40 e na Figura 41, respectivamente.

Figura 40 – Medição de torque na junta 1 da perna LF do protótipo virtual durante a execução da marcha trote na cosimulação via Adams-Simulink



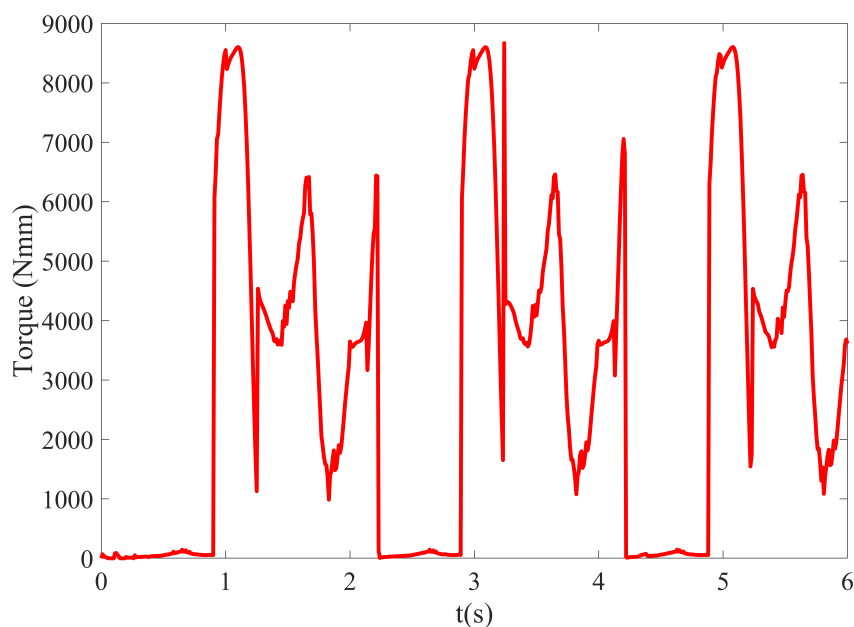
Fonte: Autoria própria.

Analisando-se tanto o gráfico da Figura 40 quanto o da Figura 41, é perceptível que os maiores torques acontecem durante a *stance phase* (entre 1 s e 2 s), já que corresponde à fase de contato, e consequentemente a de maior esforço, em que a pata LF deve exercer uma força contra o solo de modo a propulsionar o robô para a frente. Já durante a *swing phase* (entre 2 s e 3 s), a junta motorizada não deve impulsionar o protótipo e sim sustentar apenas a perna LF durante a execução de sua trajetória, resultando em valores de torque menores.

Comparando-se os torques de ambas as juntas, percebe-se que os valores medidos na junta motorizada 1 são maiores do que os na junta motorizada 2, uma vez que o esforço empregado pela primeira deve ser suficiente não apenas para mover o *link* superior, mas também o *link*

inferior acoplado, o que não ocorre com a junta motorizada 2, responsável exclusivamente pela movimentação do *link* inferior.

Figura 41 – Medição de torque na junta 2 da perna LF do protótipo virtual durante a execução da marcha trote na cosimulação via Adams-Simulink



Fonte: Autoria própria.

Ademais, percebe-se que o valor máximo de torque empregado pela junta 1 está em torno de 15000 Nmm. Tal valor é coerente com o valor nominal de torque do servo motor do *Mini Cheetah*, igual a 17000 Nmm (KATZ, 2018).

6 CONCLUSÃO

Com base nos resultados expostos e nas considerações feitas ao longo deste relatório, nota-se que a implementação dos osciladores não-lineares constituintes do CPG foi bem-sucedida, já que os mesmos apresentaram comportamento de convergência para ciclos-limite tanto para as variáveis de estado resultantes quanto para as parametrizações feitas com base nestas, como a geração de trajetórias cinemáticas e a geração dos ângulos para as juntas das pernas do protótipo virtual.

Foi possível também tanto gerar curvas de Bézier que respeitaram o espaço de trabalho das pernas do robô quadrúpede, a partir da escolha correta dos pontos de controle, quanto executá-las de forma satisfatória durante a cosimulação por meio da escolha dos ângulos das juntas adequados à postura inicial das pernas do robô.

Conseguiu-se executar uma cosimulação com execução de marcha estável e com valores de torque coerentes com os projetados para o *Mini Cheetah* apesar da sensibilidade dos parâmetros de contato e da simplicidade do controlador implementado. Isto porque o ganho proporcional e os períodos da *stance phase* e da *swing phase* foram definidos adequadamente de modo a não suscitar velocidades excessivas da movimentação das pernas.

Apesar das restrições impostas à movimentação do protótipo virtual, a junta planar inserida é facilmente removível e as juntas motorizadas que representam os atuadores de adução e abdução precisam apenas de uma entrada não nula para que possam funcionar. Ademais a representação do protótipo na malha de controle do Simulink como sendo apenas um bloco permite trocas facilitadas em caso de geração de novas plantas de controle e, de forma similar, as entradas desejadas inseridas podem ser facilmente alteradas por outras providas de metodologias distintas sem que haja necessidade de se modificar qualquer outro parâmetro.

Portanto, pode-se concluir que além de ter permitido tanto a geração de trajetórias cinemáticas quanto a execução do controle cinemático de posição com acionamento por velocidade de maneira satisfatória, o ambiente de cosimulação criado trouxe consigo facilidade de alteração de diversos parâmetros-chave para o controle da locomoção em *softwares* de uso comum em cursos de Engenharia, sendo possível utilizá-lo como um ambiente virtual para a implementação de técnicas de controle da locomoção de um robô quadrúpede.

