

CENTRO UNIVERSITÁRIO SENAI CIMATEC
GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**ANÁLISE, PROTOTIPAGEM E CONTROLE DE LOCOMOÇÃO DE UM
ROBÔ QUADRÚPEDE**

Brenda Silva de Alencar¹
Felipe Mohr Santos Muniz Barreto²
Lucas Lins Souza³
Orientador: Paulo Andrade Souza⁴
Coorientador: Marco Antônio dos Reis⁵

¹Instituição: Centro Universitário SENAI CIMATEC, E-mail: brenda.s1602@outlook.com;

²Instituição: Centro Universitário SENAI CIMATEC, E-mail: felipe18mohr@gmail.com;

³Instituição: Centro Universitário SENAI CIMATEC, E-mail: lucaslinssouza@gmail.com;

⁴Instituição: Centro Universitário SENAI CIMATEC, E-mail: paulo.andrade@fieb.org.br;

⁵Instituição: Centro Universitário SENAI CIMATEC, E-mail: marcoreis@me.com;

**ANALYSIS, PROTOTYPING AND LOCOMOTION CONTROL OF A
QUADRUPED ROBOT**

Resumo: O resumo deve ser digitado em fonte Times New Roman / Arial tamanho 12, itálico, e dentro desta área. O texto deve ser justificado. O resumo deve conter o objetivo e justificativa do trabalho, sua importância e as principais conclusões. Não deve ultrapassar 250 palavras. Vide NBR 6028

Palavras-Chaves: *deve ter no mínimo 3 palavras chaves e no máximo 5 palavras separadas por ponto e vírgula;*

Abstract: O resumo deve ser digitado em fonte Times New Roman / Arial tamanho 12, itálico, e dentro desta área. O texto deve ser justificado. O resumo deve conter o objetivo e justificativa do trabalho, sua importância e as principais conclusões. Não deve ultrapassar 250 palavras. Vide NBR 6028

Keywords: *deve ter no mínimo 3 palavras chaves e no máximo 5 palavras separadas por ponto e vírgula;*

1 INTRODUÇÃO

Com o avanço da robótica, robôs móveis estão cada vez mais tomando espaço em setores chave da economia como o comercial, industrial e militar. Os robôs terrestres que se locomovem com pernas têm se mostrado mais eficientes para se locomover em terrenos irregulares, inclinados e escorregadios e também para superar obstáculos [1]. Quando comparado com robôs com rodas, robôs com pernas ainda possuem melhor mobilidade e manobrabilidade em ambientes complexos, possibilitando transitar por caminhos não necessariamente contínuos. Entre os robôs com pernas, os quadrúpedes vêm ganhando destaque por apresentarem maior estabilidade e uma estrutura mais simples que os bípedes e hexápodes [2].

O BigDog [3], desenvolvido pela Boston Dynamics em 2004, foi um dos primeiros robôs quadrúpedes a navegar em terrenos irregulares, incluindo trilhas e solos com neve ou lama. Desde então, as atuais soluções disponíveis no mercado apresentam maior maturidade de tecnologia. O Spot, por exemplo, é um robô quadrúpede para missões *indoor* e *outdoor* fabricado pela empresa *Boston Dynamics*. Ele é capaz de alcançar uma velocidade de até $1,6m/s$ e se equilibrar em ambientes incertos carregando até $14kg$ [4]. Semelhante ao Spot, o Anymal, desenvolvido pela *Anybotics*, consegue atuar de forma autônoma em ambientes difíceis. Ele possui $70cm$ de altura e apresenta uma capacidade de *payload* que comporta simples sensores até manipuladores [5].

O uso de uma plataforma com quatro pernas requer um sistema de locomoção robusto que envolve o controle de equilíbrio da plataforma, o controle das juntas do robô e o planejamento de marchas. Para poder aplicar todos esses conceitos e desenvolver aplicações reais com robôs quadrúpedes, é fundamental estudar sua estrutura física, os tipos de marcha que eles podem realizar e os métodos de controle de locomoção que são utilizados nesses

robôs.

O objetivo deste trabalho é desenvolver um sistema robótico do tipo quadrúpede capaz de se locomover de forma estável quando teleoperado. Serão estudados aspectos construtivos, de locomoção e de controle desse tipo robô. Além disso, será projetado um protótipo que deve ser simulado e fabricado, a fim de testar os algoritmos de locomoção não apenas em um ambiente virtual, mas também na prática. Por fim, serão realizados testes de performance de locomoção com o protótipo físico, os quais analisarão a estabilidade do robô em ambientes de terreno plano e irregular.

Com o propósito de guiar o desenvolvimento do projeto, foram elencados alguns requisitos que o robô deve cumprir:

- Ser capaz de operar por tempo suficiente para inspecionar um ambiente de $50m^2$;
- Ser capaz de se locomover em ambientes irregulares;
- Ser capaz de transpor obstáculos pequenos (máx. $5cm$);
- Ser capaz de operar em ambientes *indoor* e *outdoor*;
- Ser capaz de ser transportado por apenas uma pessoa;
- Ter capacidade de *payload* de $2kg$.

A próxima seção tratará dos conceitos estudados para o desenvolvimento do robô. A seção 3 tratará da metodologia utilizada no trabalho. Na seção 4, será apresentado o Caramelo, o robô que foi desenvolvido para essa pesquisa. A seção 5 mostrará os resultados do trabalho e a seção 6 a conclusão e as considerações finais.

2 O ROBÔ QUADRÚPEDE

O robô quadrúpede é um sistema robótico móvel que se locomove com a ajuda de pernas. Quando comparado a robôs terrestres que utilizam outros meios de locomoção como rodas ou esteiras, ele apresenta diversas particularidades que o confere muitas vantagens quanto a mobilidade, robustez a dife-

rentes terrenos e superação de obstáculos [6]. Robôs com rodas e esteiras conseguem navegar pelo espaço desde que haja um caminho contínuo entre os pontos de origem e de destino. Robôs com pernas, por outro lado, são capazes de escolher os melhores pontos no terreno para apoiar suas patas, o que permite uma navegação em caminhos discretos (com obstáculos de grande inclinação e variação de altura) [7]. Essa capacidade de se adaptar a terrenos desnivelados favorece sua aplicação em diversos setores: industriais, militares, missões de inspeção e de resgate. Por outro lado, esse tipo de robô apresenta menor estabilidade de locomoção e, por consequência, maior complexidade de controle.

Robôs com pernas também apresentam diversas diferenças entre si, majoritariamente ligadas à quantidade de pernas que possuem. A quantidade de pernas de um robô está diretamente relacionada a sua estabilidade, capacidade de locomoção e eficiência. Os bípedes possuem baixa estabilidade, visto que se apoiam em apenas uma perna para poder andar. Os com múltiplas pernas (mais de quatro) possuem maior estabilidade, afinal conseguem manter pelo menos três pontos de apoio no solo enquanto realizam um passo. No entanto, cada perna representa um conjunto adicional de juntas e atuadores, diminuindo a eficiência do sistema como um todo. Os quadrúpedes conseguem unir vantagens desses dois tipos ao apresentar um balanço entre estabilidade e eficiência. Eles possuem uma estabilidade passiva quando estáticos, pois se apoiam em quatro pontos. Além disso, também são capazes de navegar de forma estável em baixas velocidades, movendo uma perna por vez enquanto as outras três permanecem no solo. Isso elimina a redundância presente nos robôs com múltiplas pernas, aumentando sua eficiência [7].

