



**Paulo Teixeira Vale de Carvalho**

**Projeto, Modelagem e Controle de um Robô  
Quadrúpede Modular**

**Projeto de Graduação**

Projeto de Graduação apresentado ao Departamento de Engenharia Mecânica da PUC-Rio.

Orientador : Marco Antônio Meggiolaro - PUC-Rio  
Coorientadora: Vivian Suzano Medeiros - USP

Rio de Janeiro  
Novembro de 2022

## Agradecimentos

Meus primeiros agradecimentos à Bárbara Beato Ribeiro, minha namorada e aquela que acompanhou mais de perto minha longa trajetória durante a graduação do início ao fim, me dando apoio e paz quando precisava.

Agradeço à minha família, especialmente aos meus pais que me criaram e fizeram sacrifícios por mim. Minha mãe, Cristina Vieira Teixeira, que sempre cuidou de mim e me deu forças, e meu pai, Eduardo Vale de Carvalho, que sempre me orientou e mostrou como um engenheiro deve ser.

Agradeço à equipe Riobotz, ao LabRob e membros de ambas, pela possibilidade de contato com a robótica, seja por meio de competição ou por estudos, e também pelas amizades, conquistas e experiências que consegui. Entre eles, agradeço especialmente Gabriel Silva e João Vitor Ziliani, que foram meu capitão e coordenador de mecânica, Gabriel Fischer Abati, um grande amigo e um mentor, Vivian Suzano Medeiros, minha co-orientadora tutora em robôs com pernas, e ao professor Marco Antônio Meggiolaro, que além de ser meu Orientador, coordena Riobotz e o LabRob, sendo um grande professor e por depositar confiança em mim.

Agradeço a PUC-Rio, pela sua infraestrutura e pelos seus profissionais. Espero ter mais coisas a viver nessa faculdade que sempre foi meu desejo desde ensino médio e é uma alegria ser um graduado desse lugar maravilhoso com professores incríveis.

E por último, mas não menos importante, agradeço à todos meus amigos, sejam da faculdade ou fora, sejam colegas de sala ou de trabalho, pois não só ajudaram a me moldar academicamente e profissionalmente, mas também ajudaram a me moldar como um adulto e cidadão.



## Resumo

### Projeto, Modelagem e Controle de um Robô Quadrúpede Modular

No contexto de grande avanço dos estudos de robôs com pernas para aplicações em inspeção industrial, este trabalho visa adaptar e construir o robô quadrúpede *Solo*, um robô leve e compacto desenvolvido por pesquisadores da Universidade de Nova Iorque (NYU), cujo projeto mecânico e eletrônico foi disponibilizado de forma aberta e gratuita (*open-source*). O robô possui 8 graus de liberdade, controle de torque, velocidade e posição em todas as juntas, e uma estrutura mecânica projetada para fabricação com impressora 3D, o que reduz bastante o seu custo de fabricação e montagem. O objetivo é aumentar ainda mais a acessibilidade deste projeto por meio de melhorias e modificações, para que permita o uso de componentes nacionais e, com isso, possa servir como plataforma de pesquisa na área de robôs com pernas em universidades brasileiras. O trabalho apresenta a modelagem cinemática da uma perna do robô e simulações realizadas por meio de ferramentas como o Matlab e RaiSim. Como forma de validação, experimentos foram realizados com o controle de diferentes trajetórias em uma perna do *Solo*, com a aquisição dos dados de seus atuadores para análise do seu funcionamento.

### Palavras-chave

Robô Quadrúpede; Impressora 3D; *Open-Source*;

## Abstract

### Project, Model and Control of a Modular Quadruped Robot

In light of the great advances in the field of legged robotics for applications in industrial inspection, this work aims to adapt and build the quadruped robot *Solo*, a light and compact robot developed by researchers at the University of New York (NYU), whose mechanical and electronic design was made available openly and free of charge (*open-source*). The robot has 8 degrees of freedom, torque, speed and position control on all joints, and a mechanical structure designed to be manufacturing with a 3D printer, which reduces considerably its assembly cost. The objective is to further increase the accessibility of this project through improvements and modifications, so it allows the use of national components and, with that, it can serve as a research platform in the field of legged robots for Brazilian universities. This work presents the kinematic modeling of the robot's leg and dynamic simulations performed using available tools such as Matlab and RaiSim. Experimental validation is carried out with the control of different trajectories in one leg of the robot, with the acquisition of data from its actuators for a quantitative analysis.

### Keywords

Quadruped robot; 3D printing; Open-Source

# Sumário

1	Introdução	10
1.1	Revisão Bibliográfica	12
1.2	Objetivos	15
2	Projeto Mecânico	17
2.1	Descrição	17
2.2	Estrutura Mecânica	17
2.2.1	Alterações no Módulo de Atuação das Pernas	18
2.2.2	Alterações na Base do Robô	22
2.2.3	Motores	23
2.2.4	Sensores	23
2.3	Simulação Estática	24
2.3.1	Parâmetros	25
2.3.2	Resultados	26
3	Eletrônica	28
3.1	Eletrônica Original	28
3.1.1	Alterações	29
3.2	Controlador <i>ODrive</i>	29
3.2.1	Configuração	31
4	Fabricação e montagem	34
4.1	Motor	34
4.2	Corpo	36
4.3	Estimativa de Custo	38
5	Modelagem	41
5.1	Cinemática	41
5.2	Dinâmica	42
5.3	Parâmetros	43
6	Simulação	44
6.1	<i>Machines in Motion Laboratory</i>	44
6.1.1	<i>Kino-Dynamic Trajectory Optimization</i>	44
6.1.2	<i>Stable Walking</i>	44
6.1.3	Resultado	46
7	Testes Experimentais	49
7.1	Módulo de atuação	49
7.2	Testes de uma perna com o <i>Odrive</i>	51
7.2.1	Matlab	51
7.2.2	Comandos básicos	52
7.2.3	Trajectoria Simples	53
7.2.4	Trajectoria vertical no espaço cartesiano	57
7.2.5	Trajectoria simulando um passo no espaço cartesiano	61