REFERÊNCIAS

- ALEXANDER, R. M. The gaits of bipedal and quadrupedal animals. *The International Journal of Robotics Research*, v. 3, n. 2, p. 49–59, 1984.
- AN, L.; HENG, W.; YONGZHENG, L. Gait transition of quadruped robot using rhythm control and stability analysis. In: IEEE. *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. [S.l.], 2013. p. 2535–2539.
- BBC. *Como hobby de engenheiro japonês solucionou grande problema do trem-bala*. 2017. Disponível em: <https://www.bbc.com/portuguese/geral-42193691>.
- DEVITO, F. *Boston Dynamics: vídeo mostra robô Atlas fazendo parkour*. 2021. Disponível em: <https://olhardigital.com.br/2021/08/17/reviews/boston-dynamics-video-mostra-robo-atlas-fazendo-parkour/>.
- DYNAMICS, B. *What to Do with a Legged Robot in Academia and Research*. 2023. Disponível em: <https://bostondynamics.com/blog/what-to-do-with-a-legged-robot-in-academia-and-research/>.
- FIGUEIREDO, M. *Quais são as etapas de um projeto CAE?* 2021. Acesso em 02/11/2023. Disponível em: <https://www.manufaturadigital.com/quais-sao-as-etapas-de-um-projeto-cae/>.
- GALLAGHER, M. B. *From “cheetah-noids” to humanoids*. 2021. Disponível em: <https://meche.mit.edu/news-media/%E2%80%9Ccheetah-noids%E2%80%9D-humanoids>.
- GMBH, H. I. H. *How To Choose the right Squash Ball*. 2021. Disponível em: https://www.head.com/en_GB/squash/squash-how-tos/how-to-choose-the-right-squash-ball.
- GOLUBITSKY, M. *et al.* A modular network for legged locomotion. *Physica D: Nonlinear Phenomena*, v. 115, n. 1, p. 56–72, 1998.
- GRABCAD. *About GrabCAD*. 2017. Acesso em 05/11/2023. Disponível em: <https://resources.grabcad.com/company/>.
- HEXAGON. *Adams/View help - Adams 2014*. 2021. Disponível em: <https://simcompanion.hexagon.com/customers/s/article/adams-view-help---adams-2014-doc10647>.
- HILDEBRAND, M. Analysis of asymmetrical gaits. *Journal of Mammalogy*, v. 58, n. 2, p. 131–156, 1977.
- HONGTAO, Z. Inverse kinematics simulation of a parallel robot with few degree of freedom. *Mechanical Engineering Automation*, 2009.
- HYUN, D. J. *et al.* High speed trot-running: Implementation of a hierarchical controller using proprioceptive impedance control on the MIT Cheetah. *The International Journal of Robotics Research*, v. 33, n. 11, p. 1417–1445, 2014.
- IJSPEERT, A. J. Central pattern generators for locomotion control in animals and robots: a review. *Neural networks*, v. 21, n. 4, p. 642–653, 2008.

JORDAN, D. W.; SMITH, P. *Nonlinear ordinary differential equations : an introduction for scientists and engineers*. 4. ed. -. ed. New York: Oxford University Press, 2007. ISBN 9780199208241.

KATZ, B.; CARLO, J. D.; KIM, S. Mini cheetah: A platform for pushing the limits of dynamic quadruped control. In: IEEE. *2019 international conference on robotics and automation (ICRA)*. [S.l.], 2019. p. 6295–6301.

KATZ, B. G. *A low cost modular actuator for dynamic robots*. Tese (Doutorado) — Massachusetts Institute of Technology, 2018.

LELE, A. S. *et al.* Learning to walk: Bio-mimetic hexapod locomotion via reinforcement-based spiking central pattern generation. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, IEEE, v. 10, n. 4, p. 536–545, 2020.

LI, Z.; KOTA, S. Virtual prototyping and motion simulation with adams. *J. Comput. Inf. Sci. Eng.*, v. 1, n. 3, p. 276–279, 2001.

LIPENG, Y.; BING, L. Research on gait switching control of quadruped robot based on dynamic and static combination. *IEEE Access*, IEEE, v. 11, p. 14073–14088, 2023.

MA, Z.; LIANG, Y.; TIAN, H. Research on gait planning algorithm of quadruped robot based on central pattern generator. In: IEEE. *2020 39th Chinese Control Conference (CCC)*. [S.l.], 2020. p. 3948–3953.

MATHWORKS. *MATLAB ode45 - Solve nonstiff differential equations - medium order method*. 2023. Disponível em: <https://www.mathworks.com/help/matlab/ref/ode45.html>.

MATHWORKS. *Simulink - Simulation and Model-Based Design - MATLAB*. 2023. Disponível em: <https://www.mathworks.com/products/simulink.html>.

MATHWORKS. *Workspace Variables and MAT-Files*. 2023. Disponível em: <https://www.mathworks.com/help/matlab/workspace.html>.

MORAIS, S.; SINGH, H.; ACHARYA, A. Trajectory and gait planning of a quadruped using adams and simulink. In: IEEE. *2021 4th Biennial International Conference on Nascent Technologies in Engineering (ICNTE)*. [S.l.], 2021. p. 1–6.

MÜLLER, A. L.; SAFFARO, F. A. Virtual prototyping for detail design in construction. *Ambiente Construído*, SciELO Brasil, v. 11, p. 105–121, 2011.

OGATA, K. *Engenharia de controle moderno*. 5ª. São Paulo: Pearson, 2011.

PAULA, D. T. d. *Locomoção dinâmica quadrúpede uma abordagem bio-inspirada baseada no controle hierárquico*. Sorocaba: Trabalho de conclusão de curso - Universidade Estadual Paulista "Júlio de Mesquita Filho", Instituto de Ciência e Tecnologia, Sorocaba, 2021.

RIGHETTI, L.; IJSPEERT, A. J. Design methodologies for central pattern generators: an application to crawling humanoids. In: *Proceedings of robotics: Science and systems*. [S.l.: s.n.], 2006. p. 191–198.

RIGHETTI, L.; IJSPEERT, A. J. Pattern generators with sensory feedback for the control of quadruped locomotion. In: IEEE. *2008 IEEE International Conference on Robotics and Automation*. [S.l.], 2008. p. 819–824.

SANDINI, G.; METTA, G.; VERNON, D. Robotcub: an open framework for research in embodied cognition. In: *4th IEEE/RAS International Conference on Humanoid Robots, 2004*. [S.l.: s.n.], 2004. v. 1, p. 13–32 Vol. 1.

