Universidade de Brasília
Faculdade de Tecnologia

Análise de Parâmetros de Movimentação e da Influência do atrito em Robôs Cobra

Mariana Mendanha da Cruz

PROJETO FINAL DE CURSO

ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Brasília

2023

Universidade de Brasília
Faculdade de Tecnologia

Análise de Parâmetros de Movimentação e da Influência do atrito em Robôs Cobra

Mariana Mendanha da Cruz

Projeto Final de Curso submetido como requisito parcial para obtenção do grau de Engenheiro de Controle e Automação

Orientadora: Prof. Dr. Carla Maria Chagas e Cavalcante Koike
Coorientadora: Prof. Dr. Dianne Magalhães Viana

Brasília
2023

Mendanha da Cruz, Mariana.
M537a Análise de Parâmetros de Movimentação e da Influência do atrito em Robôs Cobra / Mariana Mendanha da Cruz; orientadora Carla Maria Chagas e Cavalcante Koike; coorientadora Dianne Magalhães Viana. -- Brasília, 2023.
67 p.

Projeto Final de Curso (Engenharia de Controle e Automação) -- Universidade de Brasília, 2023.

1. Robótica Bioinspirada. 2. Robô Modular. 3. Locomoção de Robôs Cobra. 4. Análise de Parâmetros. 5. Atrito Anisotrópico. I. Chagas e Cavalcante Koike, Carla Maria, orient. II. Magalhães Viana, Dianne, coorient. III. Título

Universidade de Brasília
Faculdade de Tecnologia

Análise de Parâmetros de Movimentação e da Influência do atrito em Robôs Cobra

Mariana Mendanha da Cruz

Projeto Final de Curso submetido como requisito parcial para obtenção do grau de Engenheiro de Controle e Automação

Trabalho aprovado. Brasília, 20 de novembro de 2023:

Prof. Dr. Carla Maria Chagas e Cavalcante Koike, UnB/CIC
Orientador

Prof. Dr. Dianne Magalhães Viana, UnB/FT/ENM
Coorientador

Prof. Dr. Marcus Vinícius Lamar, UnB/CIC
Examinador interno

Prof. Dr. Flávio de Barros Vidal, UnB/CIC
Examinador interno

Brasília
2023

Dedico este trabalho à minha família e mentores, cujo apoio incansável e sabedoria foram meu farol de inspiração e motivação ao longo desta jornada.

Resumo

A intersecção entre a biologia e a tecnologia tem inspirado inovações notáveis, particularmente no campo da robótica. Este trabalho foca na locomoção de robôs-serpente, que possuem a capacidade de navegar por terrenos complexos e restritos. O objetivo principal deste projeto foi investigar estratégias para aprimorar a eficiência e a estabilidade dos movimentos desses robôs, com especial atenção aos parâmetros de movimento e ao impacto do atrito anisotrópico.

Este trabalho mostra a simulação da locomoção de robôs-serpente em um ambiente ROS e Gazebo, aplicando modelos de movimento yaw-yaw, pitch-pitch, e pitch-yaw. Empregou-se otimização Bayesiana e métodos de busca exaustiva para examinar uma vasta gama de combinações dos parâmetros de curva, além de analisar o efeito dos coeficientes de atrito. As simulações foram complementadas com análises de deslocamento e velocidade.

Para o modelo pitch-yaw, selecionou-se movimentos distintos após uma análise detalhada dos dados de simulação. Este processo permitiu a identificação de padrões de movimento fundamentais para a navegação do robô, incluindo manobras retas, curvas e diagonais, essenciais para a locomoção adaptativa em diferentes cenários. A seleção desses movimentos destaca a importância da versatilidade e da capacidade de resposta do robô às mudanças ambientais.

As conclusões deste estudo ressaltam o potencial de aplicação de robôs-serpente em cenários reais, desde operações de busca e resgate até manutenção de infraestruturas complexas. A investigação aprofundada dos parâmetros de movimento e o uso criterioso de técnicas de otimização contribuem para o desenvolvimento de robôs mais eficientes e versáteis. Este trabalho não apenas fornece um caminho para futuras pesquisas na área mas também estabelece uma base sólida para a exploração prática dessas tecnologias inovadoras.

Palavras-chave: Robótica Bioinspirada. Robô Modular. Locomoção de Robôs Cobra. Análise de Parâmetros. Atrito Anisotrópico.

Abstract

The intersection between biology and technology has inspired remarkable innovations, particularly in the field of robotics. This work focuses on optimizing the locomotion of snake robots, which have the ability to navigate through complex and confined terrains. The main goal of this project was to investigate strategies to enhance the efficiency and stability of these robots' movements, with special attention to the motion parameters and the impact of anisotropic friction.

Using a rigorous methodological approach, the locomotion of snake robots was simulated in a ROS and Gazebo environment, applying yaw-yaw, pitch-pitch, and pitch-yaw motion models. Bayesian optimization and brute force methods were employed to examine a wide range of curve parameter combinations, as well as to analyze the effect of friction coefficients. The simulations were complemented with displacement analyses and speed.

For the pitch-yaw model, distinct movements were selected after a detailed analysis of the simulation data. This process allowed the identification of fundamental movement patterns for the robot's navigation, including straight, curved, and diagonal maneuvers, essential for adaptive locomotion in different scenarios. The careful selection of these movements highlights the importance of the robot's versatility and responsive capability to environmental changes.

The conclusions of this study underscore the potential application of snake robots in real-world scenarios, from search and rescue operations to the maintenance of complex infrastructures. In-depth investigation of movement parameters and the judicious use of optimization techniques contribute to the development of more efficient and versatile robots. This work not only provides a pathway for future research in the field but also establishes a solid foundation for the practical exploration of these innovative technologies.

Keywords: Bioinspired Robotics. Modular Robot. Snake Robot Locomotion. Parameter Analysis. Anisotropic Friction.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

Lista de símbolos

Símbolos romanos

Símbolos gregos

Sumário

1 Introdução

1.1 Contextualização em Robótica Bioinspirada

No cenário atual de avanços rápidos em robótica, os robôs-serpente emergem como uma solução no campo da robótica bioinspirada, oferecendo possibilidades únicas em ambientes desafiadores e inacessíveis. Combinando simulação e técnicas de otimização Bayesiana, este projeto foca na análise da locomoção destes robôs, com análises do impacto do atrito anisotrópico.

A inspiração para tais robôs vem da biologia das serpentes, cuja locomoção ápode oferece informações interessantes para o desenvolvimento de sistemas robóticos capazes de se locomover em terrenos irregulares e restritos. Estes robôs, com sua capacidade de se contorcer e adaptar a diferentes superfícies, são ideais para tarefas como busca e resgate em locais de difícil acesso, inspeção de infraestruturas e até exploração espacial (Hopkins; Spranklin; Gupta, 2009).

A robótica modular, um conceito central na construção de robôs-serpente, permite a criação de sistemas reconfiguráveis que se adaptam a diferentes tarefas e ambientes. Esta flexibilidade, aliada à capacidade de alterar sua forma física, permite várias possibilidades para o design e a funcionalidade desses robôs (Liu; Tong; Liu, 2021).

Um desafio significativo na construção de robôs-serpente é a reprodução das propriedades anisotrópicas do atrito encontradas nas escamas das serpentes reais. Essas propriedades são cruciais para a eficácia do movimento ondular, onde o atrito age de maneira diferenciada ao longo de várias direções do corpo do robô (Hirose; Yamada, 2009). Além disso, a definição da trajetória de movimento é fundamental, e a curva serpenoide, conforme proposta por Hirose, oferece um modelo matemático que tem sido amplamente adotado na robótica ápode.

1.2 Objetivos

Este projeto tem como objetivo principal investigar a dinâmica de movimento dos robôs-serpente, utilizando uma abordagem que combina simulação e otimização Bayesiana. O foco está em compreender como os parâmetros da curva serpenoide, e as condições de atrito anisotrópico afetam o deslocamento.

1.2.1 Implementação de Modelos de Movimento

Um dos objetivos é implementar modelos de robôs-serpente em ambiente de simulação capazes de realizar os movimentos yaw-yaw, pitch-pitch, e a combinação pitch-yaw. Cada um destes estilos de movimento apresenta desafios e características únicos, essenciais para a compreensão da adaptabilidade dos robôs em diferentes ambientes.

1.2.2 Avaliação de Desempenho em Simulações

Outra meta importante é a avaliação do desempenho desses modelos em simulações computacionais. Estas simulações focam na análise do impacto dos parâmetros da curva de Hirose e do atrito anisotrópico na locomoção do robô planar. Também são avaliados os parâmetros da curva composta para robôs de ligações Pitch-Yaw, sendo cruciais para identificar ajustes e melhorias na otimização do movimento, ainda no caso do Pitch-Yaw, é desejado encontrar parâmetros para realizar diferentes deslocamentos, como retilíneo, lateral e curvo.

1.2.3 Avaliação de Desempenho em Simulações

No contexto deste trabalho, o desempenho é definido pela eficiência em realizar deslocamentos específicos no menor espaço de tempo. Nos modelos planares (yaw-yaw e pitch-pitch), a avaliação se concentra no deslocamento retilíneo no eixo x. Para o modelo pitch-yaw, considera-se o desempenho nos deslocamentos retilíneo, lateral e curvo, para os três modelos, buscando a estabilidade do movimento, isto é, evitar tombos ou movimentos descontrolados, que podem vir a danificar a estrutura do robô.

A avaliação de desempenho envolve simulações computacionais focadas nos efeitos dos parâmetros da curva de Hirose e do atrito anisotrópico na locomoção dos modelos planares. Os parâmetros da curva composta para robôs com ligações Pitch-Yaw também são analisados para identificar necessidades de ajuste e melhoria na otimização do movimento. O objetivo com o modelo Pitch-Yaw é definir parâmetros para diferentes tipos de deslocamento, que incluem os movimentos retilíneo, lateral e curvo.

1.2.4 Otimização Bayesiana dos Parâmetros

Por fim, o projeto visa aplicar técnicas de otimização Bayesiana para aperfeiçoar a escolha dos conjuntos de parâmetros. Este processo de otimização busca não apenas a eficiência, mas também melhorar a adaptabilidade dos robôs para navegação bem-sucedida em diversos contextos.

Em resumo, os objetivos do projeto são:

- Implementar modelos de robôs-serpente com movimentos yaw-yaw, pitch-pitch e pitch-yaw.;
- Avaliar o desempenho desses modelos em simulações, focando nos efeitos dos parâmetros de curva e do atrito anisotrópico.; e
- Utilizar a otimização Bayesiana para identificar os conjuntos de parâmetros mais eficazes para melhorar a locomoção dos robôs;

1.3 Organização do Relatório

Este relatório está estruturado em cinco capítulos principais, complementados por seções específicas que detalham cada fase do projeto.

A introdução no Capítulo 1 contextualiza o campo da robótica bioinspirada e define os objetivos do estudo.

Já a revisão bibliográfica e teórica no capítulo 2 apresenta uma revisão detalhada sobre locomoção de serpentes e princípios da robótica bioinspirada, abordando também os fundamentos matemáticos e práticos do método Bayesiano de aprendizado.

O Capítulo 3 descreve os modelos robóticos Unified Robot Description Format (URDF) para os movimentos yaw-yaw, pitch-pitch e o modelo ondulatório composto pitch-yaw, e discute a metodologia para simulação e controle dos modelos.

No Capítulo 4, as análises e resultados são divididos em seções que discutem os resultados obtidos para cada modelo de movimento, incluindo a análise dos parâmetros de curva e o efeito do atrito anisotrópico.

O Capítulo 5 sintetiza os resultados, discute as implicações para aplicações em ambientes reais e sugere direções para futuras pesquisas.

2 Revisão Bibliográfica e Teórica

Este capítulo contém uma revisão bibliográfica e teórica focada na locomoção de serpentes e nos princípios da robótica bioinspirada. Também inclui uma introdução dos fundamentos matemáticos e práticos do método Bayesiano de aprendizado, que foi o método escolhido para otimizar a escolha dos parâmetros de curva do movimento do robô modular.

Na Seção 2.1, detalhou-se os diferentes modos de locomoção de serpentes: serpentina, concertina, rastejamento retilíneo e sidewinding. Essa análise serve de base para a compreensão dos movimentos que inspiram o design de robôs-serpente.

A Seção 2.2 aborda os princípios de robótica bioinspirada, trazendo exemplos de aplicações para diferentes estruturas biológicas, essenciais para o desenvolvimento de robôs baseados em serpentes.

A Seção 2.3 e 2.4 discutem a locomoção Serpenoide plana e composta, respectivamente, focando nos fundamentos matemáticos, equações e parâmetros relevantes.

A Seção 2.5 explora a modelagem e simulação de robôs-serpente, importantes para testar teorias e validar designs. Diferentes modelos de módulos de robôs-serpente são apresentados, além de suas características e limitações. É apresentado também uma breve descrição dos métodos e ferramentas utilizadas para as simulações dos modelos.

Finalmente, a Seção 2.6 introduz o método Bayesiano de aprendizado, começando com uma exploração do Teorema de Bayes e sua relevância na atualização de probabilidades em resposta a novas informações. Esta seção também detalha a implementação prática desse método na função `gpminimize` do Python, parte da biblioteca Scikit-Optimize. O foco será em como esse método é aplicado na otimização de parâmetros para robôs cobra, particularmente na melhoria da locomoção e estabilidade.

Cada seção desse capítulo contribui para a compreensão dos mecanismos de locomoção das serpentes e sua aplicação na robótica, estabelecendo a base para os desenvolvimentos seguintes neste trabalho.

2.1 Locomoção de Serpentes

A locomoção das serpentes é um fenômeno fascinante e complexo, com várias estratégias adaptadas ao seu ambiente e necessidades (Yang *et al.*, 2022). Esta seção explora os principais tipos de movimento das serpentes, fundamentais para o desenvolvimento da robótica bioinspirada.

2.1.1 Locomoção Serpentina

A ondulação lateral é o tipo de locomoção mais comum em serpentes. Caracteriza-se pela propagação de ondas musculares ao longo do corpo da serpente, da cabeça até a cauda, permitindo que a serpente se mova em linha reta empurrando contra irregularidades no terreno. Este movimento é eficiente em uma variedade de superfícies, desde solo até água, e é marcado pela ausência de pontos estáticos durante o contato do corpo com o solo (Socha, 2002). Esse tipo de movimento pode ser visto na Figura 2.1.

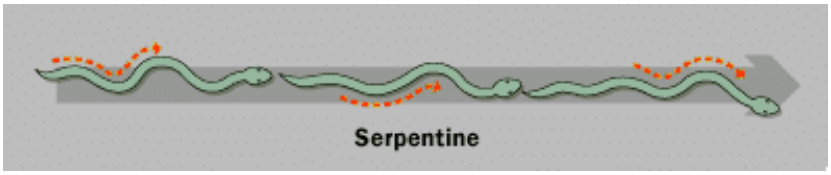

Figura 2.1 – Movimento Serpentina

Fonte: (Prada *et al.*, 2012)

2.1.2 Locomoção Concertina

A locomoção concertina é utilizada em ambientes restritos, como túneis ou buracos, onde a ondulação lateral não é eficaz. Este movimento envolve a alternância entre a extensão do corpo e o encurvamento para ancoragem. A cobra estende a parte frontal do corpo, enquanto a parte traseira se encurva e agarra ao terreno, proporcionando a força necessária para empurrar o corpo para frente (Smith; Johnson, 2023). Esse tipo de movimento pode ser visto na Figura 2.2.

Figura 2.2 – Movimento Concertina

Fonte: (Prada *et al.*, 2012)

2.1.3 Rastejamento Retilíneo

O rastejamento retilíneo é um método de locomoção lento e menos comum, geralmente utilizado por serpentes grandes ou quando se aproximam de uma presa. Neste movimento, a serpente utiliza suas escamas ventrais como pontos de ancoragem para se projetar para a frente. O processo envolve uma sequência de extensões e contrações segmentadas ao longo do corpo, permitindo um avanço cuidadoso e silencioso (Miller; Thompson, 2023). Esse tipo de movimento pode ser visto na Figura 2.3.

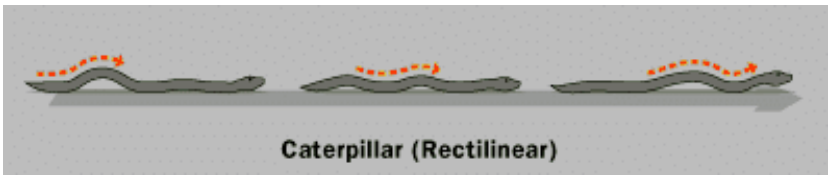

Figura 2.3 – Movimento Retilíneo

Fonte: (Prada *et al.*, 2012)

2.1.4 *Sidewinding*

O sidewinding é adaptado para locomoção em ambientes arenosos, como desertos. Neste método, a serpente levanta e move seções do corpo para a frente, enquanto outras seções permanecem ancoradas no solo. Este movimento cria um padrão diagonal distinto nas marcas deixadas na areia e permite que a serpente se mova eficazmente em superfícies soltas e escorregadias (Davis; Walters, 2023). Esse tipo de movimento pode ser visto na Figura 2.4.

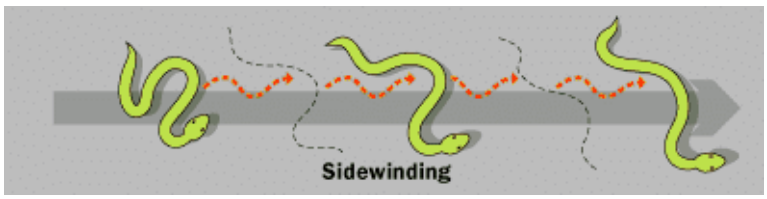

Figura 2.4 – Movimento Sidewinding

Fonte: (Prada *et al.*, 2012)

2.2 Princípios de Robótica Bioinspirada

A robótica bioinspirada representa um campo promissor e inovador que se baseia na observação e na emulação de sistemas biológicos para o desenvolvimento de robôs. Este campo é motivado pela compreensão de que os organismos vivos evoluíram ao longo de milhões de anos para se tornarem mestres da adaptação, capazes de realizar tarefas complexas de maneira eficiente.

As serpentes, com sua habilidade de navegar por uma variedade de terrenos desafiadores, servem como uma inspiração particular no design de robôs-cobra. Suas técnicas de locomoção, que incluem ondulação lateral, movimento concertina, rastejamento retilíneo e sidewinding, são estudadas e replicadas para criar robôs capazes de realizar busca e resgate, inspeção de infraestrutura e exploração em ambientes que seriam inacessíveis ou perigosos para humanos (Aupperlee, 2021).

Uma dessas funções pode ser observada na figura 2.5, onde um robô cobra subaquático é demonstrado navegando por complexas estruturas de tubulação submersa, uma tarefa que destaca a sua capacidade de inspecionar e manter infraestruturas em condições que seriam desafiadoras ou impossíveis para humanos ou robôs de forma mais convencional.

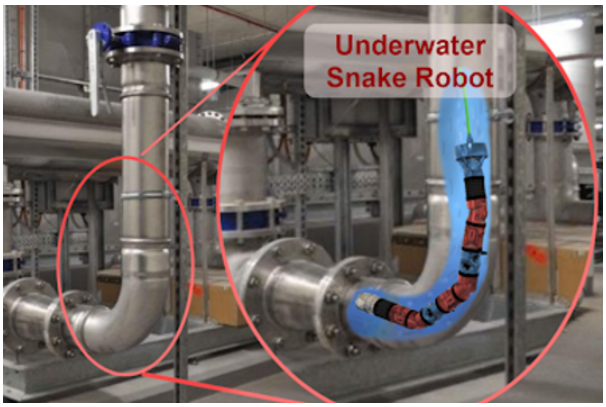

Figura 2.5 – Robô cobra em operação

Fonte: https://www.gordonrobertson.online/humrs-project-management

Anisotropia é uma propriedade fundamental de certas estruturas, onde suas características físicas variam dependendo da direção em que são medidas. Isso significa que a resposta de uma estrutura anisotrópica, como a resistência ou a fricção, é diferente ao longo de diferentes eixos. Estruturas anisotrópicas biológicas, como as escamas de uma cobra, são exemplos ilustres, onde a fricção diferenciada permite movimentos eficientes em várias superfícies.