2.1 Estrutura e design

Pelo fato de robôs quadrúpedes terem se tornado um grande foco de pesquisa nos

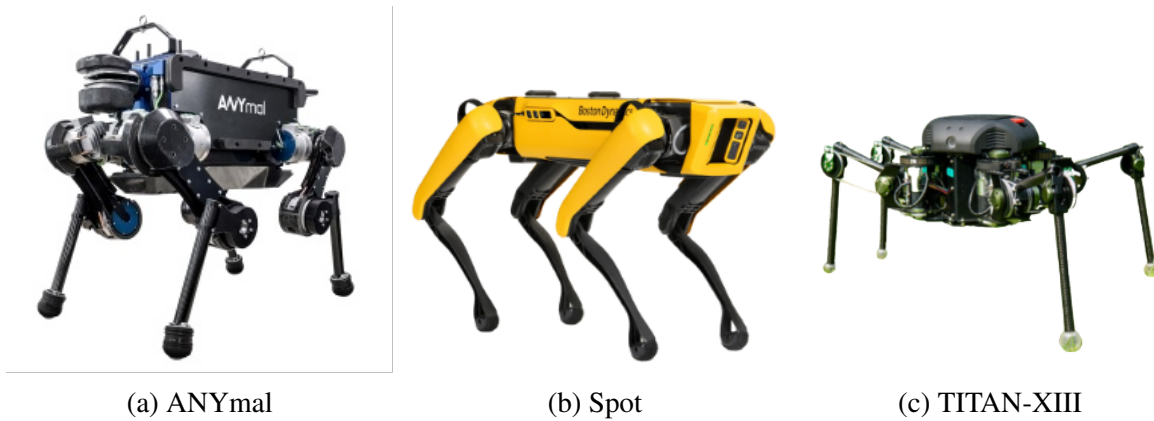
últimos anos, diferentes *designs* já foram pesquisados. Esses *designs* se diferenciam em aspectos como estrutura, configuração de pernas e o número de graus de liberdade (GDL) por perna.

Um dos tipos de estrutura que existem é a de tipo mamífero. A estrutura tipo mamífero tem esse nome por conta da sua semelhança com a postura de quadrúpedes como cachorros e cavalos. Kitano et al., em [10], analisa dois tipos diferentes de estruturas de robôs quadrúpedes: a do tipo mamífero (Figuras 1a e 1b) e a do tipo *sprawling* (Figura 1c). Segundo sua análise, a primeira permite alcançar maiores velocidades por possuir duas juntas no plano sagital. Além disso, ela também é mais eficiente, pois os atuadores requerem menos torque para sustentar o robô: sua estrutura mais compacta diminui o braço de alavanca sobre o qual a força peso do robô atua. Essa estrutura também favorece a navegação em ambientes estreitos, onde um robô do tipo *sprawling*, por exemplo, teria dificuldades de acessar.

Os robôs quadrúpedes que utilizam essa estrutura também se diferenciam quanto à configuração das pernas. Os quatro tipos de configuração podem ser vistos na Figura 2. Entre elas destacam-se a *full-elbow* e o *elbow-knee*. Robôs como o Spot, MIT Cheetah e Stanford Pupper utilizam a configuração *full-elbow*, enquanto outros como o ANYmal, StarlETH e BigDog adotam a configuração *elbow-knee*. Yao et al., em [7], acreditam que a configuração *elbow-knee* possibilita maior estabilidade, mas as características de movimento da configuração *full-elbow* podem ser superiores.

O número de juntas nas pernas, que coincide com a quantidade de GDL do robô, também é um dos aspectos estudados sobre os quadrúpedes. A maioria apresenta 3 GDL por perna, o que é suficiente para que o robô consiga mover suas patas em três dimensões e realize diversos tipos de marchas. A fim de simplificar a estrutura e consequentemente o

Figura 1: Exemplos de robôs com estruturas do tipo mamífero e *sprawling*.



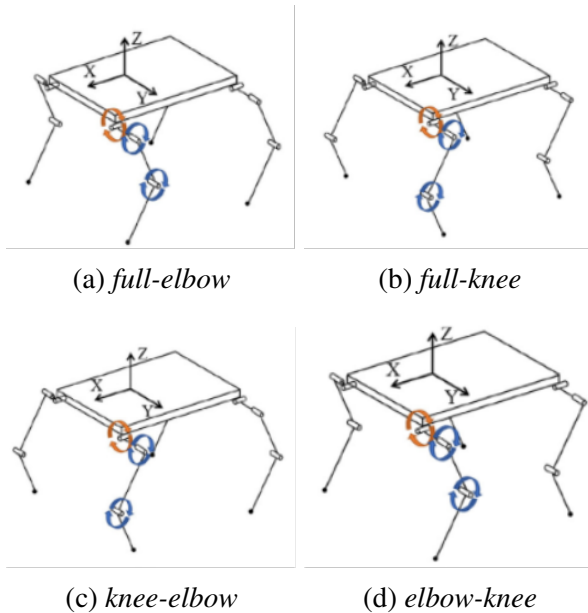
(a) ANYmal

(b) Spot

(c) TITAN-XIII

Fonte: Adaptado de [8] [9] [10].

Figura 2: Tipos de configuração de pernas para robôs com estrutura tipo mamífero.



(a) full-elbow

(b) full-knee

(c) knee-elbow

(d) elbow-knee

Fonte: [7]

controle, alguns robôs utilizam apenas 2 GDL por perna, eliminando a junta no corpo que movimenta a perna no plano frontal. Outros robôs buscam performances mais semelhantes ao andar de animais reais, o que demanda maior flexibilidade de movimento, justificando o acréscimo de uma quarta junta. No entanto, como já mencionado, esse acréscimo aumenta a complexidade do controle e prejudica a eficiência. Essa perda de eficiência se dá porque mais atuadores significa maior con-

sumo de energia e também mais massa.

A massa do robô quadrúpede deve ser a menor possível. Quanto mais leve for o sistema, menos torque será demandado dos motores e maior sua eficiência. Além disso, a distribuição de massa do robô também é um aspecto muito importante. A massa deve ser localizada majoritariamente no corpo, enquanto as pernas devem possuir baixa inércia. Isso permite que elas se movam rapidamente sem alterar, de forma significativa, o centro de gravidade do robô, o que aumenta a estabilidade e requer menos complexidade de controle. Possuir baixa inércia significa possuir baixa massa. Por outro lado, as pernas devem ser resistentes o suficiente para suportar o peso do robô, além dos distúrbios causados pelo impacto das patas com o chão, o que pode demandar um aumento de massa nas pernas. Portanto, um equilíbrio entre massa e resistência deve ser buscado ao mesmo tempo em que deve-se buscar diminuir a massa total do sistema [11].

2.2 Movimentação por marchas

Robôs quadrúpedes se movimentam conforme uma sequência de movimentos coordenados de suas pernas que compõe uma marcha. Uma marcha é definida pelo tempo e local de colocação e levantamento de cada pata, coordenado com o movimento do corpo em seus

seis graus de liberdade, para mover o corpo de um lugar para outro [12].

A marcha é um aspecto fundamental para garantir que um robô com pernas caminhe de forma eficiente e estável, especialmente em terrenos irregulares [13]. Para isso, é necessário levar em consideração suas etapas e, consequentemente, seu tipo.

Marchas são divididas em duas etapas: *stance* e *swing*. Durante a fase *stance*, as pernas estão no solo e impulsionam o robô para frente. Na etapa de *swing*, as pernas são erguidas para deslocar a pata até o próximo ponto de apoio. É importante ressaltar que as fases de *stance* e *swing* não ocorrem em todas as pernas ao mesmo tempo. A depender do tipo de marcha, algumas pernas podem estar em *swing* enquanto outras estarão em *stance*.