## Lista de figuras

1.1	Pernas versus Rodas [1].	10
1.2	BigDog [2]	10
1.3	Robôs quadrúpedes.	11
1.4	ANYmal Modelo C	11
1.5	Robôs quadrúpedes comerciais de baixo custo.	12
1.6	Princípios de sistema de navegação de um robô móvel autônomo [1].	13
1.7	O robô Squro.	14
1.8	Robôs quadrúpedes hidráulicos.	14
1.9	Robô <i>open-source</i> Solo.	15
2.1	Projeto original do robô Solo.	17
2.2	Módulo de atuação com porcas.	18
2.3	Área de encaixe do Motor	18
2.4	Área Central do módulo de atuação.	19
2.5	Tampa nova do módulo.	19
2.6	Polias finais.	20
2.7	Pinhão	20
2.8	Comparação das polias intermediárias.	21
2.9	Comparação com vista de corte.	21
2.10	Módulo de atuação aberto.	21
2.11	Comparação das bases do robô.	22
2.12	Furos tangentes a estrutura da base.	22
2.13	Lateral Tripartida	23
2.14	Motor Antigravity MN4004 KV400.	24
2.15	Encoder E2.	24
2.16	Configuração da simulação.	25
2.17	Resultado da simulação estática.	26
2.18	Resultados da simulação para a extremidade da perna.	27
2.19	figure.caption.32	
3.2	Placa MicroDriver atualizada.	30
3.3	Placa Masterboard atualizada.	30
3.4	Controlador <i>Odrive</i> .	31
3.5	Estrutura experimental para testes de controle de torque dos motores.	33
4.1	Projeto final.	34
4.2	Retirada do colar, anel e moscas.	35
4.3	Retirada do eixo antigo.	35
4.4	Montagem do Motor com novo eixo.	36
4.5	Montagem do módulo de atuação.	36
4.6	Montagem de uma perna	37
4.7	Corpo montado.	37
4.8	Robô apoiado em caixa.	38
5.1	Representação do modelo dinâmico de um robô com pernas.	41

5.2	Configuração cinemática da perna do robô.	42
5.3	Termos Dinâmico	42
6.1	Ilustração de uma trajetória estaticamente estável com um robô quadrúpede.	45
6.2	Movimento <i>Stable Walking</i> .	47
6.3	Posições das patas do robô ao longo da trajetória.	47
6.4	Deslocamento.	48
7.1	Bancada para teste do módulo de atuação da perna do robô Solo.	49
7.2	Primeiro teste com o módulo de atuação.	50
7.3	Segundo teste com o módulo de atuação.	50
7.4	Modelo do robô Solo no Matlab.	51
7.5	Bancada para teste de comandos.	52
7.6	Bancada para teste de trajetórias.	53
7.7	Posição inicial na horizontal.	53
7.8	Teste de visualização da trajetória simples no Matlab.	54
7.9	Fotos do Vídeo da trajetória simples.	55
7.10	Variação de ângulos na trajetória simples.	55
7.11	Gráfico do erro na trajetória das juntas.	56
7.12	Movimentação da pata durante a trajetória simples.	56
7.13	Gráfico ampliado de trajetória simples.	57
7.14	Calibração e visualização da trajetória vertical.	58
7.15	Fotos do Vídeo da trajetória vertical.	58
7.16	Variação de ângulos na trajetória vertical.	59
7.17	Gráfico do erro na trajetória vertical.	59
7.18	Gráfico da trajetória vertical.	60
7.19	Gráfico ampliado da trajetória vertical.	60
7.20	Trajetoória simulando um passo para o a base do robô parada.	61
7.21	Fotos do Vídeo da trajetória de passo.	62
7.22	Variação de ângulos na trajetória de passo	62
7.23	Gráfico do erro na trajetória de passo.	63
7.24	Gráfico da trajetória do passo executada pela perna do robô.	63

## Lista de tabelas

2.1	Informações das Correias	20
2.2	Informações do motor.	23
2.3	Informações do <i>encoder</i> .	24
2.4	Comparativo da tensão de escoamento de materiais comumente utilizados em impressão 3D.	27
3.1	Configuração.	32
3.2	Ganhos do controlador PID.	33
4.1	Custo de compras	39
4.2	Custos de impressão 3D.	39
4.3	Custo da eletrônica original.	40
4.4	Custo de materiais mecânicos.	40
4.5	Custo total estimado do projeto.	40
5.1	Tabela de Parâmetros	43
6.1	Stable walking gait.	45
6.2	Perna traseira esquerda.	45
6.3	Pata direita frontal.	46
6.4	Perna direita traseira.	46
6.5	Pata esquerda dianteira.	46

# 1

## Introdução

Robôs móveis com pernas possibilitam uma locomoção segura em terrenos acidentados e são capazes de ultrapassar obstáculos, como aclives e escadas, uma vantagem considerável em comparação a robôs com rodas (Figura 1.1). Dentro da categoria de robôs com pernas, os robôs quadrúpedes se destacam por serem mais estáveis, já que possuem quatro pontos de apoio, e por possuírem uma interface considerada amigável para seres humanos, uma vez que sua estrutura é inspirada na biologia de animais terrestres de estimação, como cães e cavalos.

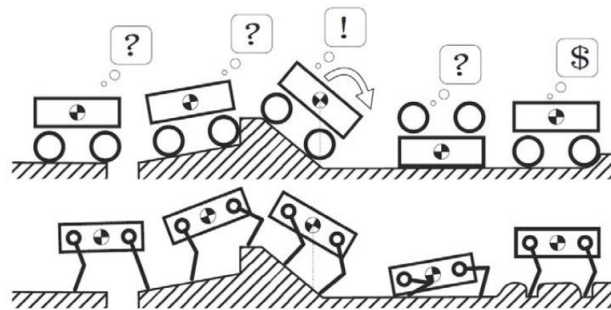


Figura 1.1: Pernas versus Rodas [1].

Um marco importante para área é o robô quadrúpede *BigDog* [2], apresentado na Figura 1.2, desenvolvido pela empresa americana *Boston Dynamics* em 2005. Seu propósito era a locomoção e a movimentação de cargas pesadas de uso militar em terrenos acidentados. Ele era movido à combustão interna e suas juntas eram acionadas hidraulicamente.



Figura 1.2: BigDog [2]

A partir dele, novos modelos foram criados, pensando na utilização além do campo militar, mas também no ramo industrial para atividades tanto de inspeção quanto de transporte de cargas. Outras possibilidades de aplicação



são a exploração de áreas acidentadas, tarefas de busca e resgate e até mesmo de interação com humanos. Desde então, diversos outros robôs quadrúpedes foram desenvolvidos, entre eles o *MIT Cheetah* (Figura 1.3(a)), desenvolvido pelo *MIT*, um robô notável pela sua alta velocidade, tendo atingido velocidades de até  $3.7 \text{ m/s}^2$  [3]; e o *Spot* (Figura 1.3(b)), criado e comercializado pela empresa *Boston Dynamics*, que ganhou bastante destaque na mídia por sua alta mobilidade e por seus vídeos realizando movimentos dinâmicos e de dança<sup>1</sup>.

1.3(a): *MIT Cheetah*.1.3(b): *Spot*, da *Boston Dynamics*.