SEBRAE. *Biomimética: o que é e ideias de negócio*. 2018. Disponível em: <https://inovacaosebraeminas.com.br/biomimetica-o-que-e/>.

SHARBAFI, M. A.; SEYFARTH, A. *Bioinspired legged locomotion: models, concepts, control and applications*. [S.l.]: Butterworth-Heinemann, 2017.

SICILIANO, B. *et al. Robotics: Modelling, Planning and Control*. Springer London, 2010. (Advanced Textbooks in Control and Signal Processing). ISBN 9781846286414. Disponível em: <https://books.google.com.br/books?id=jPCAFmE-logC>.

SINGULARIDADES, I. *Biomimética: o que é e como ela pode ser utilizada na escola. O Estado de S. Paulo*, 2020.

STROGATZ, S. H. *Nonlinear dynamics and chaos with student solutions manual: With applications to physics, biology, chemistry, and engineering*. [S.l.]: CRC press, 2018.

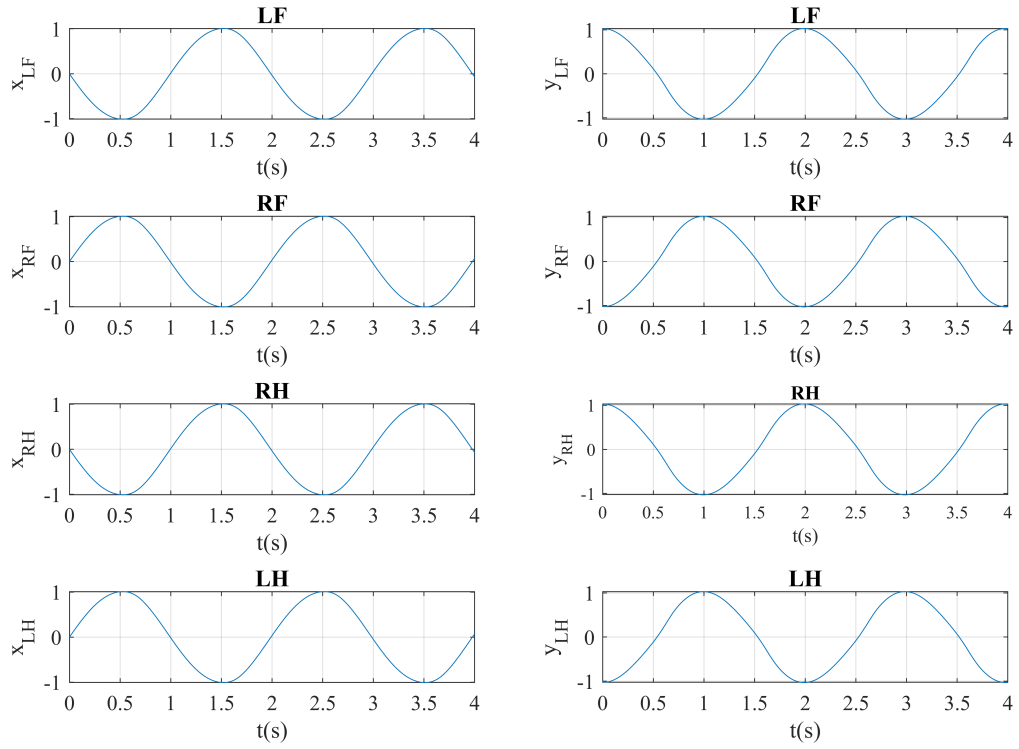
ULRICH, K.; EPPINGER, S. *Product Design and Development*. McGraw-Hill Higher Education, 2008. (McGraw-Hill). ISBN 9780071259477. Disponível em: <https://books.google.com.br/books?id=Bn4UQAAACAAJ>.

WU, X. *et al.* CPGs with continuous adjustment of phase difference for locomotion control. *International Journal of Advanced Robotic Systems*, v. 10, n. 6, 2013.

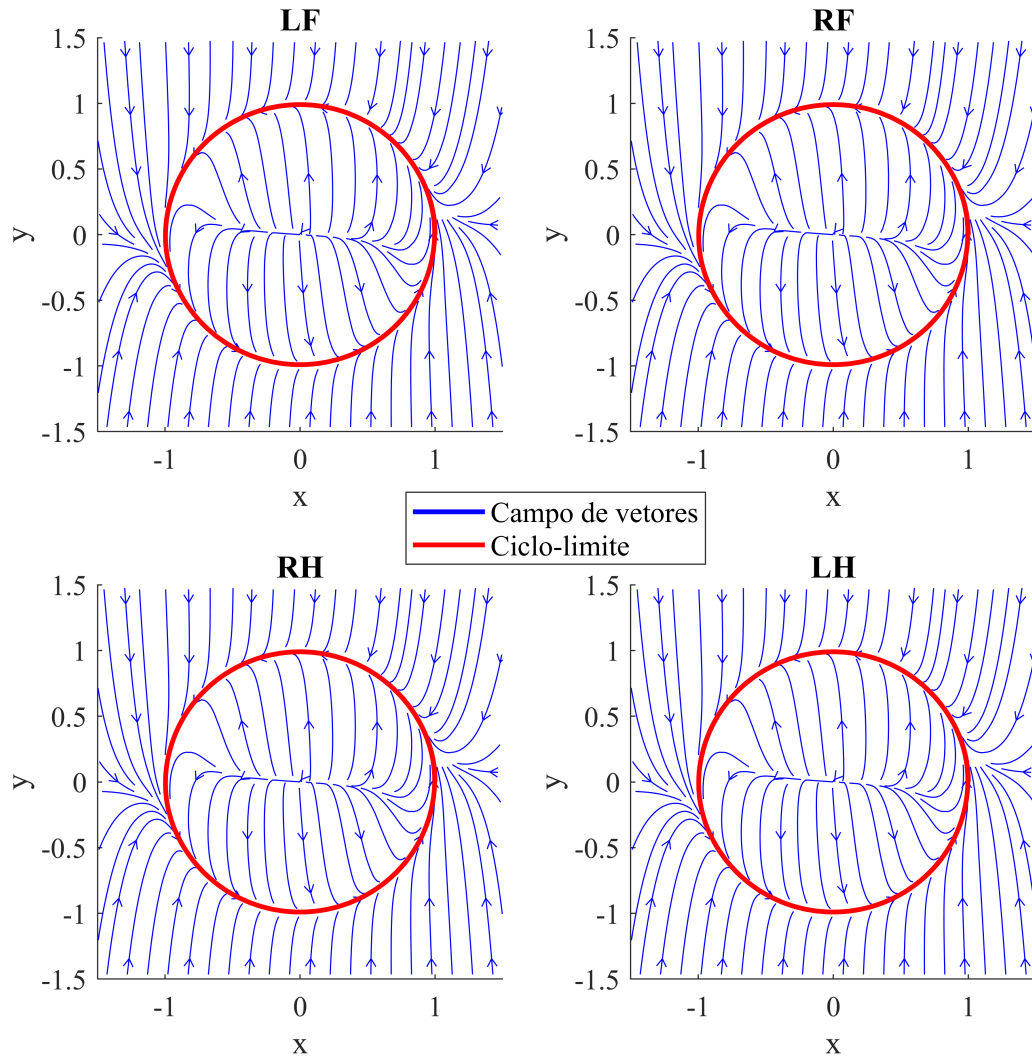
XIE, J. *et al.* Adaptive walking on slope of quadruped robot based on cpg. In: IEEE. *2019 2nd World Conference on Mechanical Engineering and Intelligent Manufacturing (WCMEIM)*. [S.l.], 2019. p. 487–493.