O *trot* é um tipo de marcha muito utilizado por robôs quadrúpedes devido a sua simplicidade e eficiência. Este tipo de marcha é periódico e simétrico. Marchas periódicas são caracterizadas pelo fato de que os mesmos movimentos sempre se repetem no mesmo instante dentro de um ciclo de locomoção [14]. Já a simetria é uma característica de marchas que movimentam um par de pernas em conjunto, saindo e voltando para o solo de forma sincronizada. No *trot*, as pernas diagonais se movimentam em pares e quando um par está na etapa de *swing* o outro está na etapa de *stance*. Outra característica da marcha *trot* é que ela pode ser contínua ou descontínua. Uma marcha contínua mantém o corpo do robô em movimento constante, enquanto a descontínua submete o corpo a um movimento intermitente [14]. Portanto, quando a marcha *trot* é contínua, as pernas em *stance*, além de sustentar o robô, deslocam o corpo na direção do movimento, o que exige maior capacidade de controle. Em contrapartida, quando ela é descontínua o corpo fica estático esperando as pernas em *swing* terminarem seu movimento para, então, ser deslocado quando as quatro pernas já estão no solo.

2.3 Controle de locomoção

Todo o controle de locomoção do robô é realizado pelo planejador de marchas. Ele é o responsável por enviar os comandos para que as pernas se movam para os locais desejados no momento esperado. Logo, o planejador irá apenas ditar o ponto no espaço no qual cada pata do robô deve estar, em relação a um eixo de referência, cabendo ao sistema de controle executar o movimento. A seguir, serão discutidos dois itens fundamentais do sistema de controle de um robô quadrúpede: o modelo cinemático e as estratégias de controle.

2.3.1 Modelo cinemático

O modelo cinemático de um robô quadrúpede descreve a relação entre a posição de uma pata em três dimensões com a rotação de cada junta da sua respectiva perna. Como todas as pernas do robô são iguais, pode-se formular as relações de apenas uma perna e replicá-la quatro vezes, acrescentando as devidas translações e rotações, para se ter o modelo cinemático de todo o sistema.

O modelo cinemático pode ser usado para resolver dois problemas: a cinemática direta e a cinemática inversa. A cinemática direta fornece a posição de uma pata em (x, y, z) em função dos ângulos das juntas, enquanto que a cinemática inversa fornece o ângulo das juntas que corresponde a uma posição da pata do espaço tridimensional. Esses dois problemas são complementares, sendo a saída de um a entrada do outro e vice-versa. A partir da cinemática inversa, o robô consegue determinar quanto deve rotacionar seus atuadores para mover a pata alguma distância nas direções (x, y, z) . Com a cinemática direta, é possível saber se a pata de fato chegou na posição em que ele deveria estar. Portanto, ambos são muito importantes para o controle de locomoção do robô.

2.3.2 Estratégias de controle

A locomoção de robôs quadrúpede, em geral, segue uma sequência de passos. Raibert propôs em [15] um método de controle baseado em três etapas: controle de salto, controle de velocidade e controle de postura do corpo. Essa estratégia de controle foi utilizada para controlar robôs com uma, duas e quatro pernas (o motivo de se ter um controle de salto é que robôs com apenas uma perna só podem ser locomover saltando). Sua premissa básica era a de que apenas uma perna estaria em *stance* ou em *swing* por vez. A fim de satisfazer essa premissa para robôs com mais de duas pernas, foi proposto o conceito de pernas virtuais. Isto é, um conjunto de pernas deve realizar igual comportamento quando em *swing* e *stance* e as fases de *swing* e *stance* de cada conjunto devem ser alternadas. Esse conceito foi utilizado para embasar o uso de marchas simétricas e periódicas como o *trot*.

Essa estratégia de controle em três etapas foi responsável por locomover robôs com pernas rígidas (apenas 2 GDL por perna, sendo uma junta rotativa e outra prismática) de maneira simples, porém esses robôs apenas operavam no terreno plano e controlado do laboratório. A fim de possibilitar a operação de robôs com pernas em terrenos desnivelados e de difícil mobilidade, Raibert *et. al.* propôs um outro sistema de controle no seu trabalho sobre o BigDog [3]. O BigDog é um robô quadrúpede com 4 GDL por perna movido por atuadores hidráulicos. Essa maior flexibilidade de movimentação das pernas permite controlar a locomoção do robô sem que este precise saltar, sendo possível, então, dividir o controle de locomoção em duas etapas principais: controle de *stance* e controle de *swing*. Como o nome sugere, o controlador de *stance* é o responsável por controlar o comportamento das pernas na fase de *stance*, enquanto o controlador de *swing* é o responsável por controlar as pernas em *swing*. Vale lembrar que durante a marcha, algumas pernas podem estar na etapa de *swing* enquanto outras estão da etapa de

stance, o que significa que esses controladores ora assumem o comando de um par de pernas, ora do outro (essa troca não necessariamente se dá em pares, porém isso é válido para marchas simétricas como o *trot*). Quem define o momento em que cada controlador assume o controle de uma determinada perna é o planejador de marchas.

O modo como cada perna se comporta durante as etapas de *stance* e *swing* pode variar em diversos aspectos, mais ainda é possível elencar semelhanças gerais. No início da etapa de *swing*, calcula-se o local do próximo ponto de apoio das patas com base na velocidade desejada para o robô e uma trajetória de passo até esse ponto. Essa trajetória pode ter o formato de uma curva senoidal [16], triangular [17], de Bezier [18], cicloidial [2] [19], entre outras. Uma consideração comum é a de que as trajetórias das pernas em cada etapa são independentes, ou seja, assumem que as outras pernas e o corpo se comportam de forma ideal. Sendo assim, é necessário controlar a força com que a pata toca o solo, pois este é um ponto chave para a estabilidade do quadrúpede [16]. Nesse sentido, a trajetória cicloidial ganha destaque por conta de sua primeira derivada nula no momento em que se aproxima do seu ponto mínimo.

Já na fase de *stance*, as pernas devem manter o robô em equilíbrio, além de deslocar o corpo na direção desejada de locomoção. Para isso, alguns robôs utilizam trajetórias para as patas com um formato pré-determinado assim como na fase de *swing*, não necessariamente repetindo o mesmo formato de curva [16] [19]. Além disso, controladores de equilíbrio também podem ser implementados nessa etapa. Esses controladores visam estabilizar os ângulos de *pitch* e *roll* do robô [2] [17] [18] [20] ou até ainda outros graus de liberdade [1] [21] [22]. Eles podem controlar diretamente a angulação do corpo do robô (com o auxílio de um sensor inercial) e/ou a força de contato com o solo em cada perna, por exemplo. No entanto, alguns trabalhos se

baseiam apenas no controle individual de cada junta para manter o robô em equilíbrio, o que é uma abordagem mais simples, mas que pode falhar especialmente em terrenos irregulares.

3 METODOLOGIA

Para o desenvolvimento do projeto, foi criada uma metodologia própria composta por três etapas sequenciais. A seguir, serão detalhadas cada uma dessas etapas.

3.1 Fundamentação

A primeira etapa teve como objetivo buscar referências para embasar o estudo os robôs quadrúpedes e servir de padrão para o desenvolvimento do projeto.

Primeiramente, foi realizada uma pesquisa bibliográfica com o intuito de selecionar artigos relacionados com o projeto. Para realizar essa busca, foi utilizado o método *Bibliographic and Literary Review Method (BiLi)*, que é um método de busca iterativa de artigos, guiada por análises estatísticas das palavras-chave e da rede de co-citação entre os trabalhos encontrados [23].

A partir dessa pesquisa, foi possível realizar um estudo do estado da arte com o objetivo de entender a teoria de locomoção dos robôs quadrúpedes. Além disso, foi feito com um *benchmarking* de projetos *open source* desse tipo de sistema robótico, objetivando encontrar referências para a definição do conceito do robô que será projetado e, consequentemente, para a prototipação dele. Neste *benchmarking*, foram buscados características que indicassem a viabilidade de prototipação como a disponibilidade do código na internet, linguagem de programação utilizada, disponibilidade do modelo 3D, se foi feito em impressão 3D ou não, tipo dos atuadores, sensores que foram utilizados, entre outras.