Figura 1.3: Robôs quadrúpedes.

Outra empresa de destaque é a *ANYbotics*, originária da universidade suíça *ETH Zurich*, cujo robô principal é o *ANYmal* [4], que já possui uma versão comercial, o modelo C (Figura 1.4). A empresa, em parceria com a universidade, segue fazendo projetos de melhorias e recentemente anunciaram um projeto de uma versão, o *ANYmal X*, mais robusto e com certificação à prova de água, poeira e explosão, ideal para aplicação em cenários industriais de alto risco, como plataformas petrolíferas [5].

Figura 1.4: *ANYmal* Modelo C

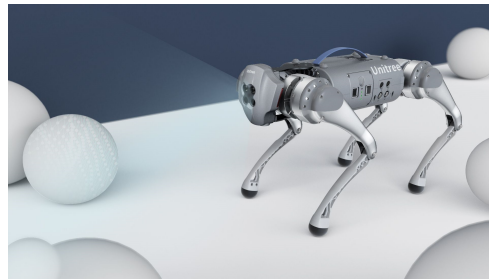
---

<sup>1</sup>Veja o vídeo em: <https://youtu.be/fn3KWM1kuAw>

Apesar dos grandes avanços, ainda existem limitações para uma maior aplicação deste tipo de robô no mercado, principalmente devido ao seu custo elevado. No entanto, empresas chinesas, como a *Xiaomi* com o robô *Cyber-dog* [6] (Figura 1.5(a)), e a *Unitree* com robô *Go1*<sup>2</sup> (Figura 1.5(b)), buscam ser alternativas comerciais mais baratas em relação as outras já citadas. Outra alternativa para o acesso a robôs quadrúpedes são iniciativas *open-source*, que disponibilizam o projeto mecânico e eletrônico de robôs quadrúpedes compactos e de baixo custo para incentivar o acesso e a pesquisa com esse tipo de plataforma.



1.5(a): *Cyber Dog*, da empresa *Xiaomi*.



1.5(b): *Go1*, da empresa *Unitree*.

Figura 1.5: Robôs quadrúpedes comerciais de baixo custo.

Nesse contexto, ficam claras as vantagens do uso de robôs quadrúpedes em cenários desafiadores e, para evoluir ainda mais as capacidades desse tipo de robôs, se faz o necessário o desenvolvimento de um plataformas acessíveis para o meio acadêmico, de baixo custo e fácil montagem, que viabiliza o desenvolvimento e a pesquisa com robôs quadrúpedes.

## 1.1 Revisão Bibliográfica

Robôs móveis são robôs cuja base não é fixa e sua posição varia por meio de interações com o meio externo [7]. Para que robôs móveis possam se locomover de forma autônoma, eles comumente utilizam um sistema de navegação que utiliza informações de sensores proprioceptivos e exteroceptivos<sup>3</sup> para analisar o ambiente, se localizar, planejar uma rota e a execução da rota acontece por meio de seus atuadores através de um sistema de controle de movimento [1]. Esse ciclo está representado na Figura 1.6. Uma categoria de robôs móveis que tem sido bastante estudada nos últimos 20 anos é a de robôs com pernas,

<sup>2</sup>Link disponível em: <https://m.unitree.com/products/go1>

<sup>3</sup>Os sensores proprioceptivos medem valores internos ao robô, tais como posições das juntas, velocidade do motor, voltagem da bateria e orientação da base (via sensores inerciais). Sensores exteroceptivos adquirem informações do ambiente em que o robô se encontra, como medidas de distância, intensidade da luz e amplitude do som [1].

especialmente robôs quadrúpedes. Trabalhos relacionados à robôs quadrúpedes podem envolver qualquer um dos conceitos de navegação apresentados na Figura 1.6, mas comumente são focados na área de planejamento trajetória e locomoção.

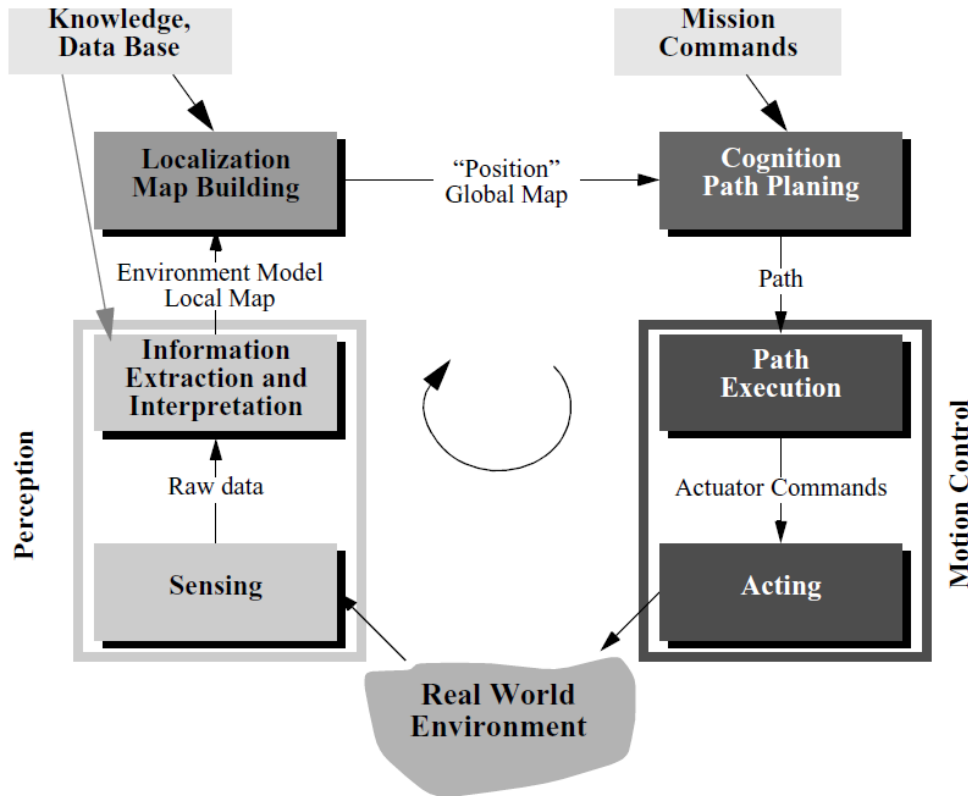


Figura 1.6: Princípios de sistema de navegação de um robô móvel autônomo [1].