ZILL, D. G.; CULLEN, M. R. *Equações diferenciais vol. 1*. [S.l.]: Pearson Makron Books, 2008.

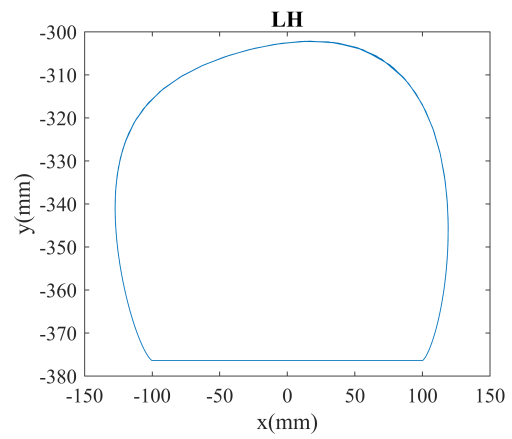
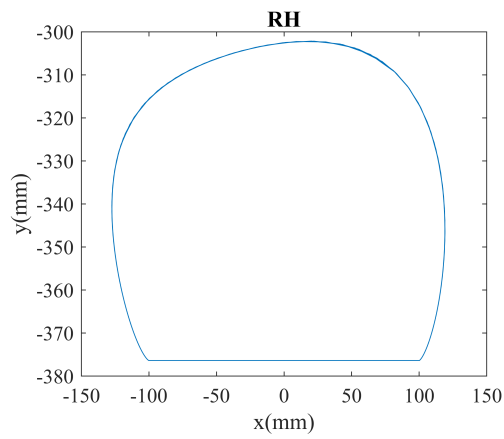
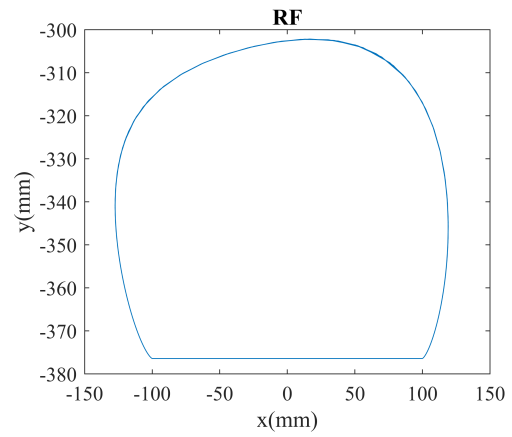
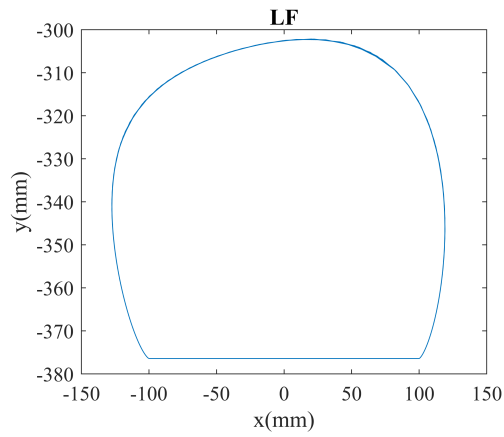
APÊNDICE A - Variáveis de estado x e y dos osciladores associados às pernas LF, RF, RH e LH do robô quadrúpede plotadas ao longo do tempo