3.2 Desenvolvimento

Durante esta etapa, ocorreu o desenvolvimento do robô propriamente dito. Foram

criadas duas frentes de trabalho que ocorreram em paralelo: a implementação e simulação dos algoritmos em ambiente computacional e a criação do *design* junto com a prototipação do robô.

Durante a implementação dos algoritmos na simulação, foram elaborados, primeiramente, o diagrama de blocos do sistema e a arquitetura de *software* do controlador. A partir disso, os algoritmos de cinemática e controle foram implementados e testados na simulação. Para auxiliar o desenvolvimento do projeto, algumas ferramentas de *softwares* serão utilizadas como o *ROS2 Humble* e o simulador *Gazebo*.

Em paralelo ao desenvolvimento do *software*, foi realizado o *design* e a prototipação do robô através da elaboração dos desenhos mecânicos e do projeto eletroeletrônico. Esta etapa incluiu também a impressão 3D das partes mecânicas, o teste dos atuadores e sensores e a configuração da comunicação entre a central de processamento e os atuadores, culminando na montagem física do protótipo. Durante essa etapa, o *software OnShape* foi utilizado para a modelagem 3D do robô e o *QElectroTech* para o projeto eletroeletrônico.

Após essas duas etapas, foi possível realizar a integração dos algoritmos no protótipo físico. Ao final da integração, o protótipo já estava funcional e pronto para a fase de testes.

3.3 Resultados e análises

Durante a etapa de resultados e análises, foram feitos alguns testes para avaliar de forma estatística a performance de locomoção do robô. Também procurou-se avaliar o cumprimento dos requisitos definidos para o protótipo.

O primeiro experimento realizado consistiu em testar a capacidade do protótipo de mover a pata por uma trajetória cicloidial, feita pelo planejador de trajetórias. Para isso, o corpo do robô foi deixado em um suporte

estático, de modo que as patas não tocassem o chão, diminuindo a carga sobre os motores. O objetivo principal deste teste foi verificar se o sistema de controle do robô era capaz de responder de forma coerente aos comandos enviados pelo planejador de marchas numa situação mais próxima da ideal (sem carga). As trajetórias foram calculadas considerando a altura de passo $0,05m$, período de $0,5$ segundos e uma resolução de 25 pontos distribuídos de forma homogênea ($P_T = P_N = 1, 0$).

O segundo teste teve como objetivo verificar a capacidade do robô de seguir um comando de velocidade pré-estabelecido e de manter a orientação do seu corpo estável em 0° nos ângulos de *roll* e *pitch*. O teste foi realizado em dois tipos de terreno: um chão de cimento plano e um terreno irregular formado por pequenas pedras soltas. Este experimento consistiu em enviar um comando de velocidade constante para frente de $0,05m/s$ e medir o tempo que o robô precisou para percorrer $1,5$ m. Dessa forma, foi possível obter a velocidade média do robô e compará-la com a do comando enviado. Além disso, a fim de avaliar a contribuição do controle de angulação do robô para sua estabilidade, metade dos experimentos foram realizados com esta funcionalidade ativa e a outra não. A estabilidade do robô foi mensurada por meio da oscilação máxima do corpo do robô nos ângulos de *roll* e *pitch*. A trajetória utilizada para o passo possui as mesmas especificações do primeiro experimento, exceto pelos parâmetros de disposição dos pontos, cujos valores adotados foram $P_T = 0,66$ e $P_N = 0,33$. Dessa forma, os testes foram organizados em quatro combinações, variando o tipo do terreno e o uso, ou não, do controle de angulação.

No terceiro experimento, foi avaliada a capacidade do Caramelo de se locomover por terrenos inclinados. Para isso, foi construída uma rampa com uma inclinação de aproximadamente $5,34^\circ$ pela qual o robô deveria subir e descer de forma teleoperada. No quarto teste, por fim, foi proposto que o Caramelo atravessasse

degraus com diferentes alturas, também de forma teleoperada, a fim de encontrar a máxima altura de obstáculo que ele conseguisse superar.

4 O ROBÔ CAMELO

Baseado nos conceitos apresentados, foi projetado, simulado e desenvolvido um robô quadrúpede nomeado de Caramelo (ver Figuras 5, 8 e 11). Caramelo é um robô quadrúpede de pequeno porte voltado para pesquisa e educação. Seu *hardware* foi modelado inteiramente pela equipe e impresso com impressora 3D no material ABS. Os atuadores do robô são servomotores do modelo *dynamixel* MX-28 e sua central de processamento é composta por uma RaspberryPi 4. Além disso, ele conta com um sensor inercial modelo MPU6050, que está instalado no corpo do robô. Este sensor contém um giroscópio e um acelerômetro, o que permite obter a aceleração linear, a velocidade angular e a orientação do corpo do robô. Todo o *software* foi desenvolvido com o ROS2 Humble (*Robot Operation System 2*) [24], que é um *framework* de robótica *open source* com vários recursos disponíveis para facilitar o desenvolvimento de sistemas robóticos.

A estrutura do caramelo é do tipo mamífero e a configuração das pernas é a *full-elbow*. Além disso, possui 3 GDL por perna, o que permite uma grande liberdade de movimentação para as patas. Seu *design* foi pensado para favorecer o balanço de massas entre o corpo e as pernas do robô, ou seja, a maior parte da massa se encontra no corpo ou próxima a ele. Os componentes eletrônicos internos, que abrangem sensores, unidades de processamento e a interface de comunicação com os atuadores, foram dispostos de forma simétrica, a fim de manter o centro de massa o mais próximo do centro do corpo. Os motores (componentes que contribuem com a maior massa para o sistema) foram dispostos o mais próximo possível do corpo. Um destaque para o motor que atua na junta da tíbia, que foi ins-

talado na parte de cima do fêmur com o objetivo de diminuir o momento de inércia da perna. Essa escolha demandou a adição de um sistema de transmissão entre o eixo do motor e a tíbia, formado por uma haste rígida de metal com duas juntas esfera nas extremidades.

A locomoção do Caramelo foi desenvolvida baseada nas marchas periódicas e simétricas. Dessa forma, foi adotada a marcha *trot* como a marcha principal do robô. Embora sua estrutura permita a realização de muitos outros tipos de marcha, neste trabalho, foi considerada apenas o *trot*, devido a sua simplicidade e eficiência. Com o objetivo de diminuir a complexidade do controle de locomoção, foi adotada uma marcha descontínua, ou seja, o corpo do robô se desloca apenas quando todas as patas estão do solo. A sequência de etapas da marcha do Caramelo pode ser vista na Figura 3, na qual as áreas em branco representam a etapa de *swing* e as em cinza a de *stance*. É possível perceber que sempre o mesmo par de pernas diagonais se move no mesmo instante. Entre duas etapas consecutivas de *swing*, há um momento em que todas as patas estão em *stance*, que é quando o corpo do robô é deslocado no sentido desejado de locomoção.

Figura 3: Padrão de movimentação da marcha.

Frontal Direita				
Frontal Esquerda				
Traseira Esquerda				
Traseira Direita				

Fonte: autores.

O sistema de controle do robô é composto por dois subsistemas principais: os controladores individuais de cada junta e os controladores da angulação do corpo do robô. Os controladores das juntas são os próprios controladores PID embarcados nos motores *dynamixel*. Foi utilizada a interface de controle de posição com o atuador, de forma que o *setpoint* de controle enviado para cada motor é o ângulo em radianos para o qual ele

deve rotacionar. O modelo cinemático do robô, apresentado na seção 4.1, é o responsável por mapear não apenas a posição tridimensional de cada pata com a angulação de cada junta, mas também a posição do corpo em seis dimensões: translação e rotação em (x, y, z) . Dessa forma, é possível controlar cada pata e o corpo do robô ao mesmo tempo de forma independente. Os controladores de angulação do corpo são dois controladores PID em paralelo, responsáveis por controlar o ângulo de *pitch* (rotação em y) e o de *roll* (rotação em x).