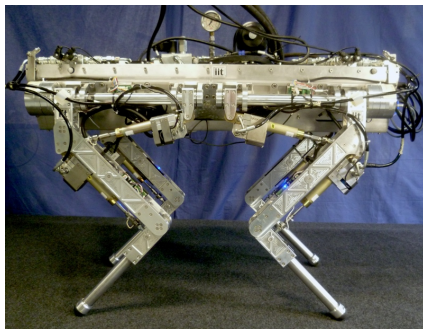
Existem diferentes tipos de estrutura mecânica no que diz respeito à robôs com pernas. No geral, o projeto busca obter uma aproximação do comportamento dinâmico de alguns animais. Exemplos são trabalhos como do robô *Squiro* [8] (Figura 1.7), que visa reproduzir a versatilidade do movimento de ratos para desvio de obstáculos com torso atuado, e um trabalho buscando inspiração nas curvaturas feitas por um guepardo enquanto corre [9].

A estrutura mais comum de robôs quadrúpedes consiste em pernas atuadas com 3 graus de liberdades presas a um torso mais pesado, onde se encontra toda a eletrônica de atuação, controle e navegação do sistema. Alguns desses robôs já foram citados anteriormente, como o *ANYmal* (Figura 1.4), o *Spot* (Figura 1.3(b)) e o *Go1* (Figura 1.5(b)). Todos esses robôs têm em comum o uso de atuadores elétricos. Outra possibilidade para esses robôs é o uso de atuadores hidráulicos, como é o caso do robô *HyQ* (Figura 1.8(a)), do Instituto Italiano de Tecnologia (IIT), em Gênova, na Itália. A desvantagem é que esse tipo de atuador requer uma unidade de pressurização de óleo embarcada, o que



Figura 1.7: O robô Squiro.

deixa o robô pesado e grande, mas em compensação, ele é capaz de realizar forças e transportar cargas muito maiores. Esse mesmo robô conseguiu mover um avião de 3300kg<sup>4</sup>, como mostra a Figura 1.8(b).



1.8(a): O robô HyQ.



1.8(b): O robô HyQ movendo um avião de pequeno porte.

Figura 1.8: Robôs quadrúpedes hidráulicos.

Outra área de estudo em relação a robôs quadrúpedes é a combinação das pernas com rodas, criando uma locomoção híbrida que combina das características dinâmicas de ambos os atuadores. No entanto, para isso, é necessário a realização de um sistema de controle que permita tirar proveito das rodas e das pernas para locomoção. Um exemplo é a versão com rodas do robô quadrúpede *ANYmal*, apresentada em [10][11], que focam na adaptação da trajetória do robô ao longo de terrenos não estruturados, usando movimentos simultâneos com rodas e pernas, se movendo com mais velocidade em cenários com obstáculos.

Uma das principais motivações para o uso de robôs quadrúpedes é a sua utilização em tarefas de inspeção de áreas industriais de risco, já que a presença de pernas permite que esses robôs se desloquem ao longo da planta industrial. Já foram realizados diversos estudos de aplicação real em cenários industriais envolvendo o robô *ANYmal*. Em 2018, o *ANYmal* realizou testes numa plataforma de petróleo *offshore*, conseguindo realizar localização própria, planejamento de trajetória, execução de trajetória com adaptações

---

<sup>4</sup>Vídeo disponível em: <https://youtu.be/pLsNs1ZSTI>

a obstáculos, sensoriamento e leitura de manômetros [12]. Já em 2021, foi também realizado um teste demonstrando a locomoção autônoma do *ANYmal* numa plataforma de geração de energia eólica, num ambiente com risco elétrico [13].

Apesar disso, ainda existem desafios a serem pesquisados para aumentar a robustez de robôs com pernas em cenários reais. Para expandir cada vez mais a área de pesquisa em robôs com pernas, existem alguns projetos *open-source*, que disponibilizam tanto o código quanto o design do robô de forma gratuita, não só para aumentar o acesso a robôs quadrúpedes, mas também para utilização como plataformas de pesquisa. Entre exemplos, podem ser citados o robô PADWQ [14] que possui estudo estrutural de marchas para locomoção (ou *gait*), o robô Charlotte [15] que possui estudos de localização e mapeamento simultâneos, e o Solo [16], que permite a realização de movimentos dinâmico com controle de torque em todas as juntas. Esses três exemplos possuem parte da sua estrutura fabricada a partir de impressão 3D e demonstram ser viáveis para aplicação em pesquisa, no entanto, alguns componentes possuem um custo elevado e são complexos de montar e construir, o que reduzem a sua acessibilidade. O robô Solo, apresentado na Figura 1.9 é utilizado como base para esse trabalho de conclusão de curso.

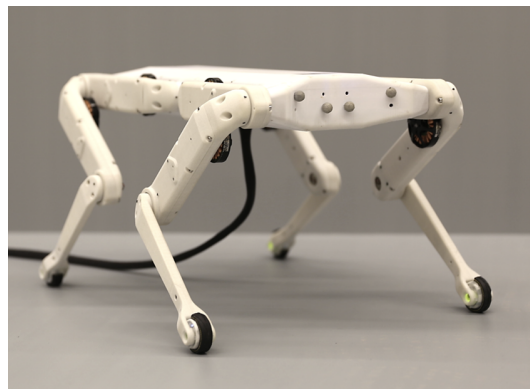


Figura 1.9: Robô *open-source* Solo.

## 1.2 Objetivos

O objetivo deste trabalho de fim de curso é realizar a montagem e o teste de um robô quadrúpede baseado no projeto *open-source* do robô quadrúpede Solo, desenvolvido pela *Open Dynamic Robot Initiative*, e disponibilizado gratuitamente no *GitHub*<sup>5</sup>. A estrutura do robô será fabricada a partir de impressão 3D e modular, facilitando assim a sua montagem e reduzindo seus custos,

---

<sup>5</sup>Link para o material *open-source*: [https://github.com/open-dynamic-robot-initiative/open\\_robot\\_actuator\\_hardware](https://github.com/open-dynamic-robot-initiative/open_robot_actuator_hardware)

o que a torna ideal para aplicação em um ambiente acadêmico, considerando a sua acessibilidade. As partes que não são impressas estão disponíveis para compra no mercado. Como trata-se de um projeto internacional, serão feitos ajustes na parte mecânica e eletrônica para permitir o uso de componentes equivalentes nacionais, reduzindo o custo e o tempo de montagem do sistema. Será desenvolvida a modelagem cinemática e dinâmica de uma das pernas do robô para estudo com o Matlab. Além disso, serão feitas simulações de um controle de trajetória simples, simulado em *PyBullet*, baseado no planejador de trajetória desenvolvido pelos próprios criadores do Solo. Serão realizados também testes de movimentação das patas do robô utilizando um *driver* comercial para os motores *brushless* que compõem as juntas do robô, validando o seu funcionamento.