A implementação de princípios bioinspirados em robótica, como demonstrado pela variedade de estruturas anisotrópicas biológicas na Figura 2.6, representa um desafio complexo. A habilidade de compreender e abstrair as propriedades únicas dessas estruturas

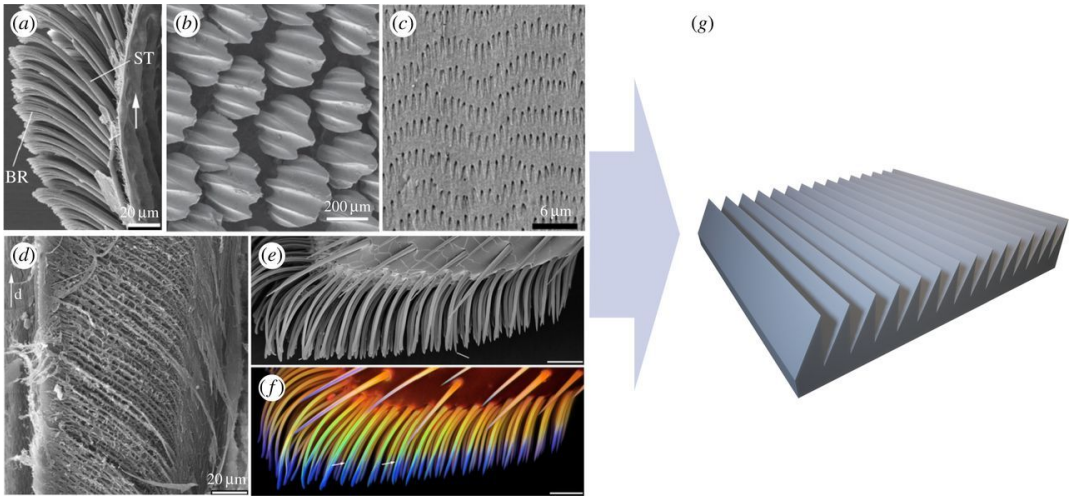

Figura 2.6 – Exemplos de estruturas anisotrópicas biológicas: (a) estrutura de um lagarto (Gekko), (b) dentículos de pele de tubarão (Lamna nasus), (c) microestrutura da pele de cobra (Lampropeltis getula californiae), (d) ultraestrutura da almofada de fixação de Tettigonia viridissima, (e) almofada adesiva da pata dianteira de uma joaninha (Coccinella septempunctata), (f) varredura a laser confocal mostrando um gradiente, (g) estrutura de amostra bioinspirada semelhante a uma serra.

Fonte: (Tramsen *et al.*, 2018)

em modelos matemáticos e algoritmos computacionais é essencial. O desenvolvimento de materiais e sistemas robóticos que emulam essas características, como a anisotropia no atrito encontrada em diversas formas biológicas, ilustra a amplitude e a profundidade dos desafios enfrentados pelos engenheiros na área.

No entanto, os avanços na robótica bioinspirada têm levado a novas aplicações práticas. Na medicina, por exemplo, robôs inspirados em serpentes estão sendo desenvolvidos para cirurgias minimamente invasivas, aproveitando sua capacidade de navegar em espaços confinados dentro do corpo humano. Em operações de busca e resgate, tais robôs oferecem potencial para acessar áreas afetadas por desastres naturais onde o acesso é restrito ou perigoso. A implementação desses princípios bioinspirados está expandindo o escopo da robótica em campos que exigem alta precisão e adaptabilidade.

A Tabela 2.1 apresenta um mapeamento entre estruturas biológicas e suas correspondentes aplicações na robótica. A adesão em superfícies verticais, inspirada nas patas do gecko, tem sido replicada em adesivos robóticos para escalada de paredes[1]. Revestimentos robóticos que reduzem o arrasto em fluidos têm sido desenvolvidos com base na pele do tubarão[2]. Para a locomoção em diversos terrenos, sistemas robóticos têm emulado a pele de cobra[3]. A sensibilidade tátil e a manipulação em robótica têm sido aprimoradas por meio de almofadas adesivas inspiradas em insetos[4]. Materiais com propriedades de mudança de cor, semelhantes às asas de borboleta, estão sendo explorados para camuflagem em robótica[5]. Por fim, o sistema musculoesquelético humano tem sido um modelo para o

desenvolvimento de robôs humanoides e próteses avançadas[6].

Tabela 2.1 – Exemplos de Aplicações da Robótica Bioinspirada

Estrutura Biológica	**Aplicação em Robótica**
Patas de Gecko	Adesivos robóticos
Pele de Tubarão	Revestimentos de robôs
Pele de Cobra	Sistemas de locomoção
Almofada Adesiva de Insetos	Sensores táteis e manipuladores robóticos
Asas de Borboleta	Materiais com mudança de cor
Sistema Musculoesquelético Humano	Robôs humanoides e próteses

Com o avanço contínuo da tecnologia de sensores, atuadores e inteligência artificial, o futuro da robótica bioinspirada promete inovações que poderão revolucionar a maneira como interagimos e utilizamos robôs em nossa sociedade. O compromisso com a pesquisa e desenvolvimento contínuos é essencial para superar os desafios existentes e para desbloquear o potencial completo desta disciplina.

2.3 Locomoção Serpenoide Plana

2.3.1 Fundamentos e Equações

Robôs ápodes são desprovidas de membros externos para locomoção, característica que os aproxima morfologicamente das serpentes. A compreensão da locomoção robótica Serpenoide passa pelo estudo das curvas serpenoides (Hirose; Yamada, 2009).

Estas curvas, que fornecem a base matemática para replicar o movimento sinuoso das serpentes, são fundamentais no design e controle de robôs ápodes, permitindo movimentos eficientes em diversos tipos de terreno sem a necessidade de membros tradicionais.

A curva serpenoide é matematicamente descrita pela angulação de cada segmento do robô em relação ao seu eixo longitudinal. Onde α_s é o ângulo de referência que descreve esta angulação, α_0 é o ângulo inicial, k é o número de ondulações, l é o comprimento total do robô, e S é a posição ao longo do corpo do robô. A curva pode ser visualizada destacando o ângulo de referência e o comprimento do robô na Figura 2.7.

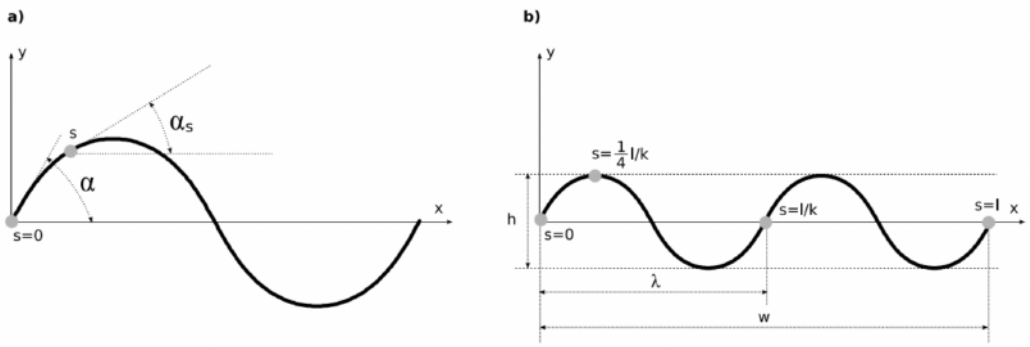

Figura 2.7 – Representação da curva serpenoide: a) Definição do ângulo de referência α_s. b)Comprimento l do corpo da cobra.

Fonte: (Ereko, 2015)

Para o modelo de conexão pitch-pitch, o ângulo de cada segmento é articulado em um plano vertical, enquanto no yaw-yaw, a articulação ocorre em um plano horizontal. A relação entre a posição do segmento S e o ângulo de referência α_s é governada pela seguinte equação:

$$\alpha_s = \alpha_0 \cos\left(\frac{2\pi k}{l} S\right), \tag{2.1}$$

Esta equação não só define a forma da curva serpenoide, mas também direciona o desenvolvimento do algoritmo de controle para a locomoção nos referidos modelos robóticos.

A posição em coordenadas cartesianas de cada ponto na curva pode ser obtida por

integração das componentes de α_s, resultando em:

$$x(S) = \int_0^S \cos(\alpha_s) dS, \tag{2.2}$$

$$y(S) = \int_0^S \sin(\alpha_s) dS. \tag{2.3}$$

No entanto, para um robô com um número finito de segmentos, a curva precisa ser discretizada. A posição do i-ésimo segmento é aproximada por somas parciais como:

$$x_i = \sum_{k=1}^{i} \frac{1}{M} \cos\left(a \cos\left(\frac{kb}{M}\right) + \frac{kc}{M}\right), e \tag{2.4}$$

$$y_i = \sum_{k=1}^{i} \frac{1}{M} \sin\left(a \cos\left(\frac{kb}{M}\right) + \frac{kc}{M}\right), \tag{2.5}$$

onde $\boldsymbol{a}$, $\boldsymbol{b}$ e $\boldsymbol{c}$ são os parâmetros da curva serpenoide que influenciam respectivamente o enrolamento da curva (*winding angle*), o número de períodos ao longo do corpo (*joint actuation angular frequency*) e a direção do movimento (*heading*).

Na Figura 2.8 é possível visualizar a modulação da amplitude da curva com base no parâmetro $\boldsymbol{a}$, a variação da frequência de ondas ao longo do corpo do robô conforme o parâmetro $\boldsymbol{b}$, e a orientação geral da trajetória influenciada pelo parâmetro $\boldsymbol{c}$, em que para cada gráfico é variado apenas um parâmetro ($\boldsymbol{a}$, $\boldsymbol{b}$ e $\boldsymbol{c}$, respectivamente) como demonstrado na legenda enquanto os outros parâmetros se mantém fixos.

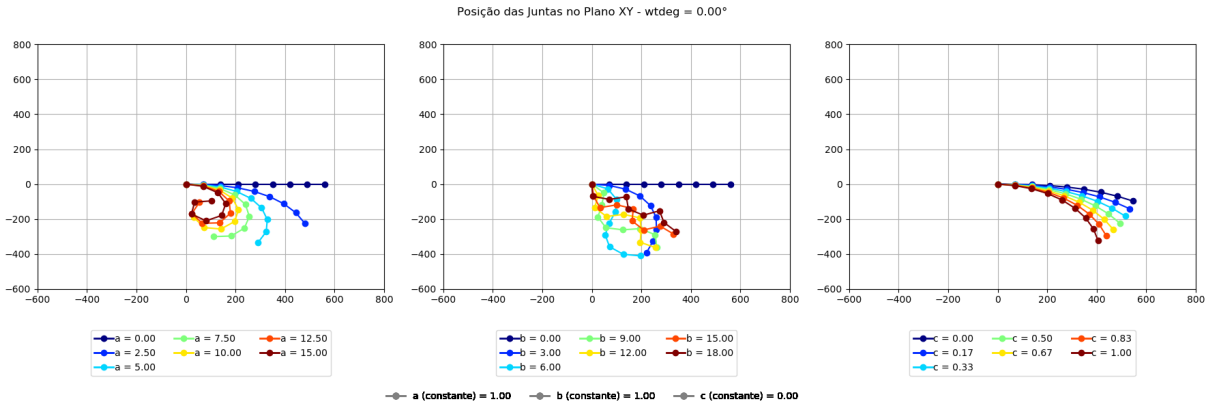

Figura 2.8 – Influência dos parâmetros $\boldsymbol{a}$, $\boldsymbol{b}$, e $\boldsymbol{c}$, respectivamente, na configuração da curva serpenoide (configuração das juntas).

Fonte: A Autora.

2.3.2 **Parâmetro a** - Ângulo de Enrolamento

O parâmetro a representa o ângulo de enrolamento na modelagem do movimento. Este parâmetro determina a amplitude dos movimentos angulares das juntas, influenciando diretamente a forma da onda gerada ao longo do corpo do robô durante a locomoção. Um

valor maior de a resulta em uma onda com maior amplitude, enquanto um valor menor produz ondas mais suaves e movimentos menos pronunciados das juntas. Podemos ver essa influência na Figura 2.9.

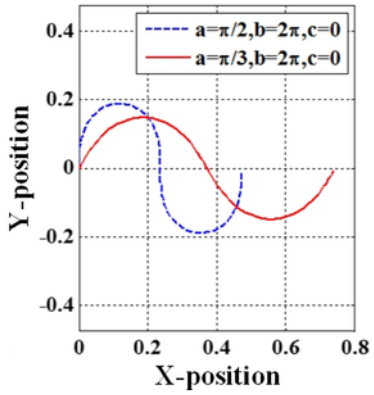

Figura 2.9 – Influência do parâmetro a na curva serpenoide.

Fonte:(Qiao *et al.*, 2017)

A mesma influência pode ser visualizada propagando-se pelas juntas do robô na Figura 2.10. Em que temos gráficos para cada junta de um robô de 10 módulos com os eixos parâmetro $\boldsymbol{a}$, ângulo da junta e a frequência angular (wt, que pode ser abstraída como variável de tempo) em graus. Fazendo um corte perpendicular ao plano wt, ou seja, em um tempo fixo, percebemos que quanto maior o valor do parâmetro, maior será a oscilação do ângulo da junta, isso para todas as juntas, sem exceção. Portanto, o parâmetro $\boldsymbol{a}$ está ligado à amplitude do movimento, como foi dito anteriormente.

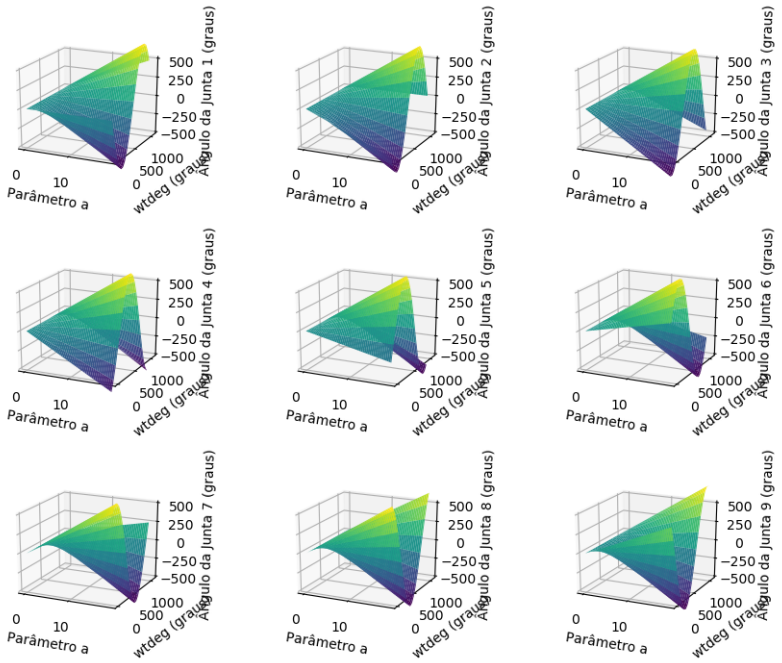

Figura 2.10 – Influência do parâmetro a no ângulo de 9 juntas pelo tempo.

Fonte: A Autora.

2.3.3 **Parâmetro b** - Número de Períodos ao Longo do Corpo

O parâmetro *b* define o número de ondulações ao longo do corpo do robô, afetando a frequência das ondas que se propagam ao longo do seu corpo. Valores maiores de *b* aumentam o número de ondulações, o que pode ser benéfico para a propulsão em certos tipos de terreno. Valores menores de *b*, por outro lado, resultam em menos ondulações e podem favorecer uma locomoção mais direta e controlada. Podemos ver essa influência na Figura 2.11.

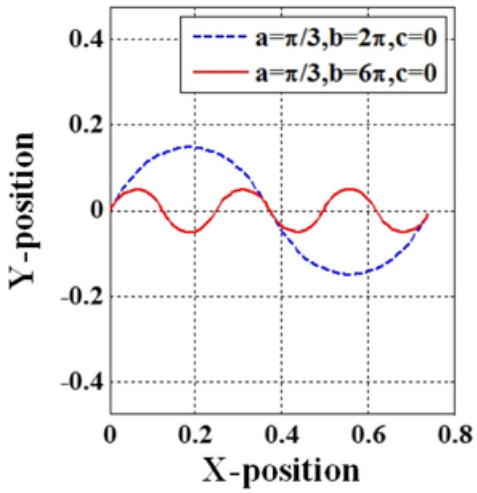

Figura 2.11 – Influência do parâmetro *b* na curva serpenoide.

Fonte:(Qiao *et al.*, 2017)

A mesma influência pode ser visualizada propagando-se pelas juntas do robô na Figura 2.12. Em que temos gráficos para cada junta de um robô de 10 módulos com os eixos parâmetro ***b***, ângulo da junta e a frequência angular (wt, que pode ser abstraída como variável de tempo) em graus. Observando a mudança entre os gráficos de cada junta, podemos observar que a junta mais distante, possui maior o número de ondulações, e também, quanto maior o valor do parâmetro ***b***, a frequência das ondas de propagação aumenta, como foi dito anteriormente.

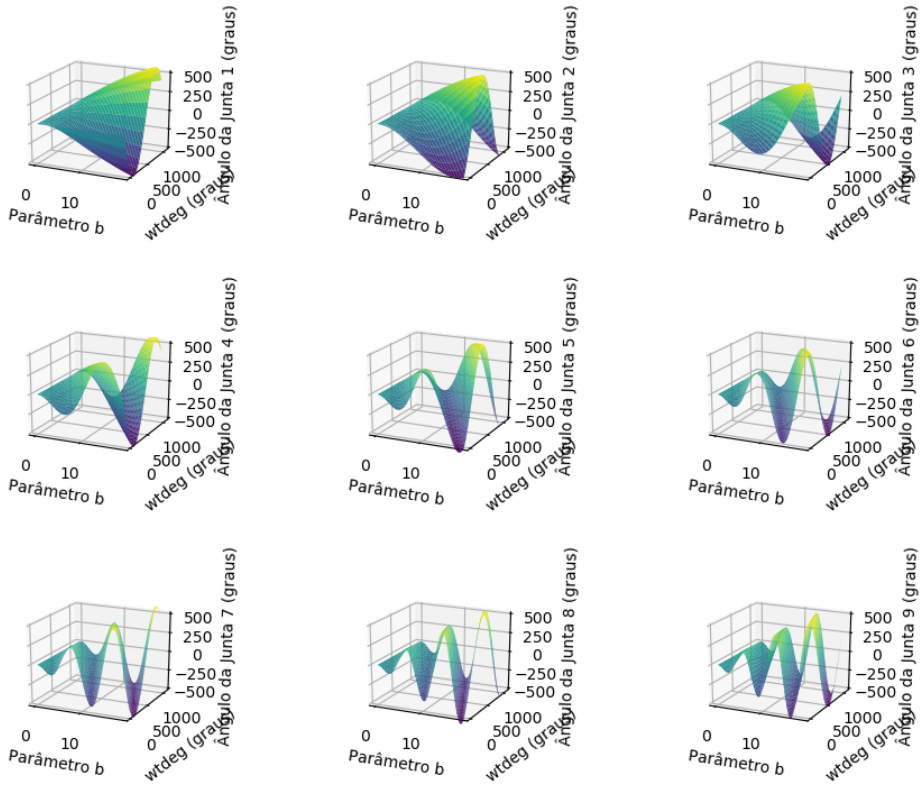

Figura 2.12 – Influência do parâmetro *b* no ângulo de 9 juntas pelo tempo.