O planejador de marchas é o responsável por controlar cada pata do robô e, por consequência, o corpo. Ele calcula a trajetória que cada pata deve realizar, com base nas etapas de *stance* e *swing*, e envia o próximo ponto em que cada pata deve estar a uma frequência de 50Hz. Além disso, ele também considera o esforço de controle enviado pelos controladores de angulação, a fim de manter o corpo do robô em 0° a todo momento.

A seguir, serão apresentados o desenvolvimento do modelo cinemático, dos controladores de angulação e da trajetória que cada pata realiza na etapa de *swing*.

4.1 Modelo cinemático

Como dito anteriormente, o modelo cinemático é utilizado para resolver a cinemática inversa e a cinemática direta do robô. Para a cinemática direta, foi utilizado o pacote *tf2*, um recurso disponível no ROS2 que facilita o gerenciamento de transformações entre eixos de coordenadas. A cinemática inversa, por outro lado, foi feita com base em uma análise geométrica. As variáveis θ_1 , θ_2 e θ_3 expressam a posição angular de cada uma das juntas de uma perna do robô e são calculadas com auxílio das equações 1 a 6 em função da posição (x_{IK}, y_{IK}, z_{IK}) desejada para a pata e dos comprimentos L_1 , L_2 e L_3 (figura 4).

$$\theta_1 = \arctan\left(\frac{x_{IK}}{y_{IK}}\right) - \arctan\left(\frac{L_1}{a}\right) \quad (1)$$

$$\theta_2 = \frac{\pi}{2} - \arctan\left(\frac{a}{z_{IK}}\right) - \arctan\left(\frac{\sqrt{1-A^2}}{A}\right) \quad (2)$$

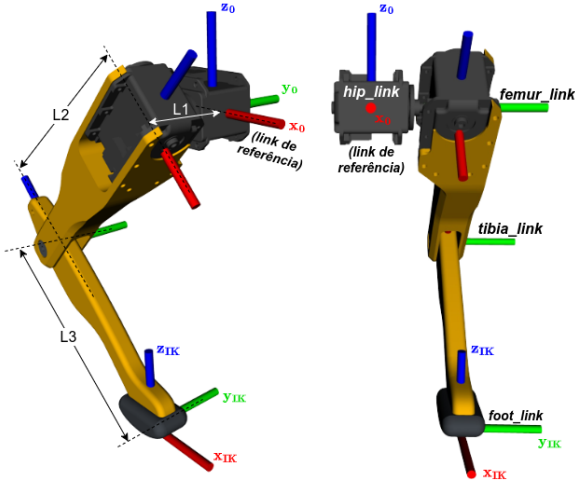
$$\theta_3 = \arctan\left(\frac{\sqrt{1-B^2}}{B}\right) \quad (3)$$

$$a = \sqrt{x_{IK}^2 + y_{IK}^2 - L_1^2} \quad (4)$$

$$A = \frac{a^2 + z^2 + L_2^2 - L_3^2}{2L_2\sqrt{a^2 + z_{IK}^2}} \quad (5)$$

$$B = \frac{a^2 + z_{IK}^2 - L_2^2 - L_3^2}{2L_2L_3} \quad (6)$$

Figura 4: Links da perna do robô.



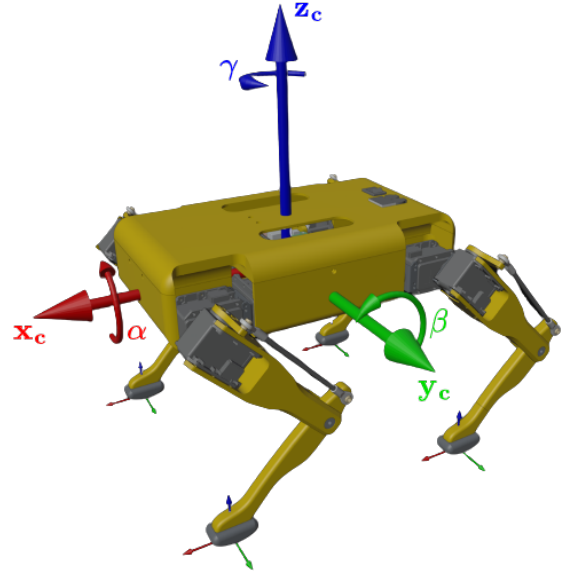
Fonte: autores.

Essas equações são úteis para o cálculo da posição de uma única perna, mas são insuficientes para realizar a cinemática do corpo do robô. Desta forma, um *frame* central, chamado de *base_link* (figura 5), é utilizado como referência, e uma matriz T_M (eqs. 7 e 8) é utilizada para realizar a cinemática do corpo, a partir das translações (x_c, y_c, z_c) e rotações (α, β, γ) desejadas, sendo possível controlar cada um dos 6 graus de liberdade. Para isso, as transformações T_{FR} , T_{FL} , T_{BL} e T_{BR} de cada um dos ombros (*hip_links*) em relação ao *base_link* são necessárias.

$$T_M = \begin{bmatrix} R_{xyz} & x_c \\ & y_c \\ & z_c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

$$R_{xyz} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix} \cdot \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \cdot \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8)$$

Figura 5: Eixos do robô em posição de repouso.



Fonte: autores.

O cálculo das angulações de cada perna então é feito utilizando como entrada os valores (x_{IK}, y_{IK}, z_{IK}) resultantes de cada uma das transformações, conforme a equação 9. O mesmo cálculo é feito para as demais pernas, utilizando T_{FL} , T_{BL} e T_{BR} .

$$\begin{bmatrix} x_{IK} \\ y_{IK} \\ z_{IK} \\ 1 \end{bmatrix} = (T_M \cdot T_{FR})^{-1} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (9)$$

Desta forma, a Cinemática Inversa é capaz de computar as angulações θ_1 , θ_2 e θ_3 de cada uma das pernas a partir da posição (x, y, z) das patas em relação ao link central do robô e às translações (x_c, y_c, z_c) e rotações (α, β, γ) desejadas para o corpo. Entretanto,

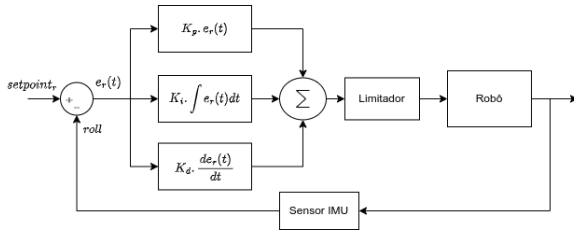
em muitos casos é mais conveniente realizar o cálculo dos ângulos passando como entrada as posições (x, y, z) das patas em relação à sua posição *default*, ou seja, a posição do seu link quando o robô está em seu estado de repouso (figura 5). Para isso, é possível realizar, previamente ao cálculo das angulações, mais uma transformação, desta vez do *base_link* para cada uma das posições *default* das patas (equação 10).

$$\begin{bmatrix} x_{ik} \\ y_{ik} \\ z_{ik} \\ 1 \end{bmatrix} = (T_M \cdot T_{FR})^{-1} \cdot (F_{FR} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}) \quad (10)$$

4.2 Controle de angulação

Os controladores de angulação são dois controladores PID em paralelo responsáveis por controlar a rotação de *roll* e *pitch* do corpo do robô. Eles atuam de forma independente, controlando a rotação do corpo em ambos os eixos simultaneamente. Ambos os controladores são iguais e foram implementados seguindo o modelo apresentado no diagrama de blocos da figura 6.