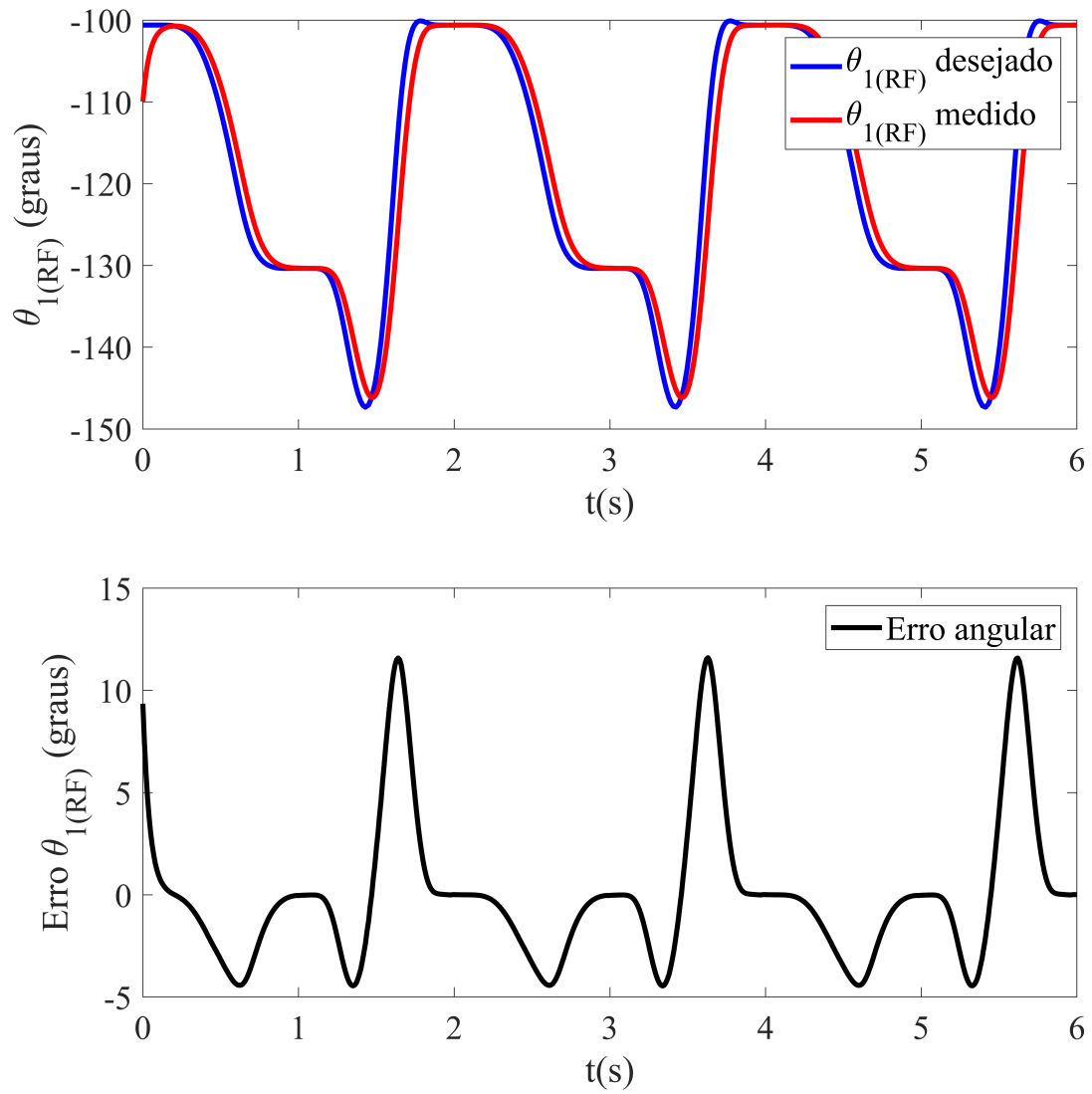
APÊNDICE B - Diagramas de fase dos osciladores associados às pernas LF, RF, RH e LH do robô quadrúpede contendo ciclos-limite de amplitude unitária



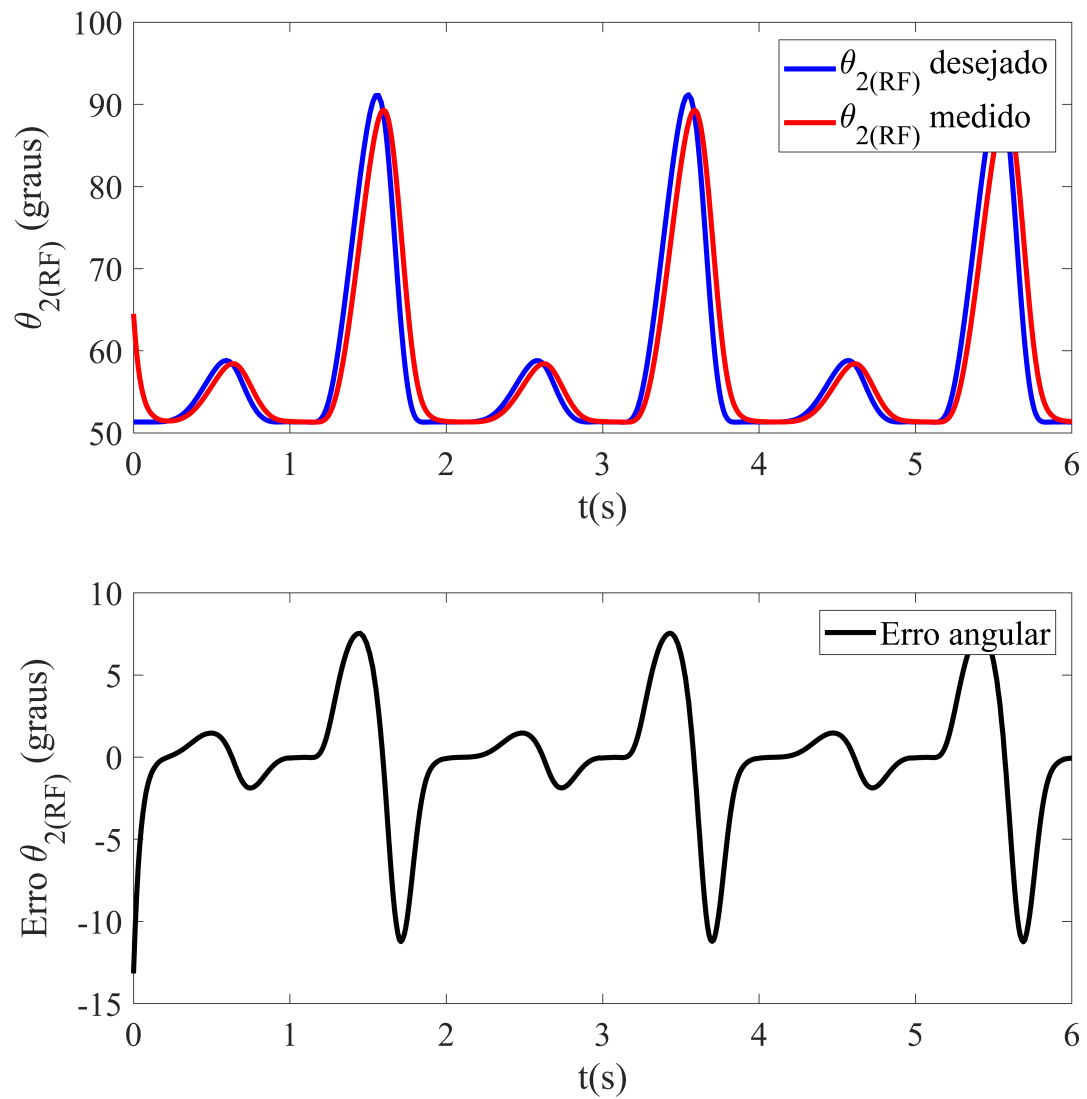
APÊNDICE C - Curvas de Bézier projetadas para as pernas LF, RF, RH e LH no MATLAB e plotadas em um sistema de coordenadas cartesianas