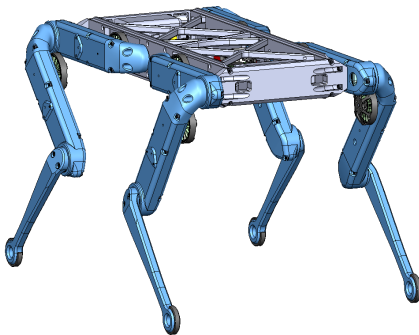
## 2

### Projeto Mecânico

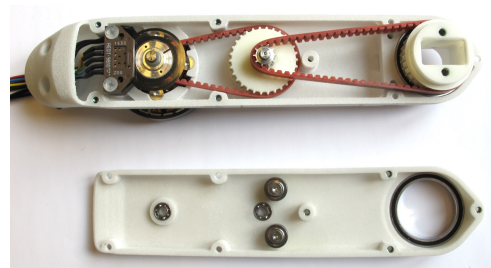
#### 2.1

##### Descrição

O projeto é baseado no robô quadrúpede Solo [16] (Figura 2.1(a)) de 8 graus de liberdade, ou seja, 2 articulações para cada perna, disponível de forma aberta no *GitHub*<sup>1</sup> pelos desenvolvedores da *Open Dynamic Robot Initiative*. As patas do robô são compostas de 3 elos, que são também os módulos de atuação de cada junta (Figura 2.1(b)). Cada módulo de atuação é composto por um motor *brushless* e um sistema de transmissão composto por um conjunto de correias e duas polias com uma redução de 3:1 cada, totalizando 9:1 no final. Sua parte estrutural é feita a partir de impressão 3D para facilitar sua acessibilidade, no entanto, para acomodar o uso de componentes nacionais, se faz necessário um novo projeto usando o programa de *CAD SolidWorks*, mantendo dimensões similares, mas adicionando melhorias e facilitando a sua construção.



2.1(a): Projeto original.



2.1(b): Módulo de atuação original.

Figura 2.1: Projeto original do robô Solo.

#### 2.2

##### Estrutura Mecânica

---

<sup>1</sup>Link para o material open-source: [https://github.com/open-dynamic-robot-initiative/open\\_robot\\_actuator\\_hardware](https://github.com/open-dynamic-robot-initiative/open_robot_actuator_hardware)

### 2.2.1

#### Alterações no Módulo de Atuação das Pernas

A primeira mudança a ser efetuada nos módulos são nos furos, aumentando o material em volta e padronizando todos para tamanho M3 com intuito de facilitar a intercambiabilidade e para suportar mais peso. Além disso, no projeto original, a fixação dos componentes é realizada através de rosqueamento direto na peça, o que não é ideal para material impresso, pois as roscas tendem a se desgastar. Outra alternativa utilizada é o uso de insertos de rosca, no entanto, ainda pode gerar alguma fragilidade já que as paredes disponíveis são finas, mesmo com o incremento de material. Para solucionar o problema, foram utilizadas porcas (Figura 2.2) na outra extremidade dos furos com um encaixe parcialmente hexagonal para não haver rotação da fixação.

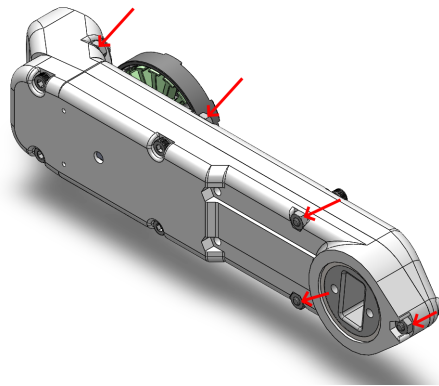


Figura 2.2: Módulo de atuação com porcas.

Na região de encaixe do motor (Figura 2.3), os furos para fixação são completos e com reforços de material para aumento da rigidez. Além disso, um ressalto é feito para apoio do *encoder* e colocação de pinos de latão deste sensor.

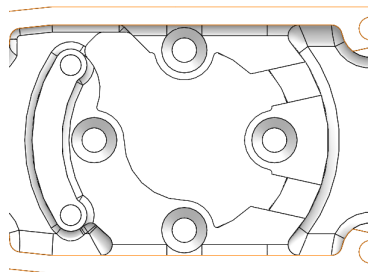


Figura 2.3: Área de encaixe do Motor

Na área central do elo (Figura 2.4), aumenta-se a base para a polia intermediária (região marcada em vermelho) para maior estabilidade e apoio de rolamento axial a ser adicionado. Nessa região, se tem um ressalto na tampa



(seta verde) para impedir que a primeira correia vá para cima e perca o contato com a polia intermediária.

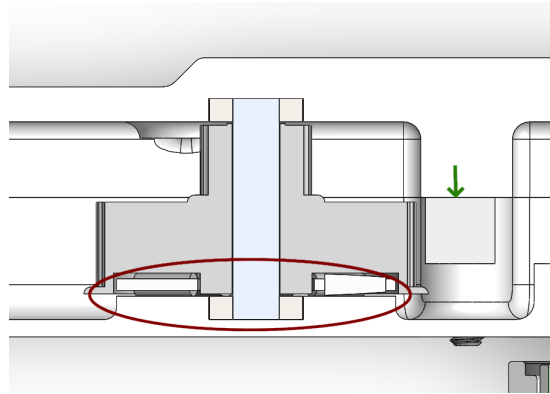


Figura 2.4: Área Central do módulo de atuação.

Na tampa do módulo (Figura 2.5), além dos rasgos para porcas, há incremento de material. Além de maior rigidez, facilita a impressão por dar maior base de apoio para a peça na mesa de impressão. Na área rebaixada, existe um alto relevo com intuito de minimizar eventuais efeitos de flexão da peça que possam ocorrer durante a impressão, enquanto os dois rasgos centrais para as porcas são mantidos para colocar tensionadores de correia, caso necessário.

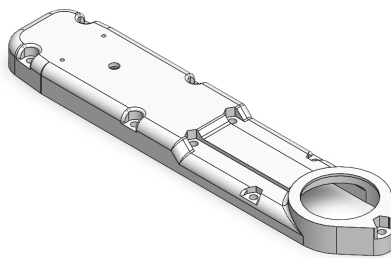


Figura 2.5: Tampa nova do módulo.

Devido à falta de oferta das correias originais, correias dentadas GT2 são utilizadas como substitutas. Estas são normalmente usadas em construção de impressoras 3D e máquinas pequenas, além de possuírem disponibilidade maior no mercado nacional. A escolha foi feita para que as dimensões fossem mantidas o máximo possível das originais (Tabela 2.1), sem ter que fazer uma mudança nas distâncias entre os eixos de transmissão.