Figura 6: Controlador PID projetado.



Fonte: autores

O IMU é o sensor responsável por medir a rotação do corpo do robô, possibilitando a realimentação das saídas do sistema. O limitador foi adicionado para evitar que sejam enviados valores que extrapolam os limites de rotação das juntas do robô. Os esforços de controle são enviados o planejador de marchas que, por sua vez, envia os comandos de movimentação para os controladores das juntas.

4.3 Planejador de trajetória

O planejador de trajetória é responsável por calcular a trajetória que cada pata deve realizar, tanto na fase de *stance* quanto na fase de *swing*. Para o caramelo, a trajetória é uma curva cicloidal em ambas as etapas. Como apresentado em [2], uma curva cicloidal pode ser definida no espaço tridimensional entre os pontos (x_o, y_o, z_o) e (x_f, y_f, z_f) em função do tempo t pelas equações (11) a (14), sendo H a altura do passo e T o período.

$$x = (x_f - x_o) \frac{K - \sin(K)}{2\pi} + x_o \quad (11)$$

$$y = (y_f - y_o) \frac{K - \sin(K)}{2\pi} + y_o \quad (12)$$

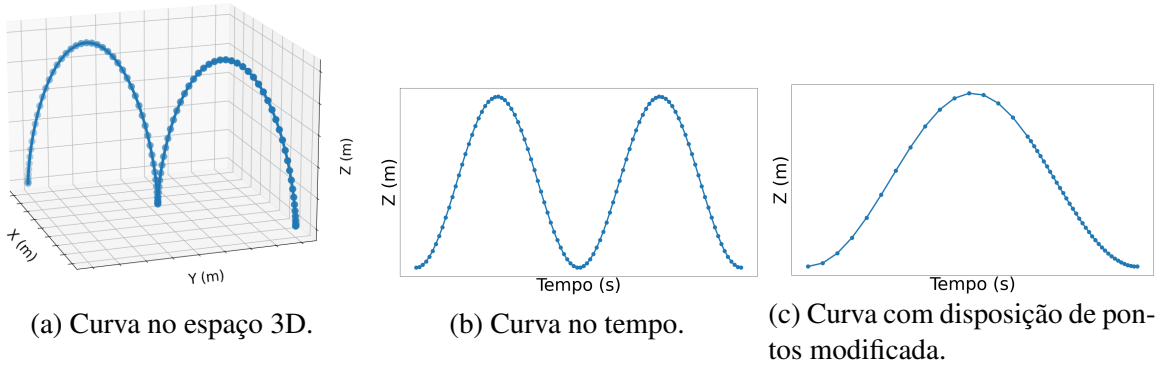
$$z = H \frac{1 - \cos(K)}{2} + z_o \quad (13)$$

$$K = \frac{2\pi t}{T} \quad (14)$$

O gráfico de uma curva cicloidal no espaço 3D pode ser vista na Figura 7a. A mesma trajetória é ilustrada na Figura 7b em função do tempo. É possível perceber que a curva possui primeira derivada nula no momento em que a pata toca o solo, o que é favorável ao controle de malha aberta já que quanto mais suave for a aterrissagem, menos distúrbios são causados no sistema.

Além da altura, distâncias em x e y e período, o planejador de trajetória do Caramelo também conta com dois parâmetros que têm como objetivo melhorar ainda mais o controle da força com que a pata toca o chão. Como os atuadores do robô são servomotores controlados por posição, o torque é proporcional ao deslocamento que este deve realizar entre os pontos da trajetória. Ou seja, quanto maior a resolução da trajetória, mais suave será o movimento. No entanto, a resolução da trajetória N é fixa, dada em função do período do passo e da frequência de controle do planejador de marchas (50 Hz) $N = 50T$. Logo, a estratégia adotada é a de espaçar a mesma quantidade de pontos de forma desigual ao longo

Figura 7: Trajetórias cicloidais para o passo de robô.



Fonte: autores.

do período do passo, de forma que haja mais pontos próximos ao momento em que a pata aterrissa no solo e menos pontos próximos ao momento em que ele é erguido. O parâmetro P_T é uma fração do período total do passo e o parâmetro P_N a fração do número total de pontos do passo que deve se encontrar entre o tempo 0 e $P_T \cdot T$. Em outras palavras, se $P_T = 0.66$ e $P_N = 0.33$, 33% de N estará nos primeiros dois terços do período, enquanto os 67% restantes estarão no um terço final. A trajetória considerando esses parâmetros está ilustrada na Figura 7c.

5 RESULTADOS E DISCUSSÃO

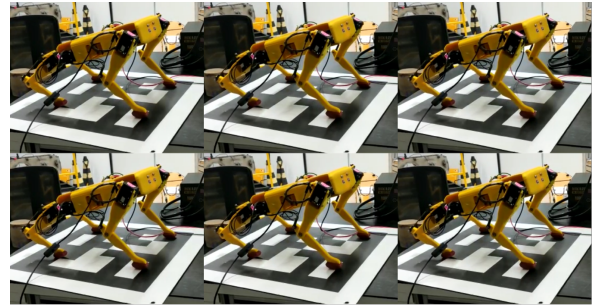
A seguir, serão apresentados os resultados preliminares e os experimentos realizados com o Caramelo. Este refere-se aos testes mencionados na seção 3.3 enquanto aquele refere-se aos resultados obtidos com as funcionalidades básicas do robô: capacidade de movimentação a partir do modelo cinemático e capacidade de controle da orientação do corpo com o controlador de angulação.

5.1 Resultados preliminares

Os primeiros testes realizados com o protótipo foram relacionados à movimentação do corpo a partir do modelo cinemático. Foi observado que o sistema não só é capaz de realizar a cinemática de cada uma das pernas individualmente como também do seu corpo em

todos os 6 graus de liberdade (translações em x , y e z e rotações em *roll*, *pitch* e *yaw*). A figura 8 ilustra o movimento do corpo do robô em cada um desses graus de liberdade.

Figura 8: Movimentação do corpo do robô.



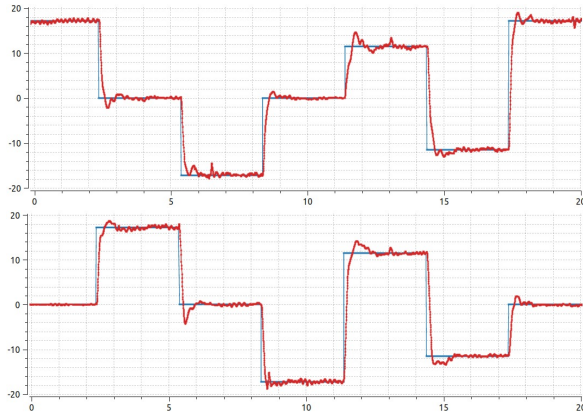
Fonte: autores.

Do mesmo modo, os gráficos da figura 9 ilustram o comportamento do sistema à variação dos *setpoints* de orientação em *roll* (gráfico superior) e em *pitch* (gráfico inferior) para o corpo do robô, a partir da aplicação de diversos degraus (curvas em azul). Nota-se que o protótipo é capaz de se adaptar rapidamente aos novos valores desejados de orientação simultaneamente em ambos os eixos.

5.2 Experimentos

Em seguida, serão apresentados os resultados dos experimentos descritos na seção 3.3.

Figura 9: Respostas dos controles de angulação



Fonte: autores

5.2.1 Análise de trajetória

Durante este experimento, o objetivo foi analisar o tempo real de execução da trajetória, as coordenadas finais (x_f, y_f) da pata e a altura máxima z_{max} atingida durante o passo, comparando-os com os valores esperados para cada um desses parâmetros. A tabela 1 mostra as médias encontradas para cada um desses parâmetros durante a realização dos testes.

Tabela 1: Resultados do experimento da trajetória da pata.