Fonte: A Autora.

2.3.4 **Parâmetro c** - Direção do Movimento do Robô

O parâmetro *c* controla a direção do movimento do robô, funcionando como um offset que modifica o eixo de simetria da onda serpenoide. Isso altera a direção em que o robô se move, permitindo manobras precisas e ajustes na trajetória para navegação em ambientes complexos ou para evitar obstáculos. Podemos ver essa influência na Figura 2.13.

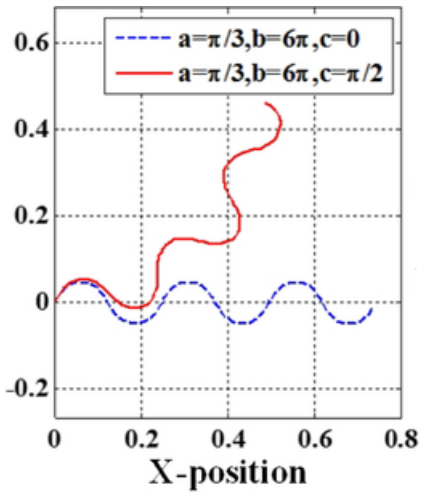

Figura 2.13 – Influência do parâmetro *c* na curva serpenoide.

Fonte:(Qiao *et al.*, 2017)

A mesma influência pode ser visualizada propagando-se pelas juntas do robô na Figura 2.14. Em que temos gráficos para cada junta de um robô de 10 módulos com os eixos parâmetro $\boldsymbol{c}$, ângulo da junta e a frequência angular (wt, que pode ser abstraída como variável de tempo) em graus. Fazendo vários cortes perpendiculares ao plano $\boldsymbol{c}$ é possível perceber que a curva se desloca verticalmente, mas mantém a frequência e a amplitude fixas.

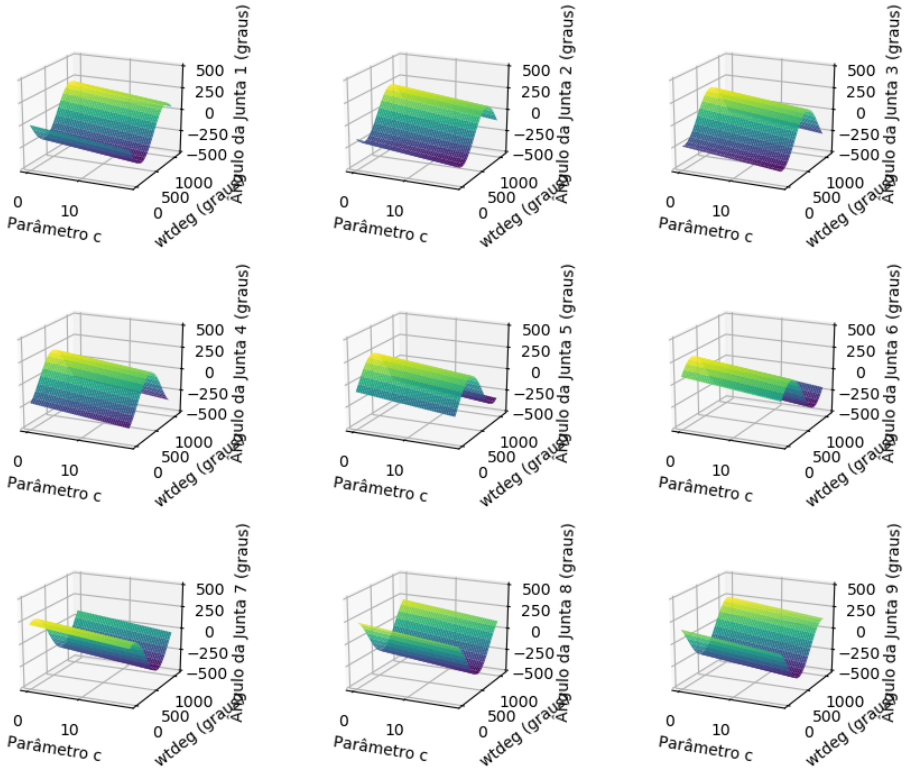

Figura 2.14 – Influência do parâmetro c no ângulo de 9 juntas pelo tempo.

Fonte: A Autora.

2.3.5 Atrito Anisotrópico

As escamas das cobras criam um coeficiente de atrito maior na direção transversal do corpo em comparação com a direção tangencial. Esta diferença é vital para o movimento de deslizamento para frente como demonstrado em (Liljebäck *et al.*, 2013).

(Liljebäck *et al.*, 2013) diz que um robô-cobra não é controlável quando o atrito do solo é isotrópico e um robô-cobra se torna fortemente acessível quando o atrito do solo é anisotrópico. A análise também conclui que os ângulos das articulações do robô-cobra devem estar desfasados durante a locomoção para aproveitar a anisotropia do atrito e gerar propulsão.

No movimento das cobras, o atrito anisotrópico é evidente pela maneira como elas interagem com diferentes superfícies. As escamas ventrais das cobras exibem propriedades de atrito diferenciadas ao longo de diferentes eixos, o que facilita o movimento para frente

enquanto minimiza o deslizamento lateral. Este fenômeno é semelhante ao efeito obtido pelo ajuste dos coeficientes $\mu 1$ e $\mu 2$ em simulações robóticas (Gazebo Tutorials, 2023), onde valores diferenciados podem replicar a eficiência do movimento serpenteante.

Os parâmetros de atrito $\mu 1$ e $\mu 2$ desempenham um papel crucial na modelagem da interação entre um objeto, como um robô, e as superfícies com as quais ele entra em contato. Eles são particularmente importantes em simulações robóticas, influenciando o movimento e a interação do robô com seu ambiente (Liljebäck *et al.*, 2013).

2.3.5.1 Coeficiente de Atrito Estático - $\mu 1$

O parâmetro $\mu 1$ representa o coeficiente de atrito estático. Ele determina a força de atrito que deve ser superada para iniciar o movimento de um objeto em repouso. Um valor mais alto de $\mu 1$ indica que uma maior força é necessária para começar a mover o objeto. Para um robô, isso significa que seria necessário mais esforço para iniciar o movimento a partir de uma posição estática. Esse coeficiente é crucial em cenários onde a inicialização do movimento é um desafio, como em superfícies inclinadas ou escorregadias (Hibbeler, 2010).

2.3.5.2 Coeficiente de Atrito Dinâmico - $\mu 2$

O parâmetro $\mu 2$ é o coeficiente de atrito dinâmico ou cinético, relacionado à força de atrito que atua em um objeto já em movimento. Um valor mais alto de $\mu 2$ significa que mais força é necessária para manter o objeto em movimento. No contexto de um robô, isso se traduz em uma maior necessidade de energia ou potência para manter ou ajustar seu movimento. Este coeficiente é importante para modelar a manutenção do movimento do robô, afetando a eficiência energética e a capacidade de manobra em diferentes superfícies (Hibbeler, 2010).

2.3.5.3 Aplicação em Robôs Cobra

Ao modelar robôs cobra, os coeficientes de atrito $\mu 1$ e $\mu 2$ podem ser ajustados para criar um efeito análogo. Ajustar $\mu 1$ e $\mu 2$ para simular o atrito anisotrópico pode melhorar a capacidade do robô de se mover eficientemente (Li Y.; Yu, 2017), replicando a locomoção natural de uma cobra real. Isso envolve buscar coeficientes que permitem o movimento fácil em uma direção (para frente) enquanto se opõe ao movimento em outras direções (lateralmente), melhorando a aderência e a propulsão.

2.3.6 Corolário da Fundamentação Matemática

A sincronização do movimento dos segmentos é realizada através da aplicação de uma onda senoidal com uma defasagem regular, proporcionando um movimento coordenado que simula o movimento serpenteante. A relação entre o ângulo de inclinação de cada junta

e o tempo proposta por (Gómez, 2008) pode ser expressa como:

$$\phi_i(t) = -2a \sin\left(\frac{b}{2M}\right) \sin\left(\omega t + \frac{ib}{M}\right) - \frac{c}{M}, \tag{2.6}$$

onde ω é a frequência angular do movimento ondulatório, M é o número de juntas, i é o índice da junta à ser calculada, a, b e c são os parâmetros da curva serpenoide que influenciam respectivamente o enrolamento da curva (winding angle), o número de períodos ao longo do corpo (joint actuation angular frequency) e a direção do movimento (heading).

Com a teoria matemática estabelecida, é possível simular a locomoção de um robô Serpenoide, aproveitando a sincronia dos movimentos segmentados e a precisão do controle de atuação. As simulações são fundamentais para verificar a viabilidade dos modelos matemáticos e para ajustar os parâmetros que resultarão em uma locomoção eficiente e realista.

2.4 Locomoção Serpenoide Composta

A locomoção Serpenoide composta (Rollinson, 2014), expande o modelo clássico da curva serpenoide para capturar melhor a dinâmica de movimento das serpentes biológicas. Esta abordagem reconhece a habilidade das serpentes em manipular o contato com o ambiente, elevando e abaixando partes de seus corpos para criar padrões de movimento mais complexos.

Especificamente, no movimento sidewinding, as serpentes exibem uma combinação única de elevação lateral e propulsão, o que permite a locomoção eficiente em superfícies como areia solta, onde outros métodos de movimento seriam menos eficazes. Adicionando um segundo plano à curva serpenoide cria-se a possibilidade da implementação de movimentos semelhantes ao sidewiding em cobras robô, possibilitando a navegação em terrenos anteriormente inacessíveis e aumentando a versatilidade destes sistemas robóticos.

2.4.1 Curva Serpenoide Composta

A locomoção Serpenoide composta expande a abordagem tradicional da curva serpenoide, introduzindo maior flexibilidade e adaptabilidade no movimento dos robôs Serpenoides. Essa abordagem é crucial para simular com mais precisão os movimentos complexos de serpentes reais.

A curva serpenoide composta é uma formulação matemática que representa o movimento de serpentes robóticas através de expressões sinusoidais separadas para ondas horizontais (movimento lateral) e verticais (movimento dorsal). A equação para o ângulo da junta θ em um índice de junta n e tempo t é dada por:

$$\theta(n,t) = \begin{cases} \beta_h + A_h(n)\sin(\omega_h t + \nu_h n) & \text{, e} \\ \beta_v + eA_v(n)\sin(\omega_v t + \nu_v n + \delta) & \text{,} \end{cases} \tag{2.7}$$

onde β, A, ω, ν, e δ representam, respectivamente, o offset angular, amplitude, frequência temporal, frequência espacial, diferença de fase e os índices h e v significam movimento horizontal e vertical, respectivamente. Esta formulação permite a coordenação de um número arbitrário de juntas em um robô serpente (Dai *et al.*, 2015).

A posição de cada módulo do robô, em um contexto discreto, pode ser aproximada através das somas parciais das expressões angulares, de forma similar à abordagem para curvas serpenoides planas. Para o módulo i-ésimo, as posições x_i e y_i podem ser descritas como:

$$x_i = \sum_{k=1}^{i} \cos\left(\theta_h(k,t)\right), \tag{2.8}$$

$$y_i = \sum_{k=1}^{i} \sin\left(\theta_v(k,t)\right), \tag{2.9}$$

onde $\theta_h(k,t)$ e $\theta_v(k,t)$ são os ângulos calculados pelas expressões horizontais e verticais da curva serpenoide composta, respectivamente.

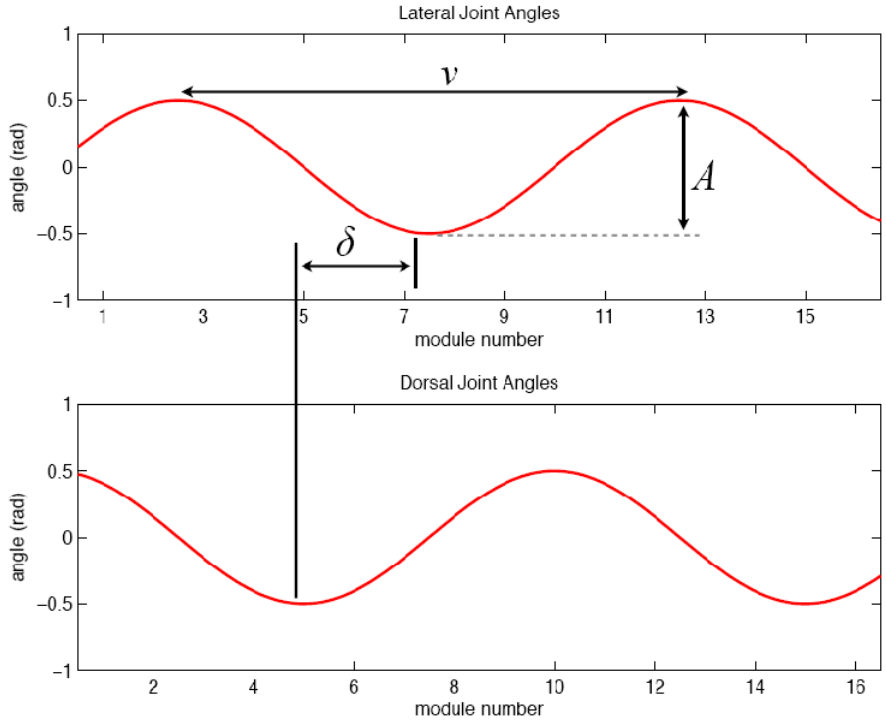

Figura 2.15 – Representação gráfica dos ângulos das juntas para locomoção Serpenoide composta. Os parâmetros mostrados correspondem à equação 4.1

Fonte: (Rollinson, 2014).

2.4.2 Parâmetro β (Offset Angular)

O parâmetro β representa um ângulo de offset ou deslocamento para as juntas. Ele pode ser usado para ajustar a postura inicial do robô ou para compensar qualquer desalinhamento nas juntas. Valores muito altos podem resultar em uma forma inicial não natural ou ineficiente para o movimento.

2.4.3 Parâmetro A (Amplitude)

O parâmetro A determina a amplitude das oscilações dos módulos do robô. A amplitude determina quão longe cada junta pode se mover de sua posição central. Uma amplitude maior resulta em movimentos mais amplos, o que pode ser útil para cobrir uma área maior, mas também pode reduzir a eficiência ou estabilidade.

2.4.4 Parâmetro δ (Diferença de Fase)

O parâmetro δ é a diferença de fase entre os movimentos horizontais e verticais na locomoção Serpenoide composta. Este parâmetro é essencial para criar um movimento coordenado e fluido, sincronizando os componentes horizontal e vertical da onda serpenoide. A diferença de fase entre as ondas laterais e dorsais é crucial para coordenar os movimentos das juntas de maneira que o robô realize movimentos específicos como retilíneo, lateral ou curvo.

2.4.5 Parâmetro ω (Frequência Temporal)

O parâmetro ω representa a frequência temporal da onda serpenoide composta. Este parâmetro determina a rapidez com que os padrões de onda se propagam ao longo do corpo do robô, afetando a velocidade e a agilidade da locomoção. Ajustes em ω podem ser usados para adaptar a velocidade do robô a diferentes condições ambientais ou tarefas.

2.4.6 Parâmetro ν (Frequência Espacial)

O parâmetro ν é responsável pela definição da frequência espacial na curva serpenoide composta, determinando a rapidez com que as juntas completam um ciclo de movimento. Com uma frequência mais elevada, as juntas movem-se mais rapidamente, o que pode influenciar a velocidade do robô, além de impactar sua estabilidade, capacidade de manobra e eficiência em diferentes tipos de superfície. Uma frequência espacial alta implica em mais ondulações ao longo do corpo do robô, favorecendo a locomoção em superfícies irregulares ou na execução de manobras complexas.

2.4.7 Desafios e Limitações

Apesar de ser um poderoso quadro de controle, a abordagem baseada em curvas serpenoides tem suas limitações. Uma delas é a dificuldade de relacionar diretamente os parâmetros com a configuração cinemática do robô. Além disso, a simplificação do controle para um sistema de baixa dimensão muitas vezes descarta a expressividade potencial das formas do robô, fazendo de seu movimento simétrico o que limita sua navegação em ambientes mais complexos.

2.5 Modelagem e Simulação de Robôs-Serpente

2.5.1 Modelos de Módulos de Robôs-Serpente

A série VDM (Variação e Design Modular) representa um avanço significativo na construção de robôs-serpente, tendo como base quase uma década de estudos de robôs modulares reconfiguráveis da equipe EREKO da Universidade de Brasília. O design destes módulos é influenciado por configurações anteriores focadas na movimentação pitch-pitch, como é evidente na similaridade estrutural com o robô Ereko que podem ser vistos na Figura 2.16.

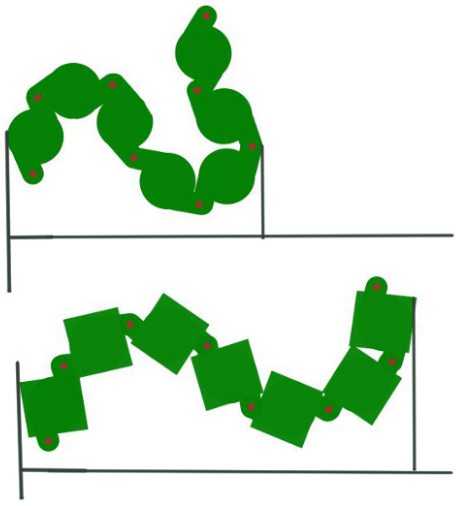

Figura 2.16 – Duas curvas com ângulos de atuação máximos em modelo VDM e modelo não arredondado.

Fonte: (Galembeck, 2018)

O módulo VDM-Y foi concebido para possibilitar a movimentação yaw-yaw, adotando uma abordagem de design que permite a inclusão de uma superfície anisotrópica (placa de atrito) para simular as propriedades de atrito encontradas na pele das serpentes o que é crucial para a reprodução de movimentações de ondulação lateral sem o uso de rodas. Este modelo pode ser complementado pelo VDM-T, destinado às transições de movimento, e

pelo VDM-C, que serve como módulo de extremidade para o robô. Estes modelos podem ser vistos na Figura 2.17

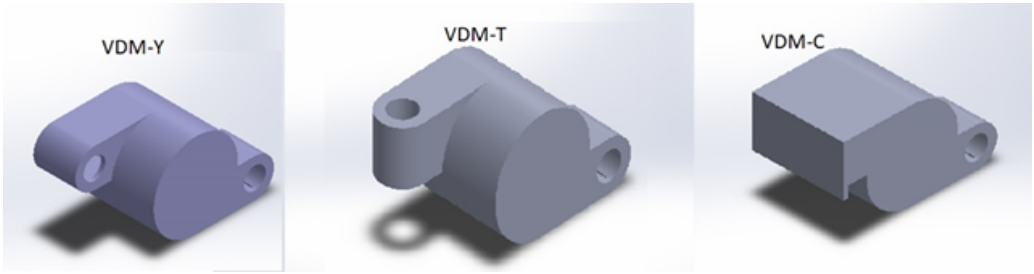

Figura 2.17 – Modelos propostos para a série VDM.

Fonte: (Galembeck, 2018)

A estrutura proposta por (Galembeck, 2018) para o protótipo VDM é uma síntese dos preceitos do NEKE, que é um design de formato cilíndrico, alongado e com extremidades arredondadas que possui alteração do sentido de atuação dos atuadores. As configurações possíveis para o robô incluem o uso exclusivo de módulos VDM-Y para movimentação lateral concertina, a combinação de VDM-Y com VDM-T para movimentação pitch-yaw, e a incorporação de módulos VDM-C para ampliar as possibilidades de análise e calibração. Uma configuração de robô composto por módulos VDM pode ser visto na Figura 2.18.