Para as polias, busca-se evitar a dependência de usinagem e os dentes foram adaptados para o novo par de correias. A polia final (Figura 2.6) é adaptada para ter o encaixe de porcas no lugar de insertos para a melhor fixação da peça que será acoplada nela.

Tabela Correias				
Estágio	Versão	Tipo	Espessura	Comprimento
Primeiro	Original	<i>Timing Belt</i>	4 mm	150 mm
	Nova	GT2	6 mm	150 mm
Segundo	Original	<i>Timing Belt</i>	6 mm	201 mm
	Nova	GT2	6 mm	200 mm

Tabela 2.1: Informações das Correias

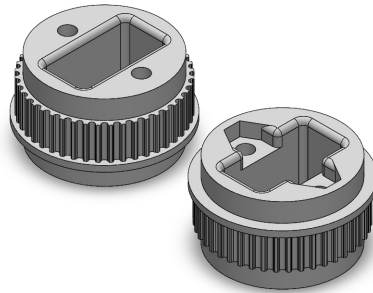


Figura 2.6: Polias finais.

O pinhão (Figura 2.7) é modificado para ter seu furo em forma de "D", com intuito de receber transmissão do eixo do motor, descartando o uso de uma mosca, o que seria prejudicial para os dentes pequenos da engrenagem. Além disso, em vez de alumínio, para evitar custo e tempo usinagem, opta-se do uso de impressão 3D com o material ABS, que tem boa resistência mecânica e é comumente utilizado para essa categoria de componente.

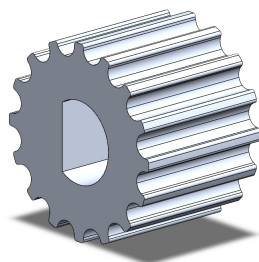


Figura 2.7: Pinhão

Na polia intermediária nova, a versão nova, à esquerda nas Figuras 2.8 e 2.9, recebe um rebaixo maior por baixo para a aplicação de rolamento axial, para melhor estabilidade da peça. Além disso, em vez de uma peça de alumínio dentada passante, se coloca um eixo liso de aço preso aos dois rolamentos de apoio, assim evitando custo e tempo de usinagem e facilitando o encaixe entre

as duas partes.

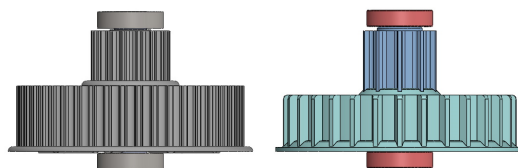


Figura 2.8: Comparação das polias intermediárias.

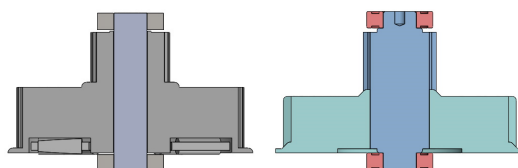
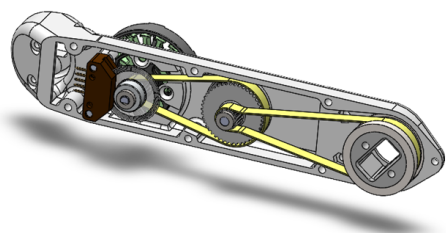


Figura 2.9: Comparação com vista de corte.

Com tais mudanças, a redução nova é de 4:11 por estágio, resultando em uma redução total de 16:121. Fazendo uma subtração com a redução original de 1:9 e dividindo pela mesma, se tem um aumento aproximado de 19,01%. As mudanças implicam um aumento da velocidade final, mas uma perda do torque final.

Por fim, o novo módulo completo com as alterações propostas é apresentado na Figura 2.10.



2.10(a): Desenho do Módulo Aberto.



2.10(b): Foto do Módulo Aberto.

Figura 2.10: Módulo de atuação aberto.

### 2.2.2

#### Alterações na Base do Robô

A nova base do robô, à esquerda na Figura 2.11, tem consideráveis mudanças em relação ao projeto original, começando pela tampa. Originalmente, a tampa consistia em treliças impressas com suportes para os componentes internos presos nas paredes laterais. A desvantagem são, além da necessidade de uma impressora 3D grande para a produção da peça, os parafusos sofrem um esforço de cisalhamento pela forma de fixação, a estrutura é mais frágil e limita as peças que podem ser colocadas internamente. No novo projeto, foram utilizadas duas chapas 3 milímetros de policarbonato fixada por cima e por baixo, assim tendo uma maior base de apoio para os componentes internos e com os esforços sobre parafusos sendo axiais. Os furos são inseridos na estrutura em extrusões que tangenciam a parte principal da estrutura (Figura 2.12), visando minimizar a fragilidade.

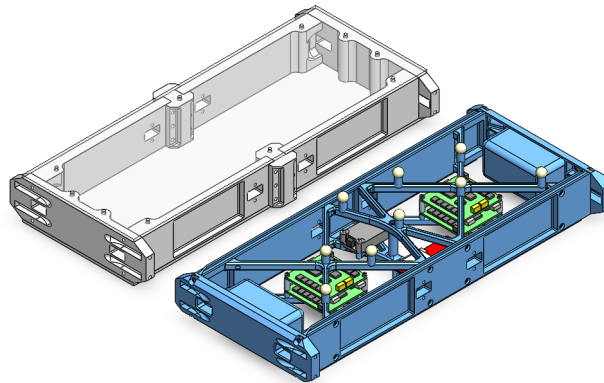


Figura 2.11: Comparação das bases do robô.

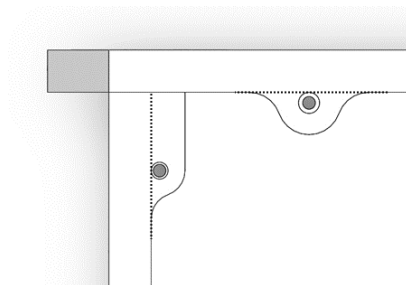
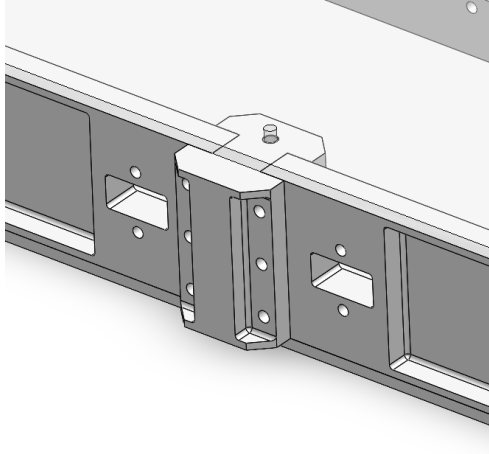


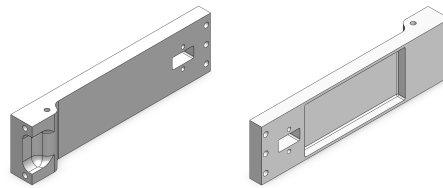
Figura 2.12: Furos tangentes a estrutura da base.