	1	2
\bar{tempo} (s)	0.569965	0.576984
σ_{tempo}	0.058441	0.077332
\bar{x}_{final} (m)	0.050344	0.030237
$\sigma_{x_{final}}$ (m)	0.000104	0.000134
\bar{y}_{final} (m)	0.029242	0.049143
$\sigma_{y_{final}}$ (m)	0.000128	0.000095
\bar{z}_{max} (m)	0.047568	0.046162
$\sigma_{z_{max}}$	0.000278	0.005562

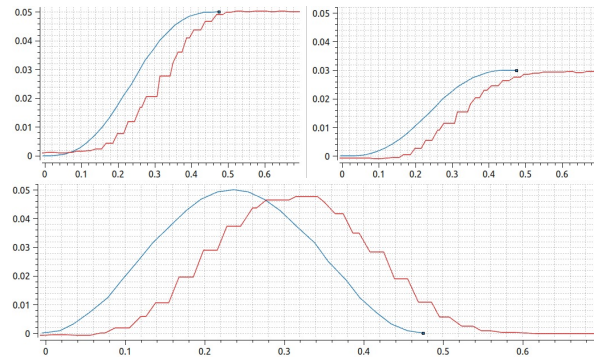
Fonte: autores.

Inicialmente, foram realizados dois testes de t-Student para amostras independentes, o primeiro relacionando o tempo de execução da trajetória nos experimentos 1 e 2, e o segundo relacionando a altura máxima alcançada, com o intuito de verificar se os resultados se alteram para diferentes valores (x, y) desejados. O resultado da análise de

t-Student traz um p_{valor} de aproximadamente 0.6931, em se tratando de tempo, e de 0.1722 para a altura máxima alcançada durante o passo. Ambos os valores encontrados são superiores a 0.05, o que mostra que, para uma confiabilidade de 95%, não há uma diferença significativa entre as médias em cada experimento, o que significa que diferentes comandos de trajetórias em (x, y) não interferem nos resultados de tempo e altura máxima do passo.

Os gráficos da figura 10 representam um dos testes realizados para o primeiro caso $(x = 0.05m, y = 0.03m)$, sendo os gráficos superiores esquerdo e direito correspondentes à trajetória em x e y respectivamente, e o gráfico inferior correspondente à trajetória em z (altura do passo). Nota-se que, como esperado pela análise da tabela 1, há um atraso na execução da trajetória (neste caso específico de aproximadamente $54ms$) e a altura máxima alcançada é levemente inferior à desejada (alcançando neste caso um valor próximo a $0.0476m$).

Figura 10: Trajetórias realizadas pelas patas.



Fonte: autores.

5.2.2 Controle de velocidade

Durante os testes de controle de velocidade foi observada uma caminhada eficiente em ambos os terrenos, porém, mais linear e com menor trepidação em terrenos regulares do que em terrenos irregulares. Por meio dos dados coletados, foram extraídas as informações da velocidade média da caminhada para cada teste, oscilação máxima de

rotação do corpo em *roll* e *pitch* e os respectivos desvios padrões, apresentados na Tabela 2. Assim, foi observado que o robô se aproximou mais vezes da velocidade solicitada no terreno regular combinado ao controle de rotação, enquanto que no irregular, o teste com controle de rotação ativo apresentou uma velocidade média inferior e baixa consistência nos dados. A diferença percebida entre a velocidade solicitada e a inferida é justificada pelo acúmulo de erro nas coordenadas finais (x_f, y_f) das patas a cada passo. Ou seja, quando levado em consideração o próprio peso do robô, o sistema de controle não é capaz de seguir fielmente a trajetória do passo.

Tabela 2: Resultados do experimento do controle de velocidade.

	1	2	3	4
Ter.	Reg.	Reg.	Irreg.	Irreg.
C. R.	Não	Sim	Não	Sim
Vel. (cm/s)	2.92	3.33	2.97	2.77
σ_{Vel} (cm/s)	0.135	0.079	0.197	0.241
Δ_{Roll}	12.54°	8.74°	13.90°	12.17°
σ_{Roll}	2.42°	2.14°	1.80°	3.27°
Δ_{Pitch}	11.46°	8.29°	15.19°	13.18°
σ_{Pitch}	1.83°	2.39°	1.44°	2.94°

Fonte: autores.

Ainda na Tabela 2, foi observado que a menor oscilação em *roll* e em *pitch* ocorreu no terreno regular utilizando o controle de rotação, enquanto que a maior oscilação ocorreu nas condições opostas (teste 3). A fim de avaliar se o controle de rotação implementado contribuiu na estabilidade da caminhada, foi utilizado o teste t-Student para amostras independentes, considerando um nível de confiança de 95%. O teste de t-Student foi feito comparando os testes 1 e 2 (mostrados na tabela), e os testes 3 e 4, ou seja. O resultado desta análise mostra que para terrenos planos, a hipótese nula de que o controle de rotação não influencia na estabilidade do robô pode ser rejeitada, pois o p-valor se mostrou

menor que 0,05. Ou seja, neste caso, o controle de rotação ajudou a estabilizar o corpo do robô. Entretanto, para terrenos irregulares, o uso do controlador não ocasionou mudanças significativas na estabilidade do robô, dado o p-valor de 0,0778. Logo, para terrenos irregulares, o controlador de rotação não se mostrou eficiente para estabilizar o corpo do robô.

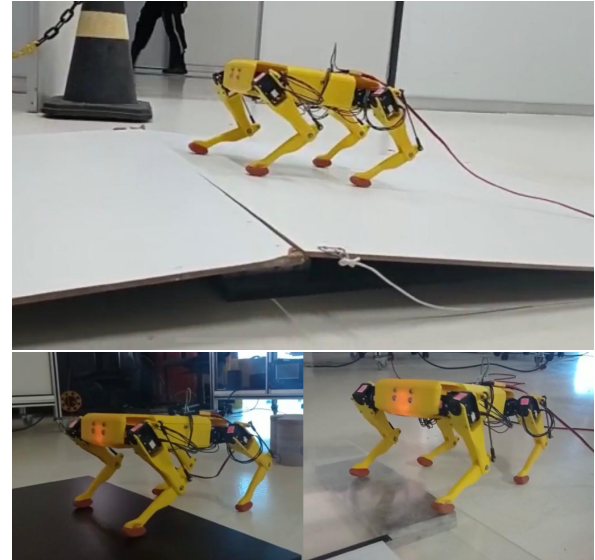
5.2.3 Plano inclinado

Durante este experimento, constatou-se que o robô possui a habilidade de andar por terrenos com 5,336° de inclinação, figura 11.

5.2.4 Superação de obstáculos

O último experimento constatou que o robô é capaz de ultrapassar pequenos obstáculos. Para isso, foram considerados degraus de três diferentes alturas: 2, 4 e 5 cm (ver figura 11). O robô foi capaz de superar degraus de 2 e 4 cm, porém falhou ao tentar subir um de 5 cm.

Figura 11: Fotos do robô no teste da rampa e do degrau.



Fonte: autores

6 CONCLUSÃO

Este trabalho abordou os conceitos utilizados para locomoção de robôs quadrúpedes

e buscou aplicá-los no desenvolvimento de um robô real. O Caramelo foi desenvolvido para fins de educação e pesquisa na área de robôs com pernas, mais especificamente quadrúpedes. Trata-se de um projeto *open source*, cujo código fonte está disponível publicamente no *GitHub* [25].

Seu requisito principal era conseguir se locomover no espaço tridimensional de forma teleoperada e, a partir dos resultados apresentados, pode-se concluir que ele atendeu esse requisito. Os testes realizados com o protótipo buscaram avaliar a performance da locomoção do robô. Com base do primeiro teste, pôde-se concluir que o modelo cinemático é capaz calcular as posições das juntas necessárias para que a pata siga a trajetória e que o sistema de controle é capaz de conduzi-la através da trajetória, independente da distância requisitada em x e y , no cenário sem carga.