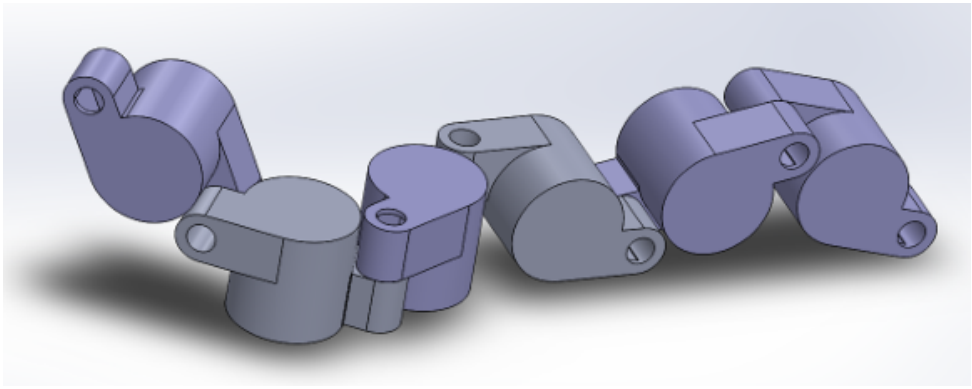

Figura 2.18 – Robô composto de módulos VDM-Y e VDM-T.

Fonte: (Galembeck, 2018)

Nesse trabalho, será utilizado apenas o módulo VDM-Y juntamente com a placa de atrito para simulações de movimento yaw-yaw e pitch-pitch, conforme ilustrado na Figura 2.19.

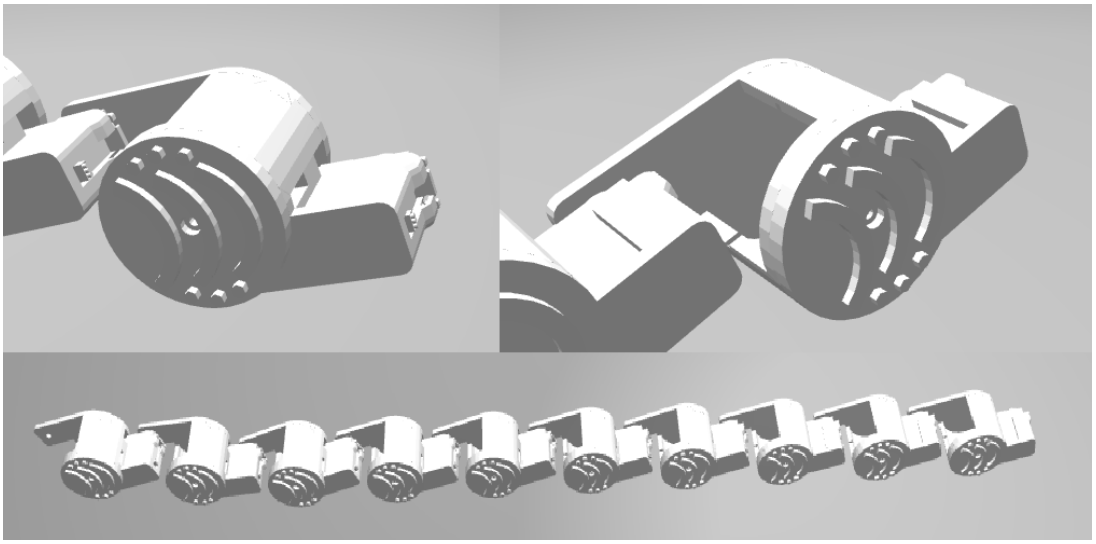

Figura 2.19 – Robô módulos VDM-Y e placa de atrito para movimentos yaw-yaw e pitch-pitch.

Fonte: A Autora.

2.5.2 Modelo ErekoChain Desenvolvido por Thiago

Desenvolvido por (Rocha, 2023), o ErekoChainBot é uma inovação nos robôs modulares do Laboratório Ereko, superando os limites de movimento dos modelos anteriores. Com módulos combinados pitch-yaw, promove uma locomoção versátil e adaptável, mais alinhada à complexidade da movimentação serpentina.

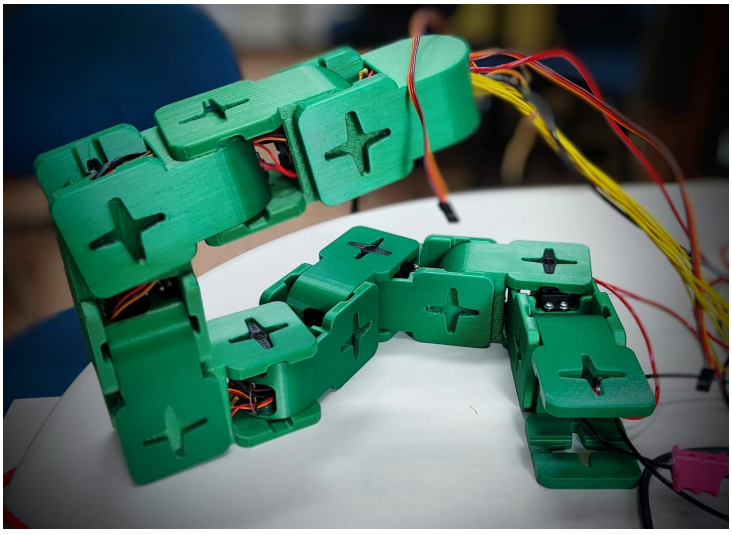

Figura 2.20 – ErekoChainBot: Robô de movimento combinado pitch-yaw.

Fonte: (Rocha, 2023)

Este robô foi projetado para enfrentar desafios de terrenos difíceis e espaços confinados, onde modelos tradicionais não se sairiam bem. A expectativa é que o ErekoChainBot, após rigorosos testes, se torne um marco na robótica modular, ampliando a aplicabilidade destas máquinas em cenários complexos e operações críticas.

No presente trabalho, o modelo ErekoChain, tem seu uso restrito a simulações focadas nos movimentos combinados pitch-yaw. O objetivo é refinar os parâmetros abc, visando alcançar uma locomoção eficiente e confiável. A aplicação deste modelo em simulações é ilustrada na Figura 2.20, evidenciando sua importância para o progresso das pesquisas desenvolvidas no Laboratório Ereko. Na Figura 2.21 é possível visualizar o ErekoChainBot em ambiente de Simulação.

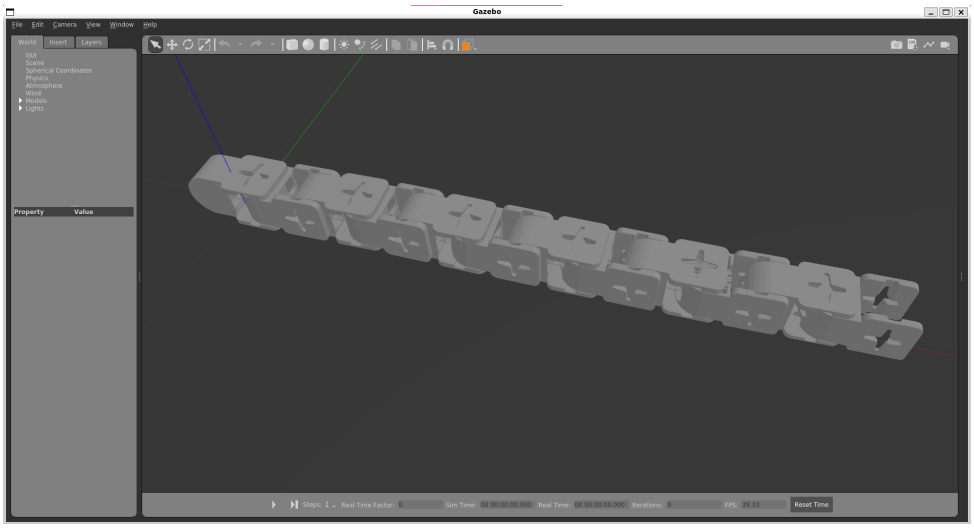

Figura 2.21 – ErekoChainBot em ambiente de Simulação Gazebo.

Fonte: A Autora.

2.5.3 Simulação de Movimentos Serpenoides

A simulação de movimentos Serpenoides em robôs é uma tarefa complexa que exige ferramentas especializadas para garantir precisão e realismo. Neste projeto, utilizou-se um conjunto integrado de tecnologias avançadas para modelar, simular e analisar o comportamento dos robôs-serpente.

O ROS (Robot Operating System) oferece uma coleção robusta de ferramentas e bibliotecas que facilitam a escrita de código para robótica. Integrado ao Gazebo, um poderoso simulador que proporciona ambientes realísticos e dinâmicos, o ROS permite a execução e teste de algoritmos complexos em um contexto virtual antes da implementação real (Sturman; Kelly; Howard, 2012).

Na estrutura de simulação implementada para o robô cobra, cada junta é controlada por um nó ROS independente, permitindo um controle paralelo dos segmentos do robô. Esta configuração facilita a locomoção adaptativa, permitindo que cada segmento responda de forma autônoma às condições dinâmicas do ambiente. A estrutura de controle implementada é inspirada na dinâmica complexa das serpentes biológicas, onde cada parte do corpo pode se mover de forma autônoma, contribuindo para a eficiência geral do movimento (Gómez, 2008).

Para a representação precisa da estrutura física dos robôs, utilizamos modelos URDF (Unified Robot Description Format) desenvolvidos no SolidWorks. Esses modelos contêm todos os detalhes mecânicos necessários, como dimensões, massas e restrições de movimento, que são essenciais para simulações precisas no Gazebo.

O controle e a automação das simulações são gerenciados por meio de scripts em Python, uma linguagem de programação versátil e amplamente adotada na robótica. Com esses scripts, automatizou-se não apenas as simulações no Gazebo, mas também a análise subsequente dos dados gerados e a produção de gráficos relevantes para a avaliação do desempenho dos movimentos simulados.

A Figura 2.22 apresenta um exemplo de uma simulação em andamento, demonstrando a aplicação prática das ferramentas mencionadas acima.

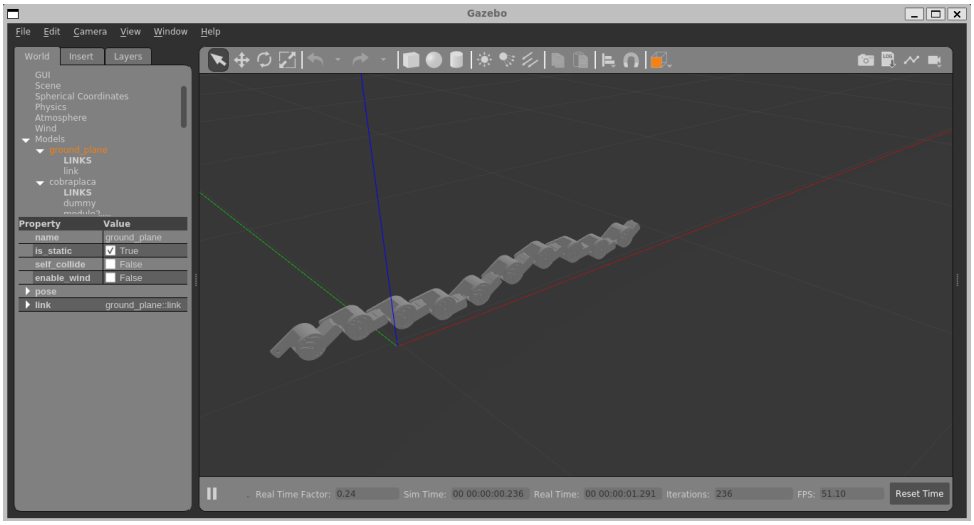

Figura 2.22 – Captura de tela de uma simulação no Gazebo de um robô-cobra pitch-pitch.

Fonte: A Autora.

2.6 Método Bayesiano de Aprendizado - Modelo Matemático e Aplicação

O Método Bayesiano, aplicado neste trabalho, é fundamental para a escolha dos parâmetros de controle do robô cobra. Esta abordagem estatística permite a atualização contínua das probabilidades de diferentes configurações com base em dados coletados durante simulações. Utilizando este modelo, é possível refinar a seleção de parâmetros que podem otimizar a eficácia da locomoção do robô, adaptando-se às características específicas do movimento serpenteante e às condições variáveis do ambiente.

2.6.1 Fundamentos Matemáticos do Método Bayesiano

O método Bayesiano baseia-se no Teorema de Bayes, que é expresso matematicamente por:

$$P(H|D) = \frac{P(D|H)P(H)}{P(D)}, \tag{2.10}$$

onde $P(H|D)$ é a probabilidade da hipótese H dada a evidência D, $P(D|H)$ é a probabilidade de observar a evidência D se a hipótese H for verdadeira, $P(H)$ é a probabilidade a priori da hipótese H, e $P(D)$ é a probabilidade de observar a evidência D (Murphy, 2012; Bishop, 2006).

Na otimização de parâmetros de um robô cobra, o Teorema de Bayes ajuda a atualizar a probabilidade de um conjunto de parâmetros ser o ideal, dadas as evidências coletadas das simulações ou testes.

2.6.2 Função `gpminimize` em Python

A função `gpminimize` do Python, parte da biblioteca Scikit-Optimize, é uma implementação do método Bayesiano de otimização (Frazier, 2018; Scikit-Optimize,). Ela usa modelos Gaussianos de Processos (Gaussian Process, GP) para aproximar a função objetivo e realizar a otimização.

Otimização refere-se ao processo de encontrar os melhores parâmetros ou configurações, visando maximizar ou minimizar uma função objetivo. A função objetivo é uma expressão matemática que define o critério de desempenho a ser otimizado. Processos Gaussianos são usados para modelar a função objetivo, proporcionando uma abordagem probabilística para estimativa e atualização sob incerteza.

Na aplicação prática, o algoritmo Bayesiano na `gpminimize` utiliza o Teorema de Bayes para ajustar iterativamente as estimativas da função objetivo. Ajustando os parâmetros baseados nos resultados das simulações, o algoritmo busca a configuração mais eficiente para o deslocamento e adaptação do movimento do robô.

O funcionamento da `gpminimize` pode ser descrito em etapas:

1. **Inicialização:** A função começa com um conjunto inicial de pontos e avalia a função objetivo nestes pontos.

2. **Atualização do Modelo:** Com base nos resultados, um modelo GP é ajustado para aproximar a função objetivo. Este modelo considera as incertezas na estimativa da função.

3. **Escolha do Próximo Ponto:** Utilizando o modelo GP, a função seleciona o próximo ponto para avaliar. Este ponto é escolhido para melhorar a otimização, explorando áreas de alta incerteza ou de potencial mínimo.

4. **Iteração:** O processo se repete, atualizando o modelo GP com cada novo ponto avaliado, até que um critério de parada seja atingido (por exemplo, um número máximo de iterações ou uma tolerância de convergência).

Na aplicação para robôs cobra, a `gpminimize` pode ser usada para encontrar os parâmetros ótimos que maximizam a eficiência da locomoção ou adaptam o movimento às características do ambiente. A função avalia como diferentes configurações de parâmetros afetam o desempenho do robô e iterativamente converge para a configuração mais eficiente.

Neste trabalho, este método foi utilizado no conjunto de parâmetros disponíveis para encontrar o maior deslocamento do robô dado o tempo de simulação, no caso dos movimentos planos esse deslocamento foi avaliado no eixo x, no caso do modelo composto, foi avaliado o deslocamento total no tempo. Importante ressaltar que um movimento instável é descartado como potencial no momento que o robô tomba (rola para os lados).

O Capítulo 2 abordou a locomoção de serpentes e a robótica bioinspirada, fundamentando a relevância do método Bayesiano de aprendizado. Foram discutidos diferentes modos de locomoção serpentina, princípios de robótica bioinspirada, e locomoção Serpenoide plana e composta. A modelagem e simulação de robôs-serpente foram exploradas, destacando a aplicação prática do método Bayesiano na otimização de parâmetros de curva de movimento. Este capítulo constitui a base teórica para o próximo, focado em Modelos Robóticos, onde serão detalhados os modelos URDF para movimentos yaw-yaw, pitch-pitch e pitch-yaw, e discutida a metodologia para simulação e controle.

3 Modelos Robóticos

Este capítulo apresenta a exposição dos Modelos Robóticos URDF (Unified Robot Description Format), focalizando nos modelos de conexão yaw-yaw, pitch-pitch e pitch-yaw. Discute-se a metodologia empregada para simulação e controle destes modelos no ambiente Gazebo, incluindo a configuração de arquivos YAML, controle dinâmico de atrito e automação das simulações. São explorados os procedimentos específicos para os modelos planos e composto, abrangendo desde a coleta de dados até a otimização Bayesiana e análise de movimento.

3.1 Descrição dos Modelos Robóticos URDF

A modelagem dos robôs cobra para os movimentos yaw-yaw, pitch-pitch e pitch-yaw é realizada utilizando uma abordagem detalhada e sistemática. Os arquivos URDF são criados a partir de modelos desenvolvidos no software de design Solidworks, utilizando a extensão "SW2URDF". Esta extensão facilita a exportação de modelos tridimensionais detalhados do Solidworks para o formato URDF, ideal para simulações e implementações em robótica.

3.1.1 Modelo Yaw-Yaw

O modelo yaw-yaw, composto por 10 juntas revolutas e 11 links. As juntas possuem limites de movimento entre -2.0 e 2.0 radianos, com esforços variando até 100 GNm (para garantir o movimento) e velocidades de até 6.28 rad/s. O modelo yaw-yaw é construído com ênfase na capacidade de realizar movimentos de guinada em sequência. Cada um dos 10 módulos do robô é conectado através de juntas de revolução, permitindo movimentos fluidos e contínuos. O modelo e suas juntas podem ser vistos na Figura 3.23

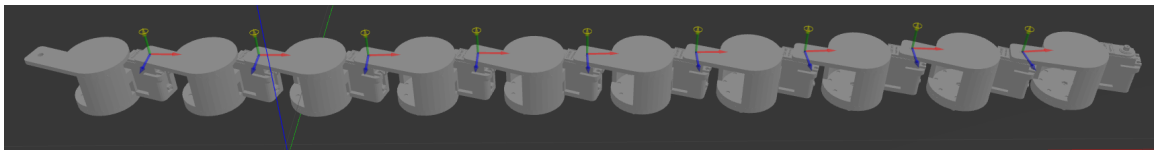

Figura 3.23 – Captura de tela do modelo yaw-yaw no Gazebo.

Fonte: A Autora.

3.1.2 Modelo Pitch-Pitch

O modelo pitch-pitch, descrito no arquivo URDF, segue uma configuração similar ao modelo yaw-yaw mas com uma rotação de 90 graus no eixo de rolamento. Ele também tem 10 juntas revolutas com limites de -2.0 a 2.0 radianos, porém esforços de até 2.0 (para

ficar mais próximo da realidade) e velocidades máximas de 5.0 rad/s. Essa configuração é otimizada para movimentos de arfagem, proporcionando uma locomoção que imita o movimento ondulatório vertical das serpentes. O modelo e suas juntas podem ser vistos na Figura 3.24

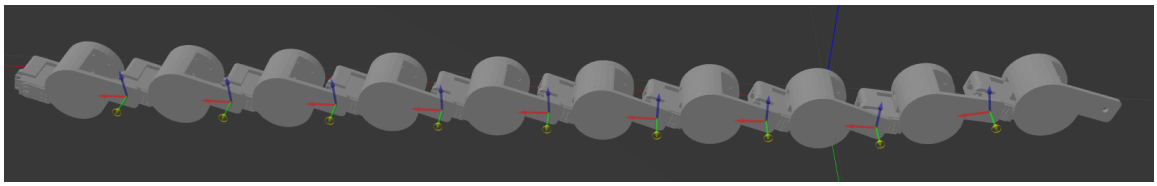

Figura 3.24 – Captura de tela do modelo pitch-pitch no Gazebo.