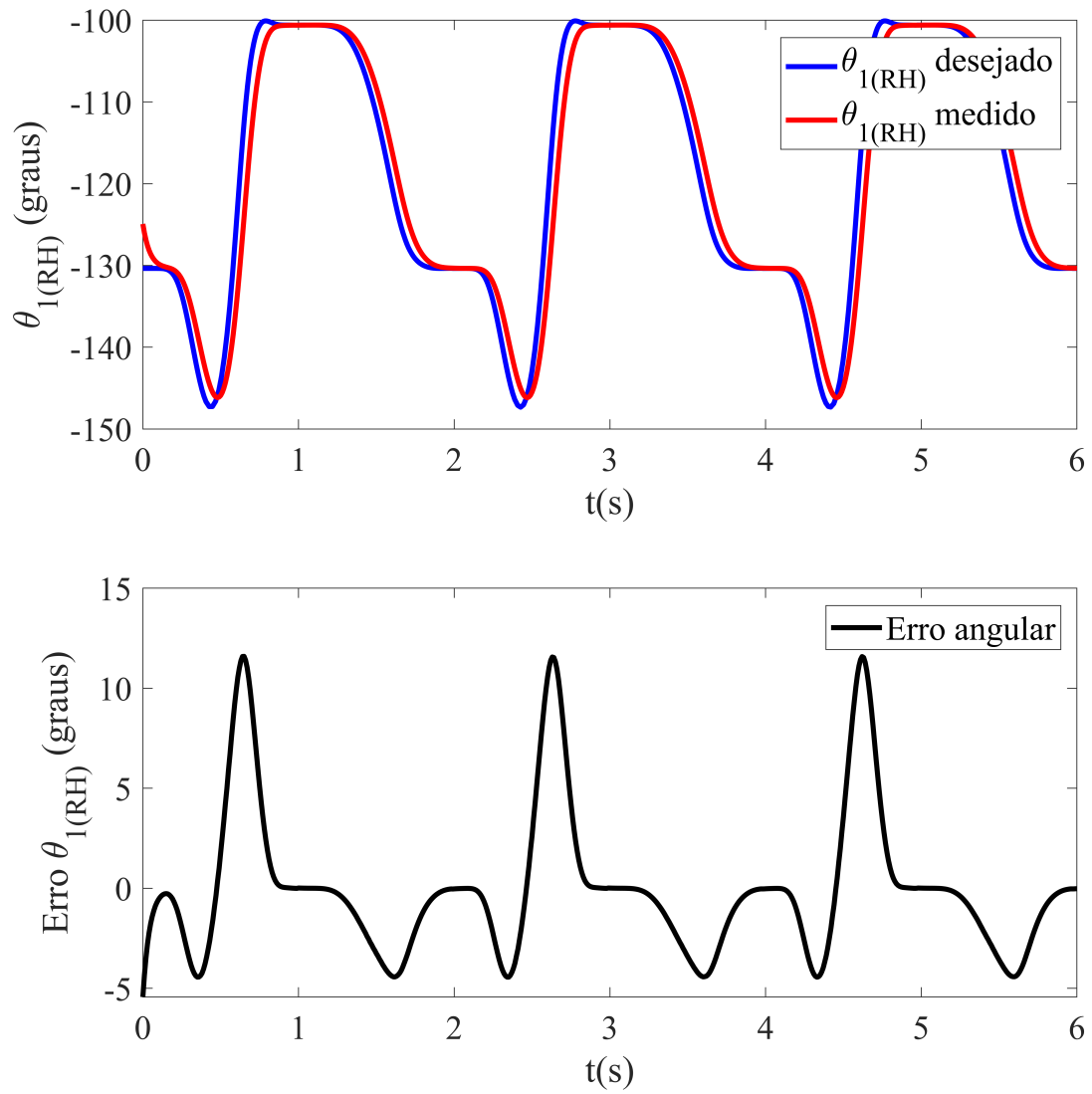
APÊNDICE D - Comparação entre os ângulos $\theta_{1(RF)}$ desejados e medidos ao longo do tempo (gráfico superior) em conjunto com o erro de posição angular correspondente (gráfico inferior)