O teste do controle de velocidade mostrou que o robô não alcançou a velocidade de $0,05m/s$ em nenhum dos cenários, estando o cenário com chão plano e controle de rotação ativo o mais próximo deste valor. Essa diferença pode estar relacionada com o fato de que o sistema de controle não conseguiu operar de forma adequada ao precisar sustentar o peso do robô. Como o controle da trajetória é feito em malha aberta (apenas os controladores individuais das juntas possuem malha fechada), não há compensação e o passo acaba sendo menor do que o esperado. Ademais, também pôde-se observar que o controle de rotação resultou numa maior estabilidade do robô quando operando em terrenos planos, porém o mesmo não vale para o cenário com terreno irregular. Neste caso, o uso do controle de rotação não ocasionou em uma melhora tão significativa na estabilidade.

Também é possível concluir que o Caramelo é capaz de superar planos inclinados e degraus de até $4cm$ de altura.

Por fim, pode-se dizer que o robô cumpriu quatro dos seis requisitos definidos anteriormente. Embora ele não tenha conseguido

superar obstáculos com $5cm$ de altura, ele conseguiu com $4cm$, o que demonstra um resultado promissor. Quanto a sua capacidade de *payload*, a partir dos testes realizados e do que pôde ser observado pela equipe ao operar o robô, ele não é capaz de carregar uma carga de 2 kg. A massa total do robô é de aproximadamente 2 kg. Portanto, seriam necessários atuadores muito mais robustos para que ele pudesse sustentar 100% do seu peso. Logo, conclui-se que esse requisito foi superdimensionado, considerando a escala do projeto proposto nesse trabalho.

Para trabalhos futuros, recomenda-se revisar os atuadores usados no robô. Ao longo da fase de testes, foi observado que eles frequentemente falhavam por conta de sobrecarga, o que é um sinal que eles foram subdimensionados. Para corrigir essa questão, é possível substituir o modelo atual por um com maior torque e/ou investigar a configuração das pernas do robô. A equipe observou que, como os motores que mais falhavam eram os traseiros, adotar a configuração *elbow-knee* pode diminuir a carga neles, já que deixaria as pernas traseira em uma configuração simétrica às frontais. Espera-se, também, que isso influencie na performance de locomoção, deixando o robô mais rápido e, talvez, com a capacidade de superar obstáculos maiores do que $4cm$.

Referências

- ¹ ZHANG, X. et al. The quadruped robot locomotion based on force control. **Proceedings of the 2015 27th Chinese Control and Decision Conference, CCDC 2015**, Institute of Electrical and Electronics Engineers Inc., p. 5440–5445, 2015.
- ² SHI, Y. et al. Structural design, simulation and experiment of quadruped robot. **Applied Sciences (Switzerland)**, MDPI, v. 11, 2021.
- ³ RAIBERT, M. et al. BigDog, the Rough-Terrain Quadruped Robot. **IFAC Proceedings Volumes**, v. 41, n. 2, p. 10822–10825, 2008.

- ⁴ DYNAMICS, B. **Spot**. Disponível em: <http://www.bostondynamics.com/products/spot>. Acesso: 24 de nov de 2022.
- ⁵ FANKHAUSER, P.; HUTTER, M. AnyMal: A unique quadruped robot conquering harsh environments. **Research Features**, p. 54–57, 2018.
- ⁶ BISWAL, P.; MOHANTY, P. K. **Development of quadruped walking robots: A review**. [S.l.]: Elsevier, 2021. 2017-2031 p.
- ⁷ YAO, L.; YU, H.; LU, Z. Design and driving model for the quadruped robot: An elucidating draft. **Advances in Mechanical Engineering**, SAGE Publications Inc., v. 13, 2021.
- ⁸ HAZARDEX. **ANYmal – A unique quadruped robot for harsh environments**. Disponível em: <https://www.hazardexonthenet.net/article/167536/ANYmal-A-unique-quadruped-robot-for-harsh-environments.aspx>. Acesso: 25 de nov de 2022.
- ⁹ EXPO, R. S. . **TEST DRIVE SPOT**. Disponível em: <https://www.roboticssummit.com/test-drive-spot/>. Acesso: 25 de nov de 2022.
- ¹⁰ KITANO, S. et al. Titan-xiii: sprawling-type quadruped robot with ability of fast and energy-efficient walking. **ROBOMECH Journal**, Springer International Publishing, v. 3, p. 1–16, 2016.
- ¹¹ ZHONG, Y. et al. **Analysis and research of quadruped robot's legs: A comprehensive review**. [S.l.]: SAGE Publications Inc., 2019.
- ¹² SONG, S.-M.; WALDRON, K. J. **Machines That Walk: Adaptive Suspension Vehicle**. United States: [s.n.], 1989. 314 p.
- ¹³ ZHANG, S. et al. A free gait generation method for quadruped robots over rough terrains containing forbidden areas. **Journal of Mechanical Science and Technology**, Korean Society of Mechanical Engineers, v. 29, p. 3983–3993, 2015.
- ¹⁴ SANTOS, P. G. D.; GARCIA, E.; ESTREMER, J. **Quadrupedal locomotion: an introduction to the control of four-legged robots**. [S.l.]: Springer, 2006. v. 1.
- ¹⁵ RAIBERT, M. **Legged robots that balance**. [S.l.]: MIT Press, 1986. 235 p. ISBN 9780262181174.
- ¹⁶ JUMANTORO, E.; ALASIRY, A. H.; HERMAWAN, H. Stability optimization on quadruped robot using trajectory algorithm. **Proceedings IES-ETA 2017 - International Electronics Symposium on Engineering Technology and Applications**, Institute of Electrical and Electronics Engineers Inc., v. 2017-Decem, p. 93–98, 2017.
- ¹⁷ STANFORD Quadruped. Disponível em: <https://github.com/stanfordroboticsclub/StanfordQuadruped>. Acesso: 24 de nov de 2022.
- ¹⁸ DIY hobby servos quadruped robot. Disponível em: <https://hackaday.io/project/171456-diy-hobby-servos-quadruped-robot>. Acesso: 24 de nov de 2022.
- ¹⁹ LIU, D. et al. Stable walking control of parallel wheel-foot robot based on zmp theory. **Chinese Control Conference, CCC**, IEEE Computer Society, p. 6620–6625, 2017.
- ²⁰ NOTSPOT robot simulation. Disponível em: https://github.com/lnotspot/lnotspot_sim_cpp. Acesso: 24 de nov de 2022.
- ²¹ CHEN, G. et al. Virtual model control for quadruped robots. **IEEE Access**, Institute of Electrical and Electronics Engineers Inc., v. 8, p. 140736–140751, 2020.
- ²² ZHANG, G. et al. Torso motion control and toe trajectory generation of a trotting quadruped robot based on virtual model control. **Advanced Robotics**, Robotics Society of Japan, v. 30, p. 284–297, 2016.
- ²³ REIS, M.; VALE, A. **Método BiLi - Uma otimização para revisão bibliográfica e literária**. Disponível em: <https://github.com>

/Brazilian-Institute-of-Robotics/bir-mini-bili-method>. Acesso: 28 de nov de 2022.

²⁴ MACENSKI, S. et al. Robot operating system 2: Design, architecture, and uses in the wild. **Science Robotics**, v. 7, n. 66, p. eabm6074, 2022. Disponível em: <<https://www.science.org/doi/abs/10.1126/scirobotics.abm6074>>.

²⁵ CAMEL. Disponível em: <<https://github.com/Brazilian-Institute-of-Robotics/bir-black-mouth>>. Acesso: 28 de nov de 2022.