A estrutura lateral da nova base é tri-partida, dividida em: extremidades frontal e traseira (Figura 2.14(b)) e peça de conexão (Figura 2.13(a)). As

extremidades se encaixam com o módulo, tendo um rebaixo para ter espaço para o motor, além de possuir furos com insertos para se prender as peças frontais e a tampa. A peça de conexão, além de se encaixar nas extremidades, possui 6 furos para porcas M3 para conexão das extremidades.



2.13(a): Peças de conexão.



2.13(b): Extremidades da base.

Figura 2.13: Lateral Tripartida

### 2.2.3 Motores

As juntas do robô são atuadas através do *brushless outrunner* Antigravity MN4004 300Kv, fabricado pela empresa *T-Motor* (Figura 2.14(a)). As especificações principais do motor estão na Tabela 2.2. Para a sua utilização no módulo de atuação, o seu eixo é substituído por um novo, tendo um rasgo para o encaixe do pinhão e do *encoder* com um furo em "D", além de ter um rasgo para as moscas do estator em sua extremidade tangenciando a tampa do módulo e a traseira do rotor.

Antigravity MN4004 300Kv			
Kv	400	Potência	300 W
Diâmetro	4 mm	Lipo/Voltagem	4s(14,8 V)-6s(22,2 V)
Dimensões	44.35x19mm	Corrente de Pico	12 A

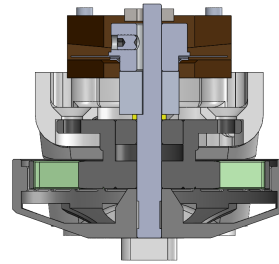
Tabela 2.2: Informações do motor.

### 2.2.4 Sensores

O módulo de atuação utiliza o *encoder* ótico E2 (Figura 2.15), da empresa *US Digital*, para quantificar a rotação do eixo do motor, assim calculando as posições dos elos e possibilitando o cálculo de suas velocidades.



2.14(a): Motor brushless MN4004.



2.14(b): Vista de corte do conjunto motor+pinhão+encoder.

Figura 2.14: Motor Antigravity MN4004 KV400.

As especificações técnicas do sensor estão na Tabela 2.3. No módulo, não é utilizada a *case* que vem com o aparato, ela é removida e seu disco óptico é preso diretamente no eixo do motor através de uma mosca.



Figura 2.15: Encoder E2.

Encoder	
Modelo	E2-5000-157-IE-D-D-B
Tipo	Óptico
Resolução	5.000
Eixo	4 mm
Saída	Onda Quadrática

Tabela 2.3: Informações do *encoder*.

### 2.3 Simulação Estática

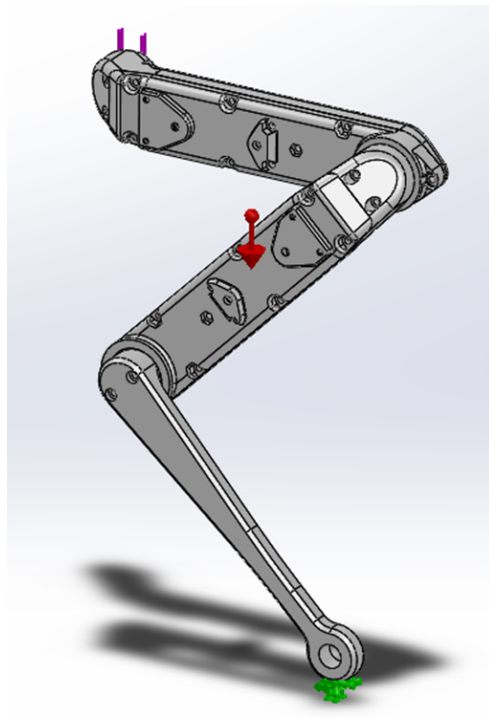
Para verificar quais partes da estrutura mecânica necessitariam de eventuais reforços, foi realizada nas primeiras versões do projeto uma simulação

estática usando elementos finitos, que divide o corpo em grãos e analisa os efeitos gerados na peça em resposta a pontos de fixação e cargas adicionadas a estrutura. O objetivo é antecipar se o projeto consegue suportar um aumento de peso devido a possíveis adaptações, com um peso estipulado de 3 quilogramas.

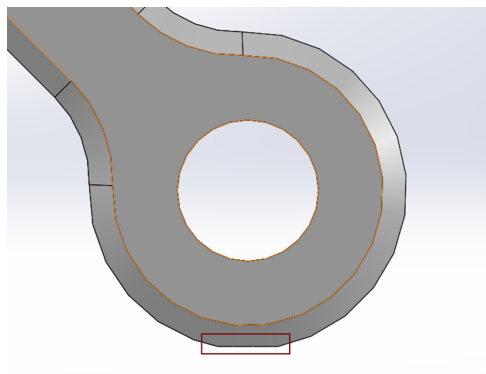
### 2.3.1

#### Parâmetros

Devido à complexidade do projeto e da restrição de desempenho dos computadores disponíveis, por meio do *SolidWorks*, é realizada a simulação do efeito da gravidade sobre uma perna só (Figura 2.16(a)), para simplificação do estudo. Foi aplicada a perna uma carga de 7,5 N, como se no total as 4 pernas estivessem recebendo juntas 30 N, o que seria o valor aproximado em relação à força para sustentar 3 quilogramas na base do robô. A força é aplicada na área de encaixe do módulo com o corpo, enquanto a fixação se dá na extremidade da peça, onde é realizado um pequeno corte (Figura 2.16(b)) para garantir tangencialidade com a superfície do chão, simulando um contato normal.



2.16(a): Configuração da simulação estática no *SolidWorks*.



2.16(b): Corte na extremidade da pata para simulação do contato com o terreno.

Figura 2.16: Configuração da simulação.

### 2.3.2 Resultados

A Figura 2.17 mostra o resultado da simulação. É possível observar uma variação de tensão de 468,9 Pa até 77,42 MPa, no entanto, as Figuras 2.18(a) e 2.18(b) mostram que as tensões acima de 6,45 MPa se concentram na extremidade da perna, especialmente na área de fixação, que simula o contato. Logo, pelas tensões mais altas serem em áreas de contato e em regiões pequenas, podem ser desconsideradas para análise.