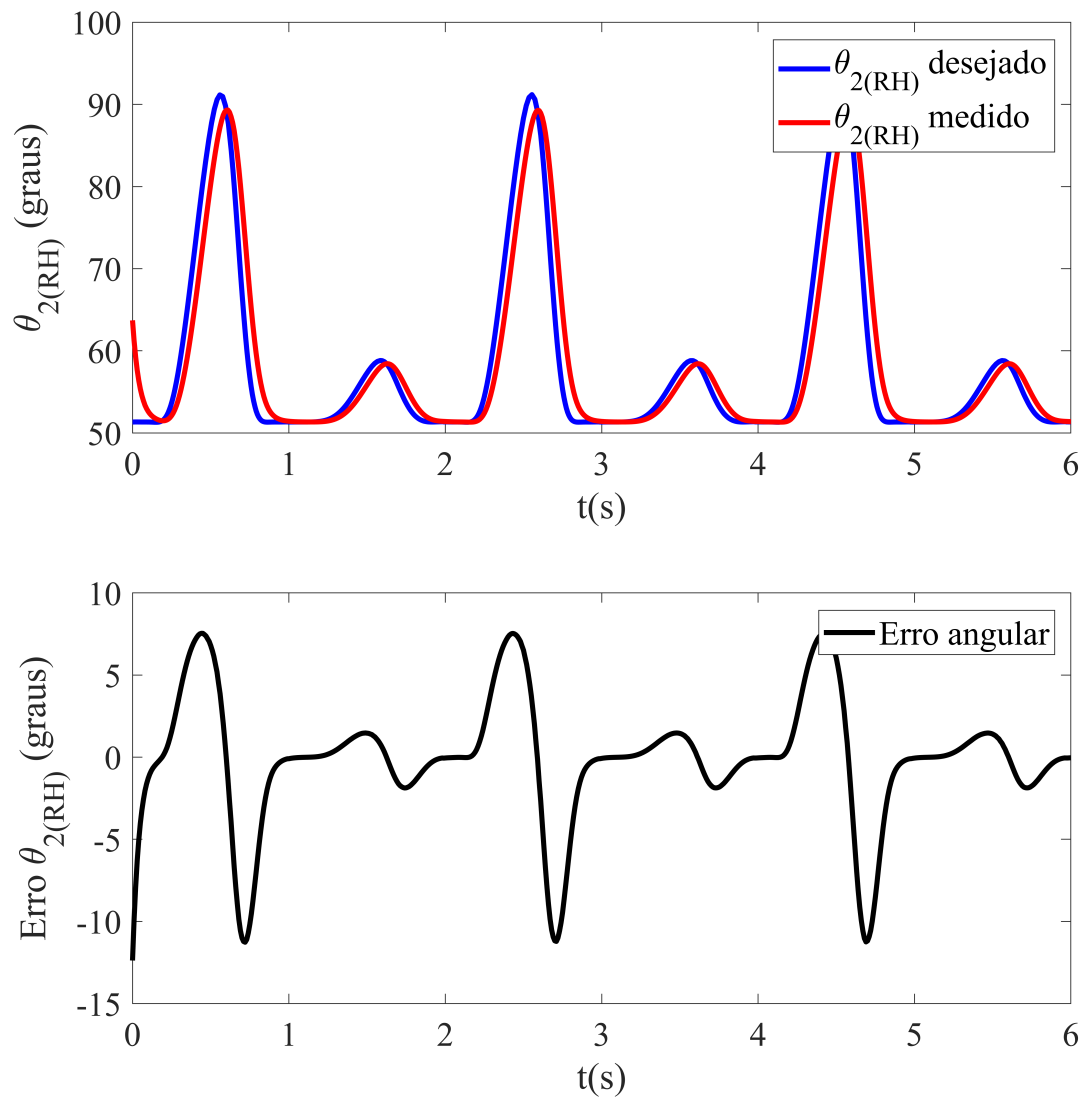
APÊNDICE E - Comparação entre os ângulos $\theta_{2(RF)}$ desejados e medidos ao longo do tempo (gráfico superior) em conjunto com o erro de posição angular correspondente (gráfico inferior)



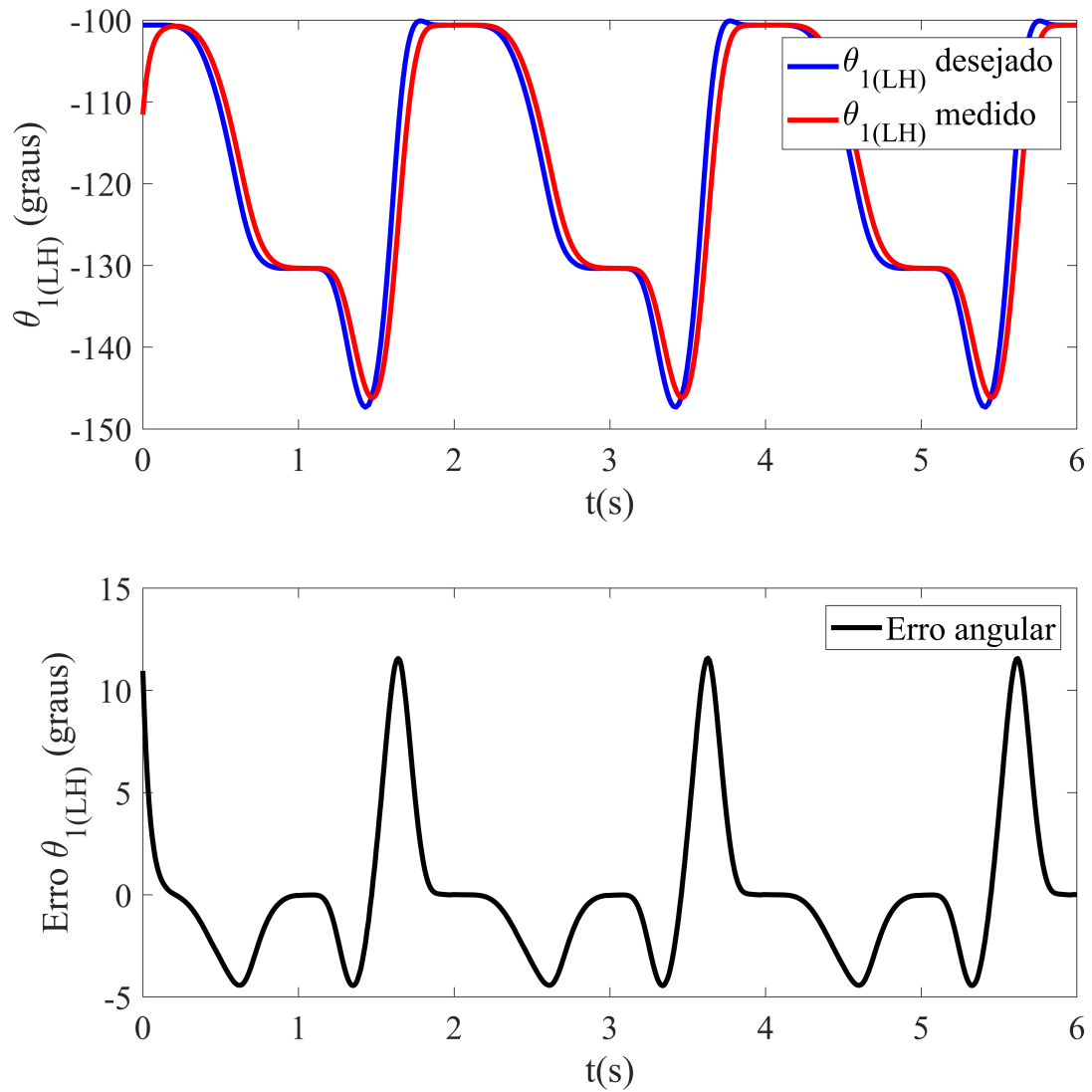
APÊNDICE F - Comparação entre os ângulos $\theta_{1(RH)}$ desejados e medidos ao longo do tempo (gráfico superior) em conjunto com o erro de posição angular correspondente (gráfico inferior)