Fonte: A Autora.

3.1.3 Modelo Pitch-Yaw

O modelo pitch-yaw, que inclui 12 juntas revolutas com limites variando de -1.4 a 1.4 radianos (para juntas laterais) e de -0.18 a 0.18 radianos (para juntas dorsais), esforços de até 8.0 e velocidades de até 15.0 rad/s. O modelo pitch-yaw combina os movimentos de arfagem e guinada, proporcionando maior versatilidade de movimento. Com esta configuração, o robô é capaz de executar uma gama mais ampla de movimentos, tornando-o adequado para navegar em ambientes complexos com uma variedade de desafios físicos. O modelo e suas juntas podem ser vistos na Figura 3.25

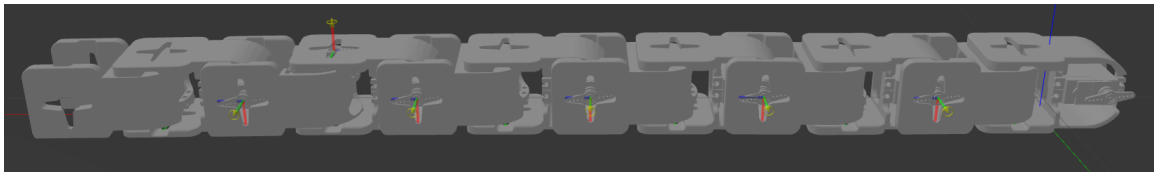

Figura 3.25 – Captura de tela do modelo pitch-yaw no Gazebo.

Fonte: A Autora.

3.2 Metodologia Empregada

A simulação dos modelos robóticos no ambiente Gazebo é realizada seguindo uma metodologia sistemática, que envolveu a integração de múltiplos códigos Python e o uso do ROS. Este processo permite a avaliação detalhada do desempenho dos robôs cobra em um ambiente virtual controlado, fornecendo insights importantes para a otimização de seus movimentos e parâmetros.

3.2.1 Simulação para Modelos Planos - Códigos e Testes

Para a simulação dos modelos planos no ambiente Gazebo, é necessário configuração detalhada, começando com a preparação do arquivo URDF. Este arquivo é configurado com

plugins de controle e odometria do Gazebo, além dos coeficientes de atrito definidos para cada modelo.

3.2.1.1 Configuração dos Arquivos YAML e Controle Dinâmico de Atrito

Os arquivos `.yaml` de cada modelo são configurados, estabelecendo os controladores de junta, controladores de estado e de posição da junta. Em seguida, implementa-se a configuração dinâmica para os coeficientes de atrito $\mu 1$ e $\mu 2$ usando scripts Python, como `AtritoNodeConfig.cfg` e `atrito_node.cfg.save`. O servidor de configuração dinâmica é inicializado com o script `dynamic_reconfigure_server.py`, permitindo a alteração dos coeficientes de atrito durante a simulação.

3.2.1.2 Lançamento da Simulação no Gazebo

O ambiente de simulação é lançado com o arquivo `gazebo_control.launch`, que carrega o mundo do Gazebo, configura a transformação estática, inicia os controladores e o modelo do robô no Gazebo, e configura o servidor de reconfiguração dinâmica. Este processo garante que o sistema de controle seja adequadamente preparado para a simulação dos movimentos planejados.

3.2.1.3 Automação das Simulações

O código Python `PYcontrol.py` **??** é utilizado para automatizar as simulações. Este script gerencia a experimentação com diferentes combinações dos parâmetros, ***'wtrad'***, ***'a'***, ***'b'***, e ***'c'***, simulando todas as combinações possíveis - em um range à passos específicos escolhido - de forma iterativa e recebendo a odometria do robô no mundo, salvando as posições do primeiro módulo. Após cada simulação, o mundo do Gazebo é resetado, e o script registra os valores dos parâmetros utilizados e os respectivos deslocamentos do robô ao longo do eixo x. O script Python `PYcontrol.py` **??** desempenha um papel fundamental na automação das simulações. Um exemplo da configuração desses parâmetros é apresentado no código a seguir:

Código 3.1 – Configuração de parâmetros, trecho do PYcontrol.py

```
1 # Exemplo de definição de parâmetros para simulação
2 wtrad = np.arange(0, 40 * np.pi/3 , 2 * np.pi / 100) # yawyaw:
     Simulação com tempo de ST: 20seg e RT: 2:15min cada
3 wtrad = np.arange(0, 20 * np.pi , 5 * np.pi / 20) # pitchpitch:
     Simulação com tempo de ST: 2.635seg e RT: 14.880seg
4
5 # YAWYAW Defina os intervalos para os parâmetros a, b e c
6 a_values = np.linspace(0.7, 3 * np.pi/5, num=7)
7 b_values = np.linspace(2, 2 * np.pi, num=8)
8 c_values = np.linspace(0, np.pi/3, num=4)
9
```

```
# PITCHPITCH Defina os intervalos para os parâmetros a, b e c
a_values = np.linspace(0.5, 2*np.pi, num=3)
b_values = np.linspace(0.5, 2*np.pi, num=3)
c_values = np.linspace(0, np.pi/3, num=3)
```

1. Criação de arrays `wtrad` para armazenar valores de frequência angular para simulações diferentes. O primeiro para o modelo yawyaw e o segundo para o pitchpitch.

2. Definição de intervalos para os parâmetros a, b, e c para o modelo yawyaw. São usados para gerar diferentes configurações de movimento do robô.

3. Definição de intervalos semelhantes para os parâmetros a, b, e c, mas desta vez para o modelo pitchpitch.

3.2.1.4 Processamento e Análise dos Dados

O script `analise_sim_results.py` **??**, processa os dados gerados pelo `PYcontrol.py` **??**, comparando os valores iniciais e finais de x para calcular o deslocamento total do robô para cada conjunto de parâmetros. Este script gera um ranking dos parâmetros com base no deslocamento observado (por meio da velocidade média calculada), ajudando a classificar as configurações de movimento.

3.2.1.5 Visualização dos Resultados

Posteriormente, o `x_displacement_animation.py` **??** é usado para visualizar os resultados. Este script cria gráficos animados mostrando o deslocamento em x ao longo do tempo para diferentes conjuntos de parâmetros. Além disso, plota gráficos de velocidade. Para todos esses pontos que se configuram como bloco de parâmetros, os gráficos gerados oferecem uma apresentação visual detalhada do desempenho do robô.

3.2.1.6 Otimização e Análise de Atrito

O próximo código, `bayesiana_optimization.py` **??**, aplica a otimização Bayesiana para testar os parâmetros ‘abc‘ nas simulações, procurando os melhores casos de deslocamento do robô. Este script também verifica a ocorrência de tombamento do robô nas simulações do modo pitch-pitch, algo que necessita de verificação visual no caso sem essa otimização.

Código 3.2 – Trecho de bayesiana_optimization.py

```
# Exemplo de definição de parâmetros para simulação
# YAWYAW
# Defina o espaço bayesiano de parâmetros para a, b e c
space  = [Real(0, 3*np.pi/2, name='a'),
          Real(0.5, np.pi, name='b'),
```

```
6             Real(0, 1, name='c')]
7
8 # PITCHPITCH
9 # Defina o espaço bayesiano de parâmetros para a, b e c
10 space  = [Real(0.5, 2*np.pi, name='a'),
11            Real(np.pi/2, 6*np.pi, name='b'),
12            Real(0, np.pi/3, name='c')]
```

1. Definição do espaço de parâmetros para otimização Bayesiana para o modelo yawyaw. Os parâmetros a, b, e c são definidos com intervalos específicos.

2. Definição similar do espaço de parâmetros para otimização Bayesiana, mas para o modelo pitchpitch, com intervalos diferentes para os parâmetros a, b, e c.

Após diversos testes com valor de atrito fixo e a comparação entre os métodos de busca exaustiva e otimização bayesiana, foi empregado o script `bayesiana_optimization_5_friction.py` **??** com blocos de parâmetros fixos testados anteriormente que obtiveram bom resultado no deslocamento e estabilidade. Dessa vez focamos na avaliação dos parâmetros de atrito $\mu 1$ e $\mu 2$, buscando o desempenho do robô (maior deslocamento possível com estabilidade) pensando que o atrito anisotrópico é um fator de extrema relevância na movimentação das cobras, que possuem escamas que aplicam tal propriedade.

Finalmente, o `x_displacement_animation_friction.py` **??** permite a visualização do impacto do atrito no deslocamento do robô para blocos de parâmetros já testados anteriormente, gerando gráficos para cada conjunto de parâmetros 'abc', com cada ponto representando diferentes valores de atrito para um mesmo bloco.

3.2.2 Simulação para Modelo Composto - Códigos e Testes

Para a simulação do modelo composto, o processo inicial segue etapas semelhantes às dos modelos planos, com a configuração do arquivo URDF com os plugins necessários de controle e odometria do Gazebo.

3.2.2.1 Configuração Específica do Modelo Composto

Diferentemente dos modelos planos, a modificação dos coeficientes de atrito não é aplicada para o modelo composto, visto que estamos preocupados com os padrões de locomoção do robô. O arquivo launch utilizado é o `gazebo_control_thi.launch`, que realiza a mesma série de tarefas de configuração do mundo do Gazebo, transformação estática, carregamento dos controladores e inicialização do robô, porém com parâmetros e configurações específicos para o modelo composto.

3.2.2.2 Automação e Coleta de Dados

Inicialmente, o script `PYcontrol_pitchyaw.py` **??** é empregado para testar os movimentos do robô. Este script foca na simulação com um único ponto de odometria, explorando diferentes posições ao longo do corpo do robô (cabeça, centro e cauda), ou seja, é necessário rodar 3 simulaçoes completas e diferentes para obter as posições dos diferentes pontos de odometria. Um trecho do código que ilustra esta etapa é apresentado a seguir:

Código 3.3 – Configuração inicial de odometria no PYcontrol_pitchyaw.py

```
# Definindo ponto de odometria no URDF
odom_type = 'odom' # head
odom_type = 'odom_3' # body
odom_type = 'odom_3' # tail
rospy.Subscriber(f'/robocobra_pkg/{odom_type}', Odometry,
    SnakeRobotController.odom_callback)
```

1. Define o tipo de odometria para o robô. A odometria pode ser configurada para diferentes partes do robô (cabeça, corpo, cauda).

2. Inicializa um assinante ROS para receber dados de odometria do tipo especificado.

Este método inicial permite a análise de movimentos individuais em diferentes partes do robô, mas não em uma mesma simulação, portanto, precisou ser alterado para conseguirmos as posições dos três pontos numa mesma simulação.

3.2.2.3 Automação e Coleta de Dados - Três Pontos de Odometria

Posteriormente, o script é atualizado para `PYcontrol_pitchyaw_odom.py` **??**, que realiza simulações usando três pontos de odometria simultaneamente. Este avanço possibilita a visualização mais intuitiva dos movimentos do robô no plano xy quando plotados no gráfico xy no tempo.

Código 3.4 – Atualização para três pontos de odometria.

```
# Configuração de múltiplos pontos de odometria
rospy.Subscriber('/robocobra_pkg/odom_head', Odometry,
    SnakeRobotController.odom_callback_head)
rospy.Subscriber('/robocobra_pkg/odom_body', Odometry,
    SnakeRobotController.odom_callback_body)
rospy.Subscriber('/robocobra_pkg/odom_tail', Odometry,
    SnakeRobotController.odom_callback_tail)
```

1. Inicialização da configuração de múltiplos pontos de odometria.

2. Assinatura para o tópico de odometria da cabeça do robô. Esta linha estabelece um canal de comunicação (*subscriber*) no ROS para receber dados de odometria da cabeça do robô, utilizando o callback `odom_callback_head` para processar esses dados.

3. Assinatura para o tópico de odometria do corpo do robô. Similarmente, esta linha cria um *subscriber* para receber e processar dados de odometria referentes ao corpo do robô, usando o callback `odom_callback_body`.

4. Assinatura para o tópico de odometria da cauda do robô. Esta linha configura um *subscriber* no ROS para obter dados de odometria da cauda do robô, processando-os com o callback `odom_callback_tail`.

3.2.2.4 Visualização e Análise dos Movimentos

O script `xy_displacement_animation.py` **??** é utilizado para gerar gráficos de posição no plano xy, permitindo a análise dos movimentos serpenteantes do robô. A análise foca em identificar os tipos de movimento (retilíneo, lateral ou curvo) baseada nos parâmetros testados por meio dos gráficos de forma mais intuitiva.

3.2.2.5 Otimização Bayesiana e Análise de Movimento

Em seguida, a otimização Bayesiana já é utilizada desde o primeiro instante de simulações já que nos casos de movimento no plano ela se mostra extremamente mais eficaz que os testes de forma bruta. Aqui, ela é utilizada para maximizar a velocidade de deslocamento total.

Para determinar se a otimização Bayesiana não caiu em um máximo local ao maximizar a velocidade de deslocamento total, é possível verificar a consistência e a generalidade dos resultados obtidos. Analisando os gráficos de deslocamento, observa-se se existe progressão lógica e gradual no aumento da velocidade à medida que os parâmetros são ajustados, em vez de saltos abruptos ou inconsistências.

Além disso, pode-se realizar testes adicionais em um conjunto diversificado de condições e verificar se os parâmetros otimizados mantêm seu desempenho superior, indicando que o máximo alcançado é global e não apenas local. Os scripts de controle geraram um ranking dos parâmetros baseado na velocidade máxima alcançada pela cabeça do robô e então os gráficos de deslocamento são analisados visualmente.

Código 3.5 – Escolha de espaço bayesiano para os parâmetros de movimento composto.

```
space = [
        Real(0.1, 2.0, name='delta_rate'),
        Real(5, 30, name='omega_rate'),
        Real(0.6, 0.95, name='omega_rate_dor'),
        Real(2, 10, name='nu_rate'),
        Real(2, 10, name='nu_rate_dor'),
```

```
7           Real(5, 7, name='pub_rate')
8           ]
```

1. Define o espaço de parâmetros para otimização Bayesiana em um modelo de movimento composto. Os parâmetros incluem taxas de variação para delta, omega, e outros, que influenciam o movimento do robô.

3.2.2.6 Estudos Futuros e Aplicações

Estas simulações abrem caminho para estudos futuros, onde seria interessante a criação de um diagrama de "Áreas de Movimentação"para diferentes parâmetros, ajudando a determinar configurações específicas para movimentos desejados, como retilíneo, lateral ou curvo. Isso seria interessante já que cada robô possui seu próprio padrão de movimento por conta de suas peculiaridades de formato. Assim teríamos um resultado mais completo de quais parâmetros usar para atingir um movimento específico e ainda ter uma margem de erro.

Este capítulo forneceu uma análise dos Modelos Robóticos URDF utilizados, com foco nas configurações yaw-yaw, pitch-pitch e pitch-yaw. Exploramos a metodologia empregada na simulação e controle desses modelos no Gazebo, abrangendo a configuração dos arquivos YAML, controle dinâmico de atrito e automação das simulações. Abordamos os procedimentos específicos para os modelos planos e composto, desde a coleta de dados até a otimização Bayesiana e a análise do movimento. Esse fundamento prepara o terreno para o próximo capítulo, "Análises e Resultados", onde serão discutidos os insights e interpretações derivados das simulações e testes aqui apresentados.

4 Análises e Resultados

O Capítulo 4, intitulado "Análises e Resultados", desdobra-se na apresentação dos resultados obtidos através das simulações das conexões robóticas yaw-yaw, pitch-pitch no modelo VDM e pitch-yaw no modelo ErekoChain. Inicialmente, o capítulo se concentra na exploração dos resultados da conexão yaw-yaw, enfatizando o impacto dos parâmetros de curva e do atrito anisotrópico no deslocamento do robô. A seção seguinte foca na conexão pitch-pitch, onde se analisa igualmente a influência desses parâmetros, com ênfase adicional na estabilidade do robô (questão de possível tombamento do robô). Por fim, o modelo de movimento ondulatório composto (com conexão pitch-yaw) é detalhado, com análises que categorizam os movimentos retilíneo, *sidewinding* e curvo. Em cada conexão é empregado métodos de otimização Bayesiana para deslocamento/velocidade e atrito para análise de movimento dos robôs.

4.1 Conexão Yaw-Yaw

A simulação e análise do modelo VDM com conexão Yaw-Yaw foi conduzida com uma metodologia sistemática, conforme descrito em 3.2, e revelaram dados sobre o desempenho do robô para o deslocamento em x com a influência dos parâmetros de Curva e o Atrito. Esta seção apresenta os resultados obtidos juntamente com suas análises.

4.1.1 Parâmetros de Curva

A aplicação da otimização Bayesiana é um passo importante para refinar os testes e obter resultados mais alinhados com o objetivo de forma mais rápida. Os dados coletados para 100 simulações (de faixa de valores observado no trecho de código 3.2) foram processados pelo script `analise_sim_results.py` **??**, que isolou os dez melhores conjuntos de parâmetros baseados no maior deslocamento da cabeça do robô em X, sendo o eixo X a direção em que o corpo do robô cobra começa apontado.

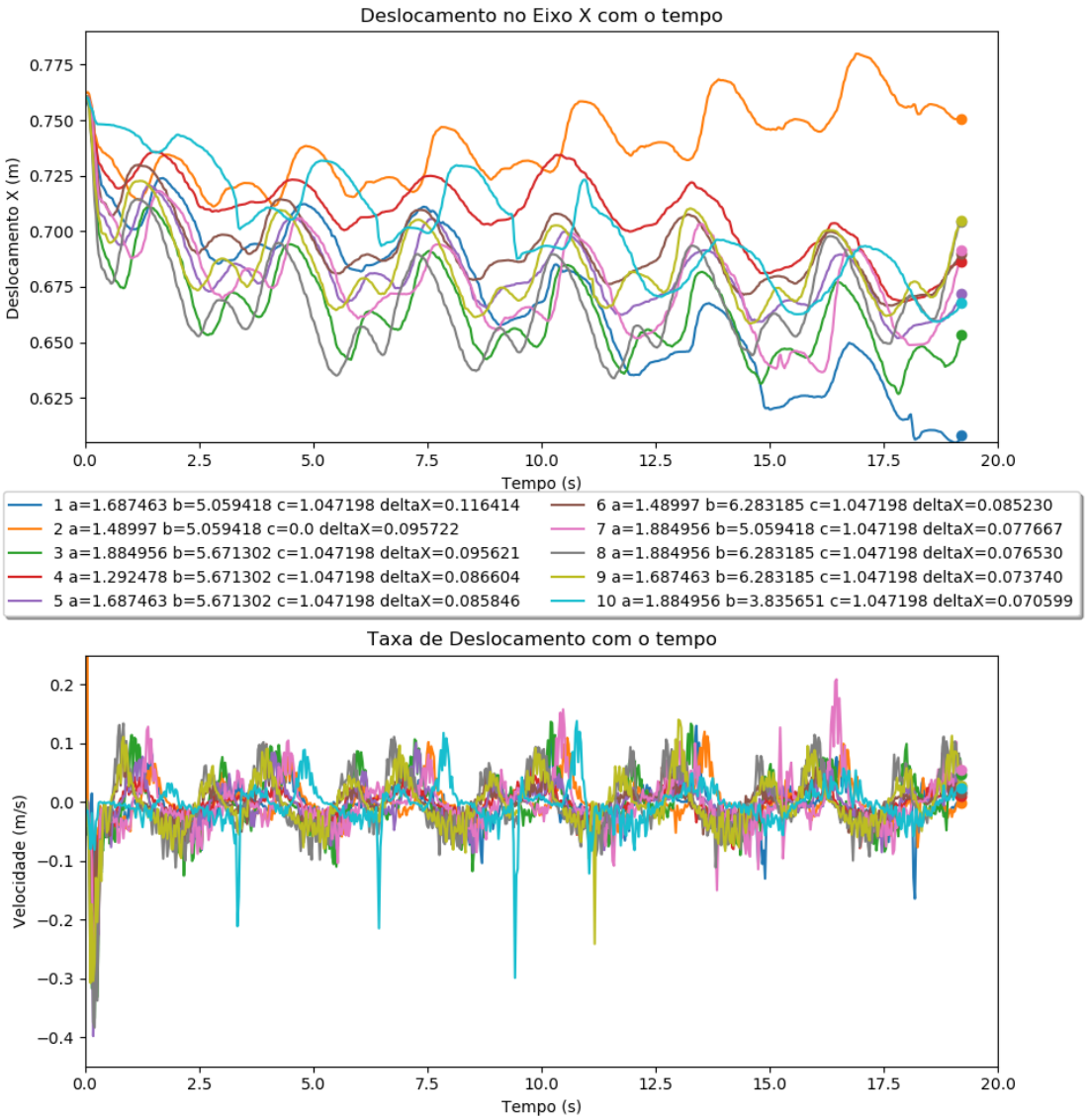

Figura 4.26 – Deslocamento em x pelo tempo para diferentes parâmetros. Velocidade de deslocamento em x no tempo para os mesmos parâmetros.

O script `x_displacement_animation.py` **??** gera um gráfico de deslocamento que ilustra a trajetória dos dez melhores conjuntos de parâmetros ao longo do tempo, ou seja, cada ponto representa um conjunto de parâmetros de curva (1 simulação) descritos na legenda, além disso, o segundo gráfico da Figura 4.26 mostra a velocidade de deslocamento de cada ponto respectivo no tempo, proporcionando uma visão mais clara do desempenho dinâmico do robô. Na tabela 4.2 é possível visualizar os resultados encontrados para cada conjunto de parâmetros.

Tabela 4.2 – Resultados das simulações para diferentes conjuntos de parâmetros abc (Yaw-Yaw).

Simulação	a	b	c	**Deslocamento Δx (mm)**
1	1.687463	5.059418	1.047198	116.414
2	1.48997	5.059418	0.0	95.722
3	1.884956	5.671302	1.047198	95.621
4	1.292478	5.671302	1.047198	86.604
5	1.687463	5.671302	1.047198	85.846
6	1.48997	6.283185	1.047198	85.230
7	1.884956	5.059418	1.047198	77.667
8	1.884956	6.283185	1.047198	76.530
9	1.687463	6.283185	1.047198	73.740
10	1.884956	3.835651	1.047198	70.599

Na tabela 4.2, os valores de **c** não parecem mudar, mas na realidade eles mudam à partir da oitava casa decimal, portanto não foi possível visualizar essa diferença, isso nos diz que dos 100 casos de simulação utilizando a otimização bayesiana os valores de **c** para os primeiros 10 melhores conjuntos de parâmetros de curva foram muito próximos.

4.1.2 Atrito Anisotrópico

Para entender melhor a influência do atrito anisotrópico, foram selecionados os quatro melhores resultados (com maior deslocamento) do gráfico de deslocamento que podem ser vistos na Tabela 4.3. A escolha de 4 conjuntos de valores foi arbitrária, apenas para afunilar o escopo de testes.

Tabela 4.3 – Coeficientes abc utilizados nas simulações de Atrito (Yaw-Yaw).

Simulação	a	b	c
1	1.687463	5.059418	1.047198
2	1.48997	5.059418	0.0
3	1.884956	5.671302	1.047198
4	1.292478	5.671302	1.047198

Então, realizou-se 20 novas simulações utilizando o método bayesiano para diferentes valores dos coeficientes de atrito na faixa de 0.1 a 1. O script `x_displacement_animation_friction.py` **??** foi usado para gerar gráficos de deslocamento selecionando as cinco melhores configurações de atrito para cada um dos quatro melhores conjuntos de parâmetros abc (como visualizado em 4.27).

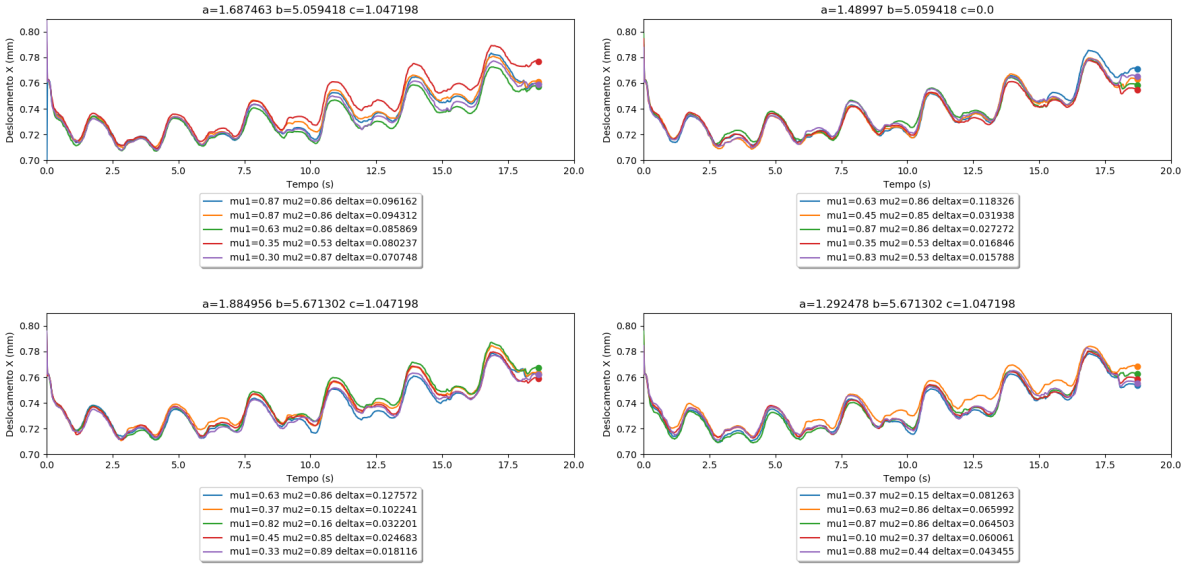

Figura 4.27 – Gráficos de deslocamento no Eixo X para quatro conjuntos de parâmetros abc sob cinco diferentes configurações de atrito. Correção: O deslocamento está em metros.

A Figura 4.27 nos mostra que a mudança dos parâmetros de atrito para os melhores casos (maior deslocamento) não é muita, mas ela existe e aumenta com o tempo, o que poderia nos mostrar que essa diferença pode aumentar conforme observamos por mais tempo.

Tabela 4.4 – Resultados das simulações para diferentes conjuntos de parâmetros abc com variações de $\mu 1$ e $\mu 2$ (Yaw-Yaw).

Conjunto	a	b	c	$\mu 1$	$\mu 2$	Δx **(mm)**
1	1.687463	5.059418	1.047198	0.87	0.86	96.162
				0.63	0.86	85.869
				0.35	0.53	80.237
				0.30	0.87	70.748
2	1.884956	5.671302	1.047198	0.63	0.86	127.572
				0.37	0.15	102.241
				0.82	0.16	32.201
				0.45	0.85	24.683
3	1.48997	5.059418	0.0	0.63	0.86	118.326
				0.45	0.85	31.938
				0.87	0.86	27.272
				0.35	0.53	16.646
4	1.292478	5.671302	1.047198	0.37	0.15	81.263
				0.63	0.86	65.992
				0.87	0.86	64.503
				0.10	0.37	60.601
				0.88	0.44	43.455

É possível perceber que o primeiro conjunto de parâmetros mostrado anteriormente

em 4.4, para uma iteração de 20 conjuntos de coeficientes de atrito (Figura 4.28) observa-se que a influência no deslocamento ocorre de maneira mais acentuada à medida que o tempo de simulação avança, portanto, com maiores tempos de simulação, veríamos uma diferença significativa.

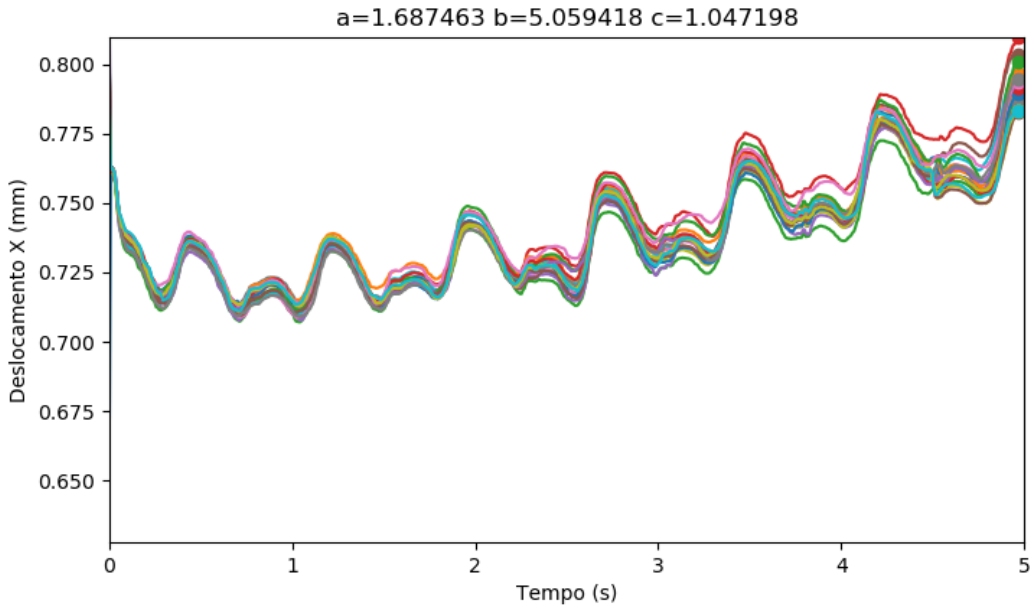

Figura 4.28 – Gráfico de deslocamento no Eixo X para conjunto de parâmetros abc fixo sob vinte diferentes configurações de atrito. Correção: O deslocamento está em metros.

4.2 Conexão Pitch-Pitch

4.2.1 Parâmetros de Curva

A análise do modelo VDM com conexão Pitch-Pitch segue uma abordagem semelhante ao Yaw-Yaw, focando na identificação dos parâmetros que maximizam o deslocamento no eixo X. Utilizando o método de otimização Bayesiana, é possível afinar a seleção de parâmetros para atingir deslocamentos maiores.

A diferença entre os modelos de movimento reside no fato de que, neste caso, não se pode escolher parâmetros que resultem no tombamento do robô durante seu deslocamento; ele necessita permanecer estável no eixo Z. Essa verificação é realizada por meio de vídeos das simulações (como mostrado na Figura 4.29), nos quais as iterações de teste são numeradas e os resultados nos quais o robô tomba são descartados como inviáveis após a criação de um ranking de conjunto de parâmetros com maior velocidade média.

Figura 4.29 – Captura de tela do vídeo de aferimento de tombamento do robô.

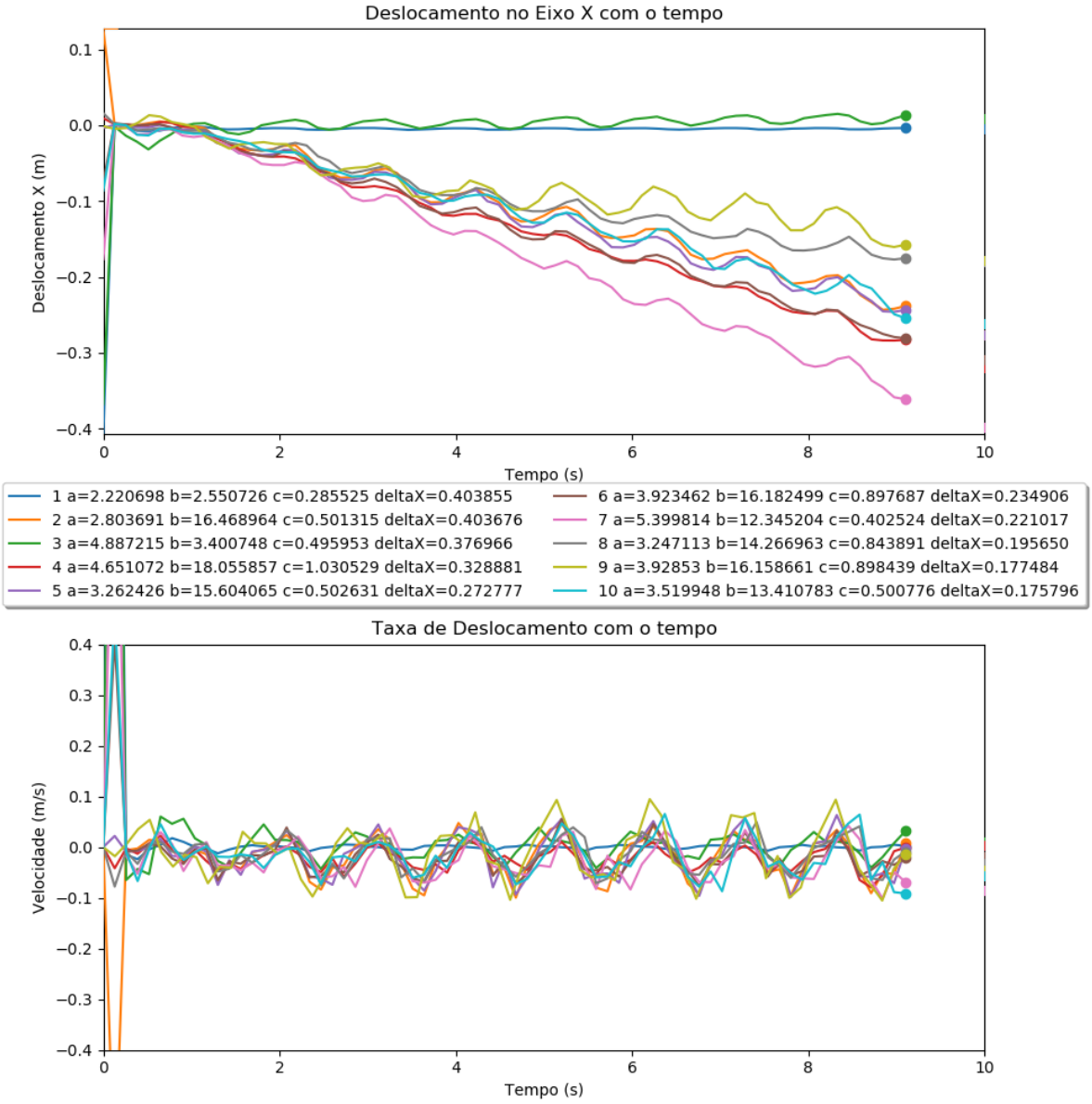

Figura 4.30 – Deslocamento no eixo X para o modelo Pitch-Pitch com diferentes conjuntos de parâmetros. Velocidade de deslocamento em x no tempo para os mesmos parâmetros.

A figura superior 4.30 mostra o deslocamento da cabeça do robô no eixo X ao longo do tempo para dez conjuntos de parâmetros e suas respectivas velocidades no tempo. Os melhores 10 resultados (com maior velocidade média) são mostrados na Tabela 4.5.

Tabela 4.5 – Resultados das simulações para diferentes conjuntos de parâmetros (Pitch-Pitch).

Rank	**Simulação**	a	b	c	Δx **(mm)**	**Descrição**
1º	12	2.513185	16.50402	-0.710921	156.63	Instável
2º	7	5.399814	12.345204	-0.402524	221.017	Instável
3º	14	6.283185	18.849556	-1.047198	98.013	Tombou
4º	4	4.651072	18.055857	-1.030529	328.881	Estável
5º	6	3.923462	16.182499	-0.897687	234.906	Estável
6º	18	2.771547	16.0172	-0.35332	180.63	Estável
7º	16	1.311675	16.604825	-0.495961	45.062	Estável
8º	20	3.10375	15.564619	-0.641756	0.415	Estável
9º	19	0.5	18.479746	0.0	-10.783	Instável
10º	2	2.803691	16.468964	-0.501315	403.676	Estável

Considera-se "Instável"aquela simulação em que o robô não tomba em tempo de simulação, mas poderia tombar facilmente com mais tempo ou com algum pequeno desnível no trajeto, pois o robô chacoalha muito, esse critério foi visual, feito por vídeo, como dito em 4.2.1. O gráfico de deslocamento (Figura 4.31) mostra mais 10 outros pontos que não estão presentes na Figura 4.30 mas que fazem parte da Tabela 4.5 por conta dos diferentes critérios de desempenho escolhidos nos dois, maior deslocamento e velocidade.

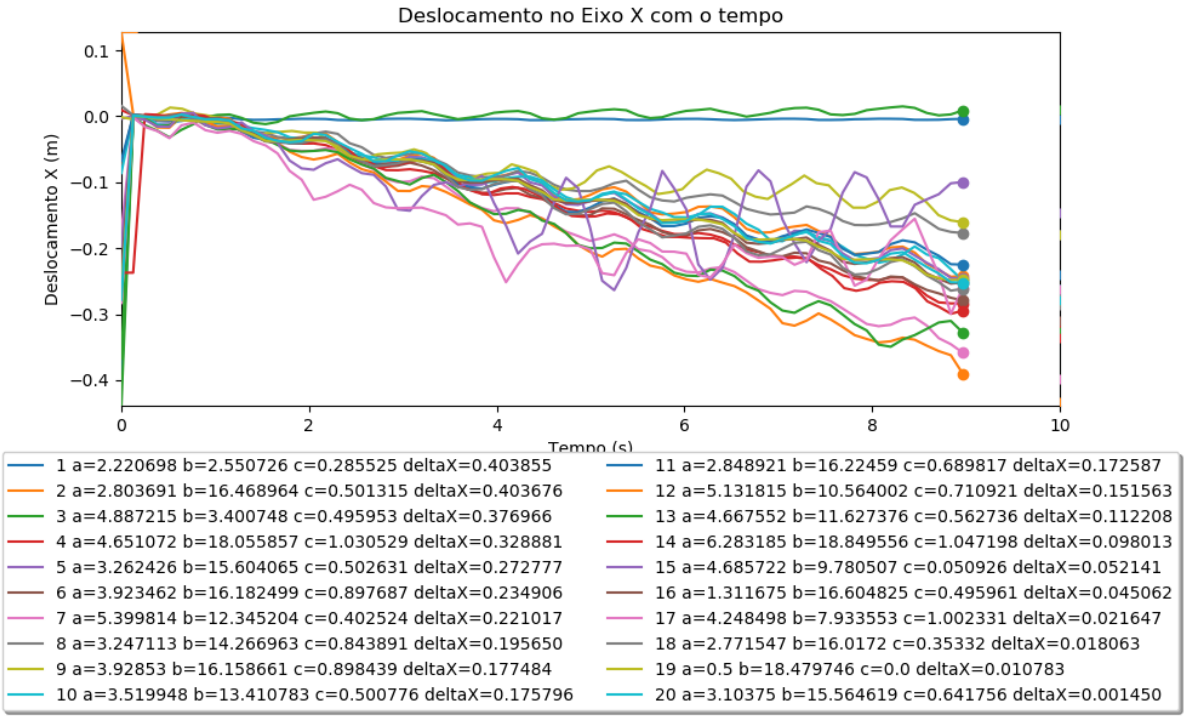

Figura 4.31 – Deslocamento no eixo X para a conexão Pitch-Pitch com diferentes conjuntos de parâmetros.