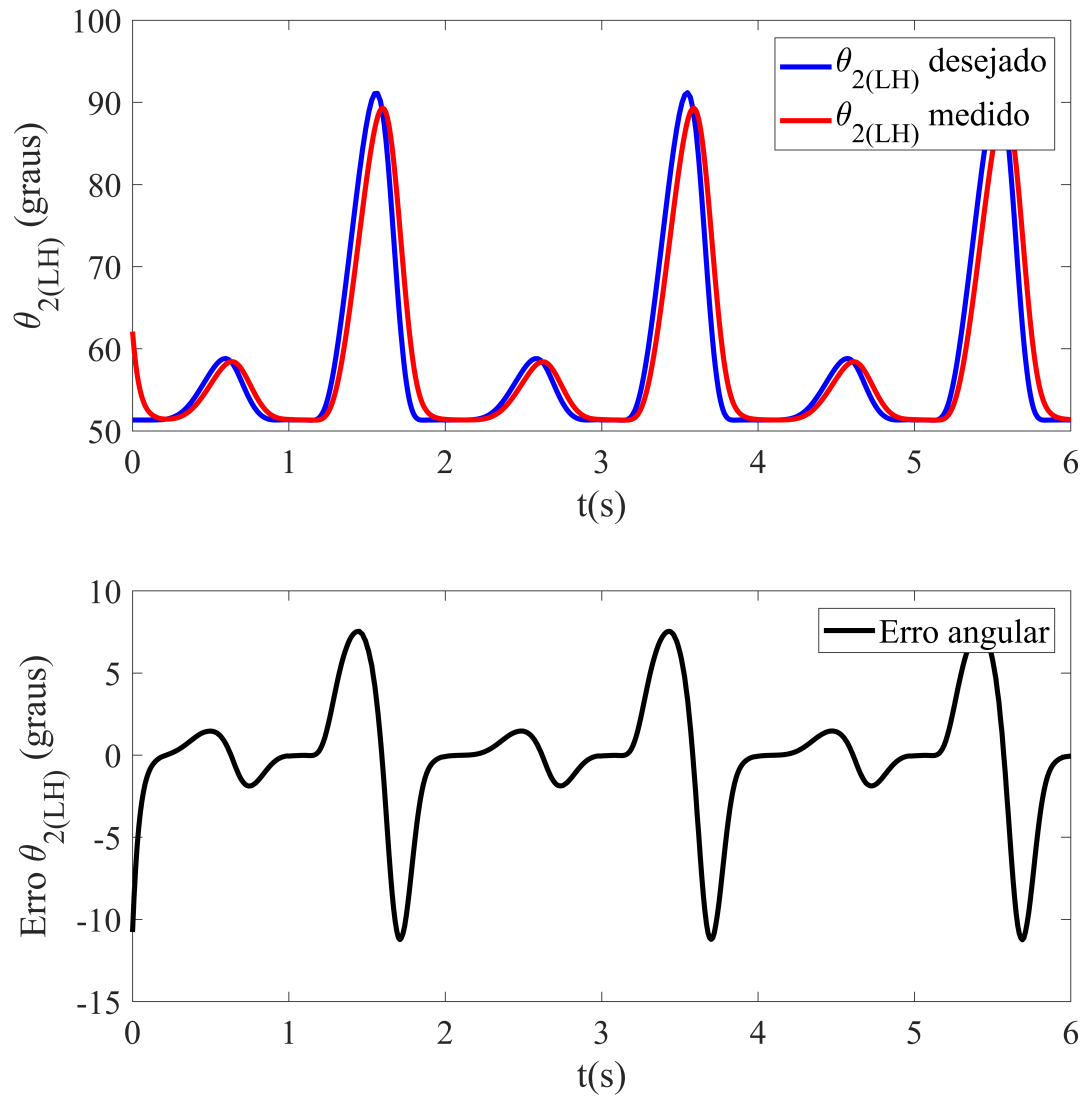
APÊNDICE G - Comparação entre os ângulos $\theta_{2(RH)}$ desejados e medidos ao longo do tempo (gráfico superior) em conjunto com o erro de posição angular correspondente (gráfico inferior)



APÊNDICE H - Comparação entre os ângulos $\theta_{1(LH)}$ desejados e medidos ao longo do tempo (gráfico superior) em conjunto com o erro de posição angular correspondente (gráfico inferior)



APÊNDICE I - Comparação entre os ângulos $\theta_{2(LH)}$ desejados e medidos ao longo do tempo (gráfico superior) em conjunto com o erro de posição angular correspondente (gráfico inferior)



APÊNDICE J - Curvas de Bézier executadas pelas pernas LF, RF, RH e LH durante a cosimulação Adams-Simulink