A estabilidade do movimento é uma condição importante para garantir que o robô possa operar eficientemente e sem falhas durante a locomoção. A Simulação 4, ao ser estável visualmente e ter um deslocamento significativamente alto, oferece um equilíbrio entre o alcance efetivo e a segurança operacional, tornando-a a escolha mais adequada para implementação em um cenário real.

4.2.2 Atrito Anisotrópico

A influência do atrito anisotrópico foi examinada de maneira análoga a conexão Yaw-Yaw. A partir dos melhores resultados de deslocamento, conduziu-se simulações adicionais variando os coeficientes de atrito $\mu 1$ e $\mu 2$, permitindo a observação direta de como o atrito afeta o desempenho do robô ao longo do tempo.

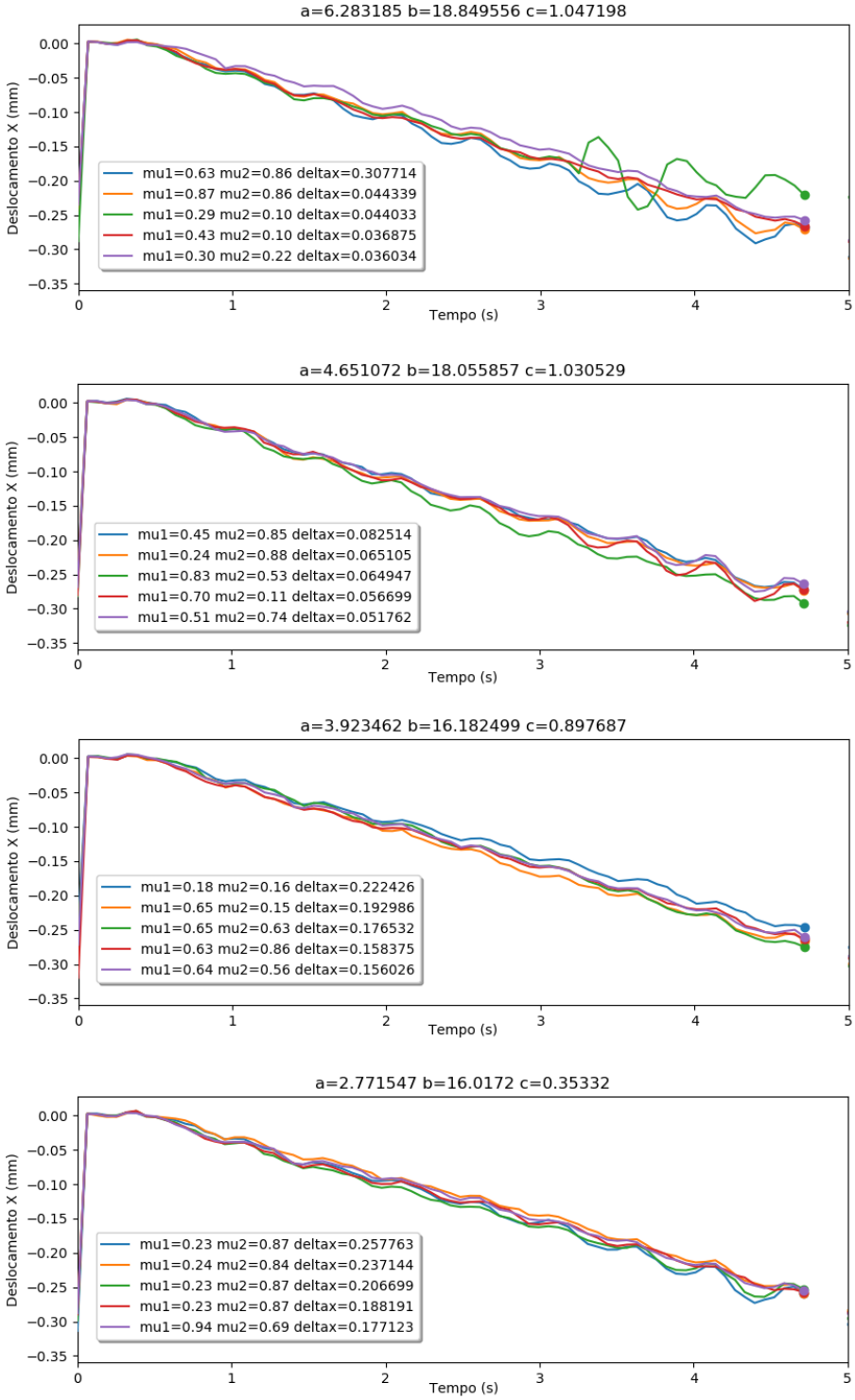

Figura 4.32 – Influência dos coeficientes de atrito no deslocamento no eixo X para a conexão Pitch-Pitch. Correção: O deslocamento está em metros.

A figura 4.32 ilustra o impacto dos diferentes coeficientes de atrito sobre o deslocamento no eixo X. Observa-se que alterações nos coeficientes de atrito resultam em variações significativas no deslocamento, o que destaca a importância do atrito anisotrópico na locomoção efetiva do robô.

Conjunto	a	b	c	$\mu 1, \mu 2$	$\Delta x(mm)$
1	6.283185	18.84956	1.047198	0.63, 0.86	307.714
				0.87, 0.86	44.339
				0.29, 2.10	44.303
				0.43, 0.10	36.875
				0.30, 0.22	36.034
2	4.651072	18.055857	1.030529	0.45, 0.85	82.514
				0.24, 0.88	65.105
				0.83, 0.53	64.947
				0.70, 0.11	56.699
				0.51, 0.74	51.762
3	3.932462	16.182499	0.897687	0.18, 0.16	222.426
				0.65, 0.15	199.286
				0.65, 0.63	176.532
				0.63, 0.86	158.375
				0.64, 0.56	156.026
4	2.771547	16.0172	0.35332	0.23, 0.87	257.763
				0.24, 0.84	237.144
				0.23, 0.87	206.699
				0.23, 0.87	188.191
				0.94, 0.69	177.123

Tabela 4.6 – Resultados das simulações com diferentes conjuntos de parâmetros abc (Pitch-Pitch). Correção: O deslocamento está em metros.

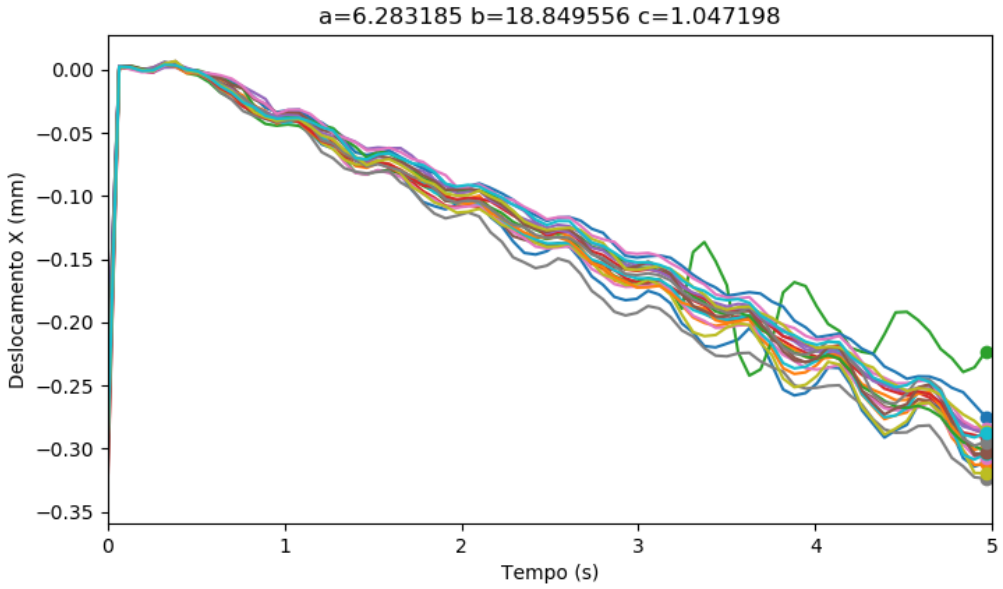

Figura 4.33 – Representação do deslocamento no eixo X para diferentes coeficientes de atrito na conexão Pitch-Pitch.

Observando o primeiro conjunto de parâmetros mostrado anteriormente (Figura 4.32), para uma iteração de 20 conjuntos de coeficientes de atrito (Figura 4.33) tem-se que a influência no deslocamento ocorre de maneira mais acentuada à medida que o tempo de

simulação avança, da mesma forma que acontecia para a conexão Yaw-yaw, e a mudança de atrito, não teve influência visível na estabilidade do robô (observada pelos vídeos).

4.3 Conexão Pitch-Yaw

Esta seção explora os movimentos ondulatórios compostos, especificamente a conexão pitch-yaw, através de um processo de simulação e análise. Os movimentos são categorizados com base no comportamento do robô, como movimento retilíneo, *sidewinding* e curva conforme vimos na Figura 4.34. Onde sidewiding é considerado um movimento lateral, qualquer movimento que faça curva evidente entra em movimento curvo (sendo ele no sentido horário ou anti-horário) e movimento retilíneo é aquele em que a cabeça do robô aponta para a dereção do movimento.

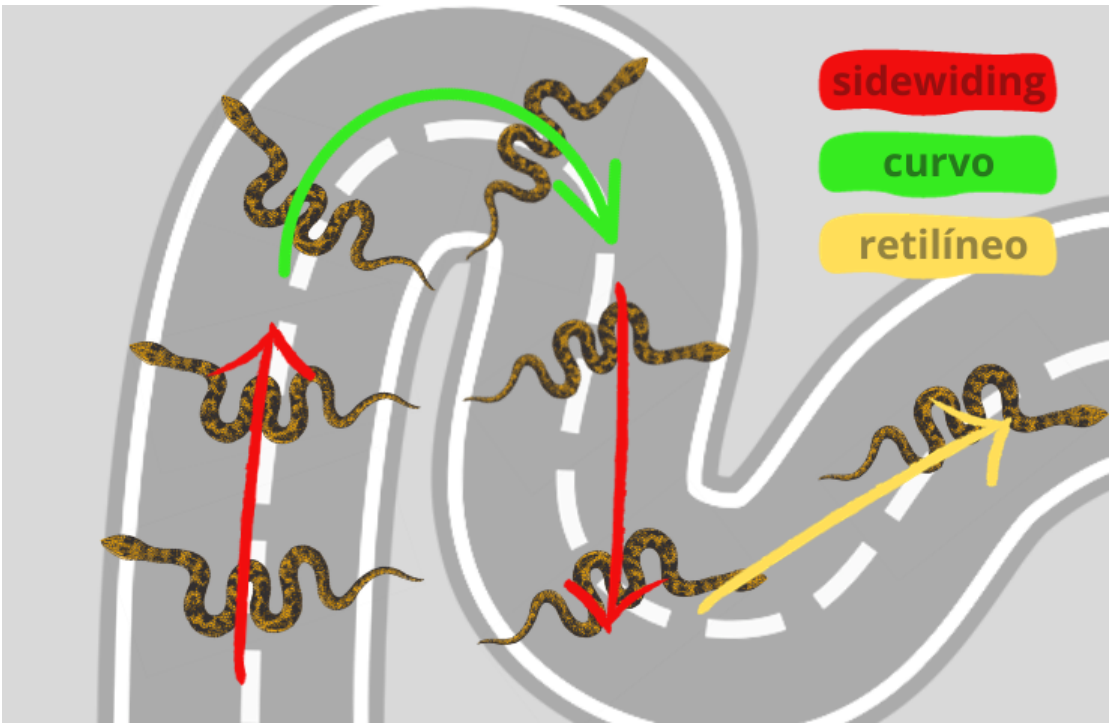

Figura 4.34 – Representação do deslocamento no eixo X para diferentes coeficientes de atrito na conexão Pitch-Pitch.

Fonte: A Autora.

4.3.1 Análise de Movimentos por Parâmetros

Inicialmente, uma simulação abrangente foi executada utilizando otimização Bayesiana com 100 execuções, mapeando o movimento da cabeça, centro do corpo e cauda do robô no plano xy ao longo do tempo. A trajetória foi traçada para cabeça e cauda, com o caminho da cauda denotado por linhas pontilhadas e as posições iniciais marcadas com um 'x' como pode ser visualizado na Figura 4.35. As simulações foram ordenadas com base na

velocidade, representada pelo módulo do vetor final menos o inicial da cabeça dentro do tempo de simulação.

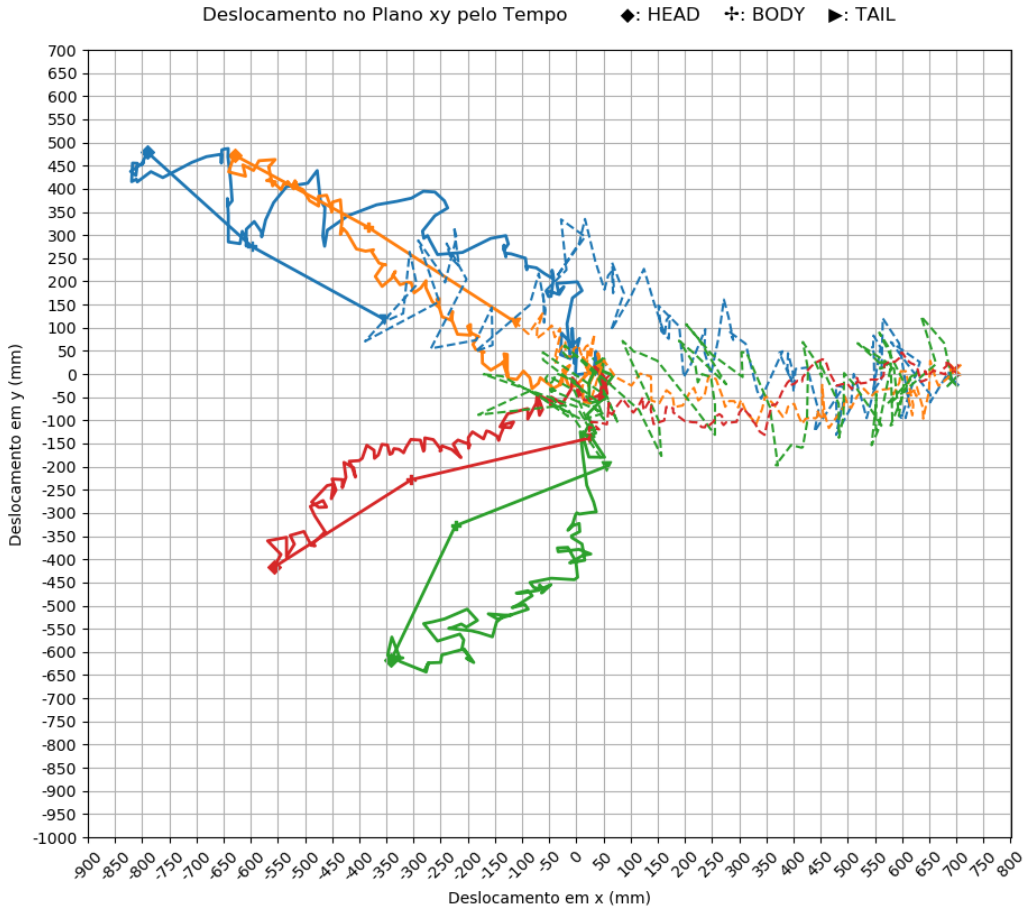

Figura 4.35 – Trajetórias exemplo para quatro casos selecionados.

Dentre os 100 casos, um subconjunto de 30 movimentos foi selecionado, incluindo padrões de movimento retilíneo (para frente), curvo e *sidewinding*. Simulações subsequentes dos parâmetros escolhidos foram inspecionadas visualmente em grupos de cinco para determinar os movimentos de interesse. Eventualmente, seis movimentos distintos foram identificados — dois curvos (um no sentido horário e outro no sentido anti-horário), dois retos e dois diagonais (um para cima e outro para baixo).

Relembrando a curva serpenoide composta:

$$\theta(n,t) = \begin{cases} \beta_h + A_h(n)\sin(\omega_h t + \nu_h n) & \text{, e} \\ \beta_v + eA_v(n)\sin(\omega_v t + \nu_v n + \delta) & \text{,} \end{cases} \tag{4.1}$$

Tabela 4.7 – Movimentos Pitch-Yaw selecionados e seus parâmetros correspondentes

Movimento	**Vel. [mm/s]**	**Δpos. [mm]**	Δt **[s]**	δ	**Pub Rate**	ω_h	ω_v	ν
Retilíneo 1	37.36	759.7	20.335	4.14689	5.3	10.98	203.512	6.46820
Retilíneo 2	34.83	682.0	19.581	2.18270	5.5	11.0159	217.760	8.57804
Sidewiding 1 (↖)	52.90	974.9	18.430	3.31263	5.9	11.4067	152.795	10.0000
Sidewiding 2 (↙)	44.16	722.3	16.355	4.50415	6.6	11.0153	178.682	5.64849
Curva Horária	19.32	387.8	20.071	2.60763	5.4	9.88330	165.750	6.11024
Curva Anti-Horária	16.34	291.7	17.847	1.53570	6.1	12.1150	207.548	9.87488

Os movimentos escolhidos representam manobras básicas que podem ser combinadas para navegação multidirecional, com os movimentos diagonais sendo selecionados por sua maior velocidade. As figuras a seguir ilustram as trajetórias para cada um dos movimentos selecionados, fornecendo uma representação visual do deslocamento do robô.

4.3.1.1 Movimento Retilíneo

Os movimentos retilíneos representam a forma mais simples de navegação, onde o robô se move diretamente em direção ao seu alvo. Podemos observar na Figura 4.36 um vetor em vermelho que representa o deslocamento resultante da cabeça do robô.

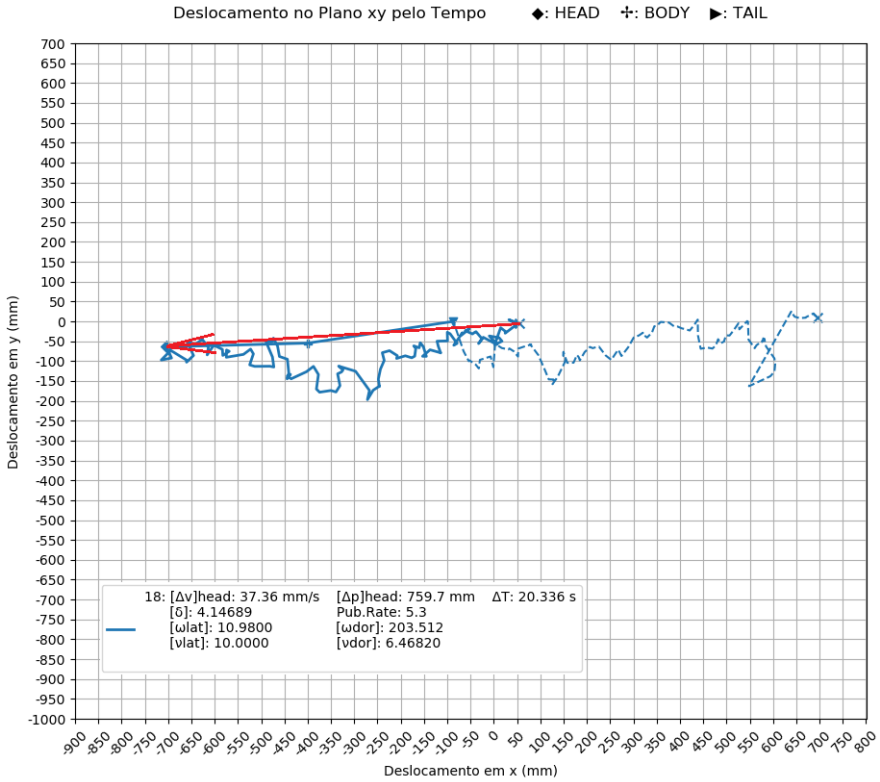

Figura 4.36 – Trajetória retilínea (1).

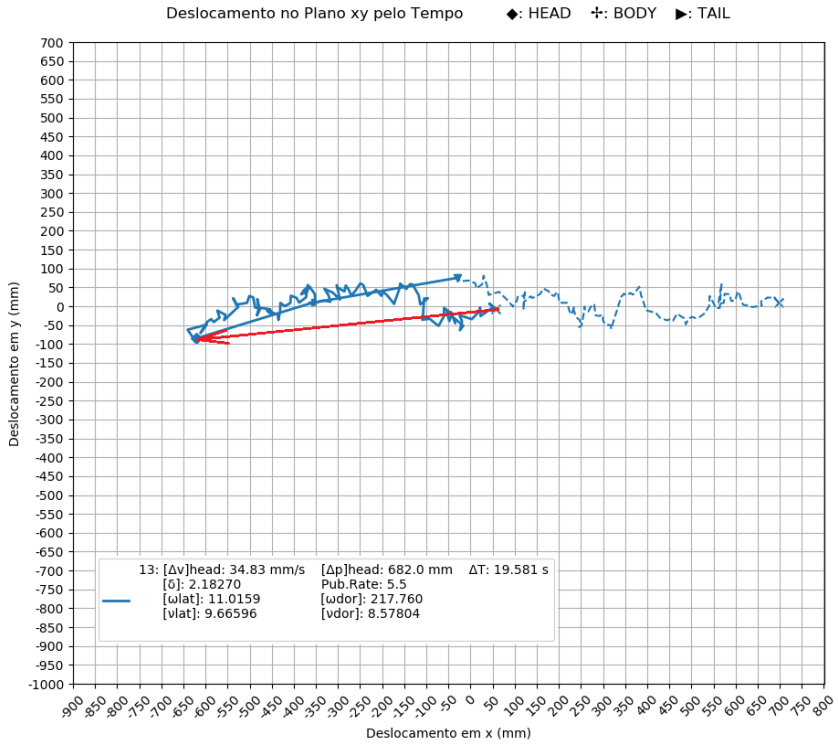

Figura 4.37 – Trajetória retilínea (2).

4.3.1.2 Sidewinding

O sidewinding é uma técnica avançada que permite ao robô navegar em terrenos desafiadores, como areia solta ou terrenos inclinados.

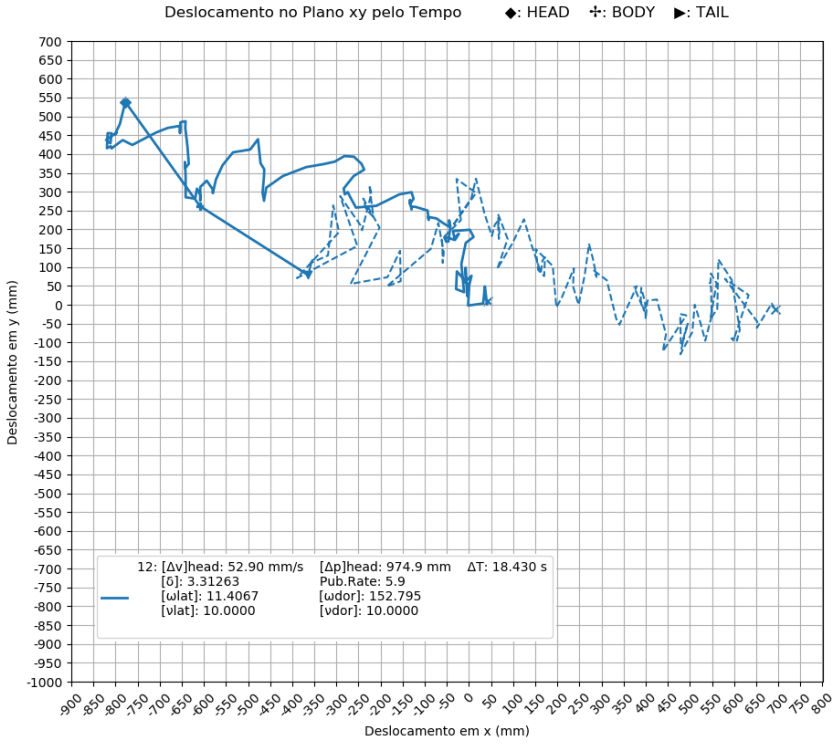

Figura 4.38 – Trajetória de sidewinding (1).

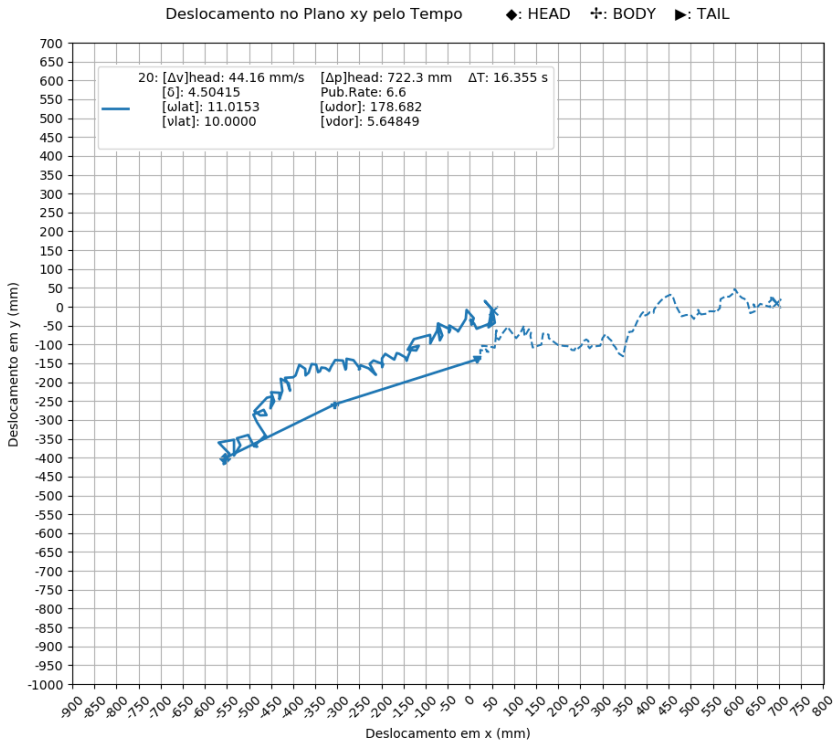

Figura 4.39 – Trajetória de sidewinding (2).

4.3.1.3 Curva

Movimentos curvos são essenciais para manobrar em espaços confinados ou quando viradas bruscas são necessárias.

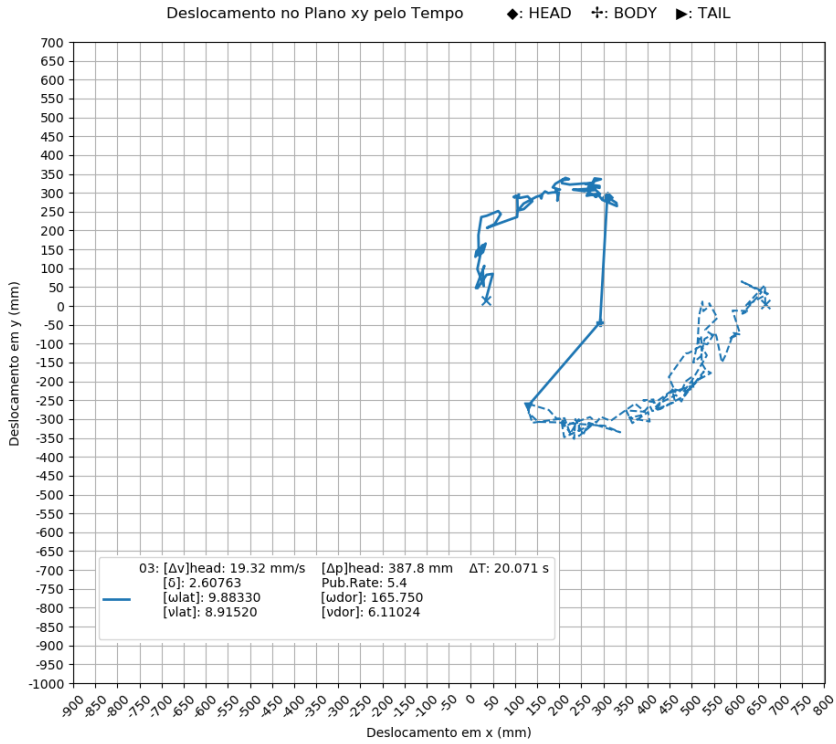

Figura 4.40 – Trajetória curva (Sentido Horário).

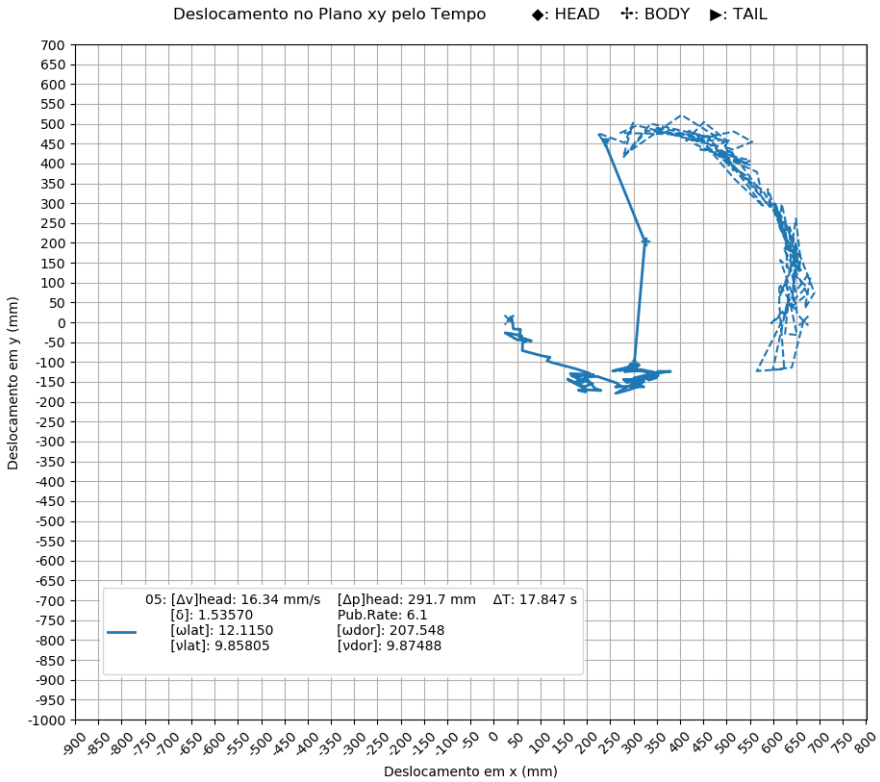

Figura 4.41 – Trajetória curva (Sentido Anti-horário).

Estas imagens e análises proporcionam uma compreensão abrangente das capacidades do robô e o potencial para manobras complexas utilizando a conexão ondulatório pitch-yaw.

5 Conclusões

Este trabalho apresentou um estudo em simulação da locomoção de robôs-serpente, concentrando-se em modelos yaw-yaw, pitch-pitch e a combinação pitch-yaw.

Para os modelos de onda plana (yaw-yaw e pitch-pitch), a otimização Bayesiana provou ser uma ferramenta poderosa na seleção de parâmetros ótimos, que melhoram a eficiência do deslocamento, e também aumentam a estabilidade durante o movimento, excluindo a escolha de parâmetros que tenham risco de tombamento. A análise do atrito anisotrópico destacou sua influência na adaptabilidade dos robôs a diferentes superfícies, ou seja, para diferentes superfícies o robô se locomoveu, para algumas mais e outras menos, mas no geral, se locomoveu. Sugerindo um caminho promissor para a otimização de movimentos em ambientes variáveis.

Para o modelo composto foram implementadas e simuladas diversas modalidades de locomoção, incluindo movimentos retos, laterais e rotativos. Concluiu-se que a locomoção dos robôs-cobra pode ser controlada por um conjunto de parâmetros fundamentais, como amplitude, offset, diferença de fase entre juntas adjacentes e a diferença de fase entre módulos horizontais e verticais.

As simulações demonstraram a viabilidade de múltiplos padrões de movimento, comprovando os princípios de locomoção dos robôs-serpente modulares com conexão pitch-yaw. Foi estabelecido que a diferença de fase é um parâmetro crucial na determinação das características dos movimentos, possibilitando a transição fluida entre diferentes tipos de movimentações.

5.1 Implicações para Aplicações em Ambientes Reais

As técnicas e estratégias desenvolvidas neste estudo têm implicações para a implementação de robôs-serpente em cenários do mundo real, para utilizá-las, basta ajustar aos parâmetros de projeto (como a taxa de publicação, a velocidade de rotação máxima do motor, entre outros fatores). A capacidade de tais robôs para manobrar em terrenos complexos e restritos pode ser crucial em missões de busca e resgate, inspeções de infraestrutura e aplicações em saúde como cirurgias minimamente invasivas. A versatilidade dos movimentos estudados, particularmente os movimentos retos, curvos e diagonais, abre novas possibilidades para a navegação autônoma e adaptativa da nova série de robôs do Laboratório Ereko (ErekoChainBot).

5.2 Trabalhos Futuros

Finalmente, existem várias direções para pesquisas futuras. Isso inclui a exploração de novos modelos de atrito, a integração de sensores e feedback em tempo real para ajuste dinâmico dos parâmetros de movimento e a aplicação de aprendizado de máquina para otimizar ainda mais os algoritmos de locomoção. Além disso, a transição das simulações para protótipos físicos e testes em ambientes externos representaria um passo fundamental para validar as técnicas propostas neste trabalho.

Referências

AUPPERLEE, A. Cmu's snakebot goes for a swim. **Carnegie Mellon University News**, 2021. Disponível em: https://www.cmu.edu/news/stories/archives/2021/april/snake-robot.html. Citado na p. 20.

BISHOP, C. M. **Pattern Recognition and Machine Learning**. [*S.l.*]: Springer, 2006. Citado na p. 38.

DAI, J.; TRAVERS, M.; DEAR, T.; GONG, C.; ASTLEY, H. C.; GOLDMAN, D. I.; CHOSET, H. Robot-inspired biology: The compound-wave control template. *In*: IEEE. **2015 IEEE International Conference on Robotics and Automation (ICRA)**. Seattle, Washington, 2015. p. Sequência de páginas do artigo. Citado na p. 31.

DAVIS, E. M.; WALTERS, S. T. Adaptive strategies: Sidewinding motion in sandy environments. **Journal of Desert Reptilian Studies**, v. 28, p. 56–72, 2023. Citado na p. 19.

EREKO, G. **Relatório Técnico Sigma Lily**. [*S.l.*], 2015. 84 p. Citado na p. 23.

FRAZIER, P. I. A tutorial on bayesian optimization. **arXiv preprint arXiv:1807.02811**, 2018. Citado na p. 38.

GALEMBECK, V. **Modelagem de Anisotropia para Movimentação de Robôs-serpente**. 2018. Disponível em: link_para_o_documento. Citado nas pp. 33 e 34.

Gazebo Tutorials. **Physics Parameters**. 2023. https://classic.gazebosim.org/tutorials?tut=physics_params&cat=physics. Acessado em 18/12/2023. Citado na p. 29.

GóMEZ, J. G. **Modular Robotics and Locomotion: Application to Limbless Robots**. Tese (Doutorado) — Universidad Carlos III de Madrid, 2008. Citado nas pp. 30 e 36.

HIBBELER, R. C. **Fundamentos de Mecânica**. 16. ed. [*S.l.*]: Pearson Education, 2010. 182 p. Citado na p. 29.

HIROSE, S.; YAMADA, H. Snake-like robots [tutorial]. **IEEE Robotics and Automation Magazine**, v. 16, n. 1, p. 88–98, 2009. Citado nas pp. 14 e 23.

HOPKINS, J. K.; SPRANKLIN, B. W.; GUPTA, S. K. A survey of snake-inspired robot designs. **Bioinspiration & biomimetics**, v. 4, n. 2, p. 021001, 2009. Citado na p. 14.

LI Y., Z. Y. Z. Y. L. Q.; YU, G. The effect of friction coefficients on robot locomotion. **Robotics and Autonomous Systems**, v. 112, p. 113–123, 2017. Citado na p. 29.

LILJEBäCK, P.; PETTERSEN, K. Y.; STAVDAHL, ; GRAVDAHL, J. T. **Snake Robots Modelling, Mechatronics, and Control**. Industrial Control Centre, Glasgow, Scotland, UK: Springer, 2013. Citado nas pp. 28 e 29.

LIU, J.; TONG, Y.; LIU, J. Review of snake robots in constrained environments. **Robotics and Autonomous Systems**, p. 103785, 05 2021. Citado na p. 14.

MILLER, A. R.; THOMPSON, H. W. Analysis of rectilinear crawling in large snakes. **International Journal of Herpetology**, v. 45, p. 89–97, 2023. Citado na p. 19.

MURPHY, K. P. **Machine Learning: A Probabilistic Perspective**. [*S.l.*]: MIT Press, 2012. Citado na p. 38.

PRADA, E.; MIKOVá, ; SUROVEC, R.; KENDEROVá, M. Complex kinematic model of snake-like robot with holonomic constraints. *In*: . [*S.l.: s.n.*], 2012. Citado nas pp. 18 e 19.

QIAO, G.; WEN, X.; LIN, J.; WANG, D.; WEI, Z. Sigmoid transition approach of the central pattern generator-based controller for the snake-like robot. **International Journal of Advanced Robotic Systems**, v. 14, p. 172988141770528, 05 2017. Citado nas pp. 25, 26 e 27.

ROCHA, T. d. D. L. Robô cobra multimodular: construção, controle e movimentação. No prelo. 2023. Citado na p. 35.

ROLLINSON, D. **Control and Design of Snake Robots**. Tese (Doutorado) — School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, June 2014. Thesis Committee: Howie Choset (Chair), Chris Atkeson, George Kantor, Art Kuo (University of Michigan). Disponível em: https://ri.cmu.edu/pub_files/2014/6/d_rollinson_robotics_2014.pdf. Citado nas pp. 30 e 31.

SCIKIT-OPTIMIZE. https://scikit-optimize.github.io/stable/. Accessed: [28/11/2023]. Citado na p. 38.

SMITH, J. K.; JOHNSON, L. P. Mechanics of concertina locomotion in narrow environments. **Journal of Reptilian Motion Studies**, v. 30, p. 102–115, 2023. Citado na p. 18.

SOCHA, J. J. Kinematics: Studies of climbing in snakes. **Journal of Experimental Biology**, v. 205, p. 1819–1827, 2002. Citado na p. 18.

STURMAN, M.; KELLY, R.; HOWARD, A. The robot operating system: A survey. **IEEE Robotics and Automation Magazine**, v. 19, n. 4, p. 84–98, 2012. Citado na p. 36.

TRAMSEN, H. T.; GORB, S. N.; ZHANG, H.; MANOONPONG, P.; DAI, Z.; HEEPE, L. Inversion of friction anisotropy in a bio-inspired asymmetrically structured surface. **Journal of The Royal Society Interface**, v. 15, n. 138, p. 20170629, 2018. Disponível em: https://royalsocietypublishing.org/doi/abs/10.1098/rsif.2017.0629. Citado na p. 21.

YANG, X.; ZHENG, L.; Lü, D.; WANG, J.; WANG, S.; SU, H.; WANG, Z.; REN, L. The snake-inspired robots: a review. **Assembly Automation**, v. 42, 07 2022. Citado na p. 18.