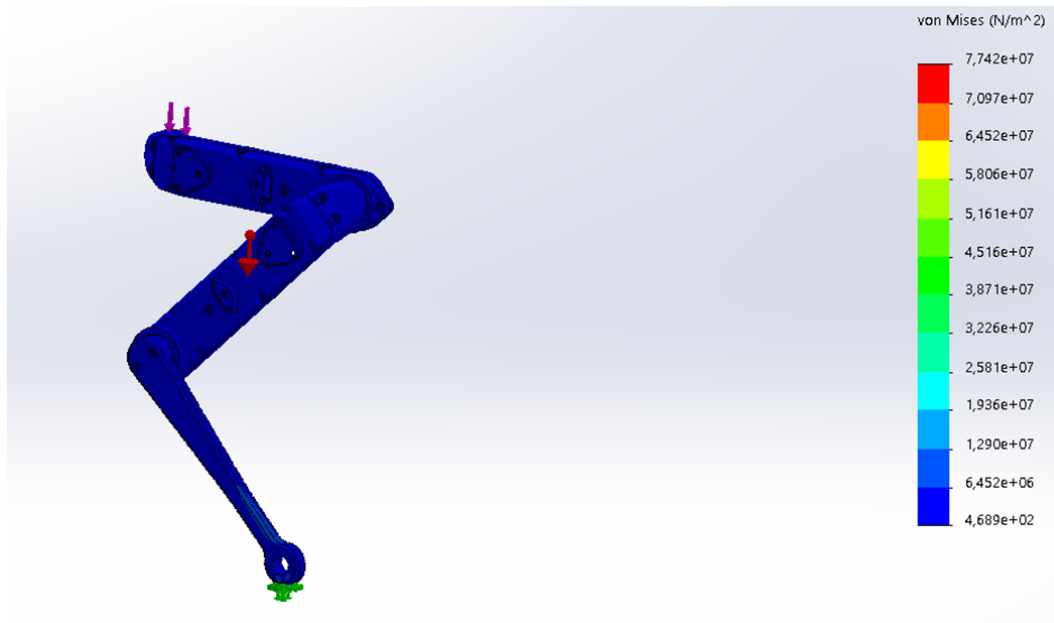
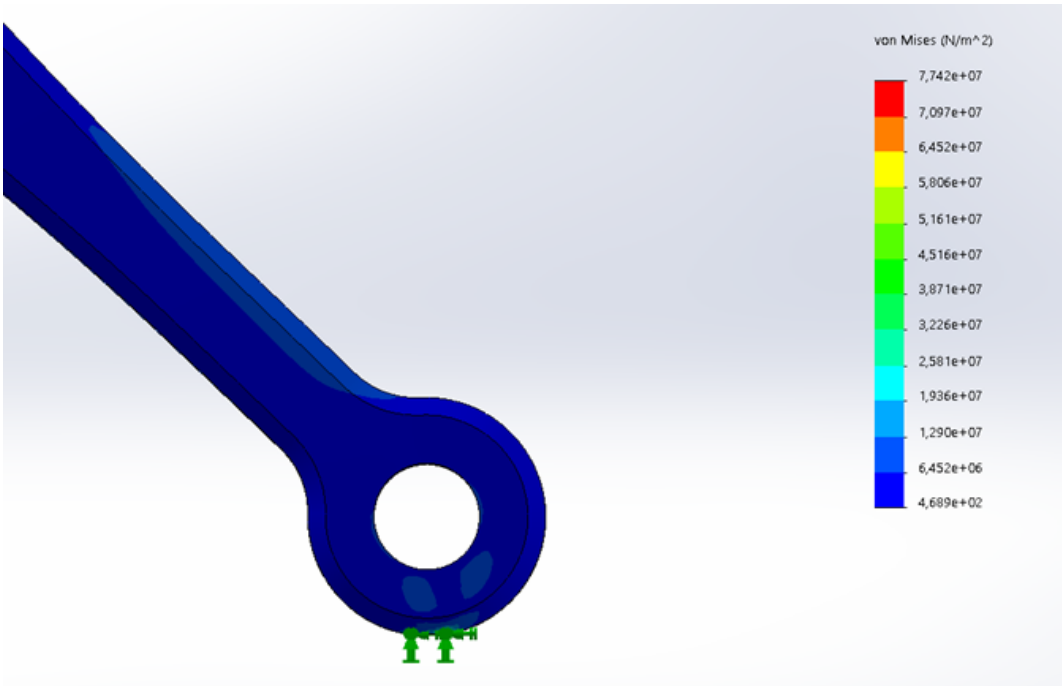


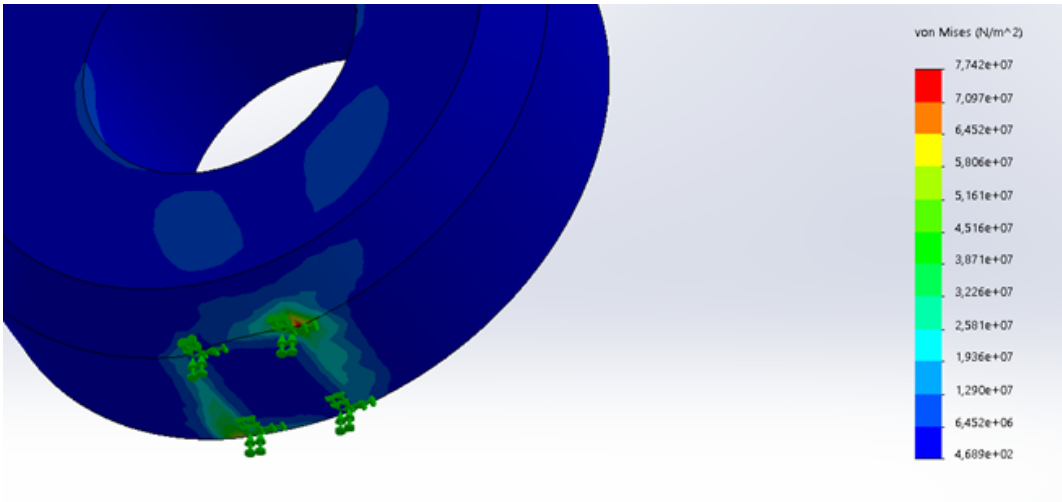
Figura 2.17: Resultado da simulação estática.

Para comparação com a tensão de escoamento de alguns materiais de impressão 3D, a tensão máxima obtida na simulação (6,45 MPa) é multiplicada por um fator de segurança para suportar efeitos dinâmicos e outras fragilidades não simuladas. É comum peças impressas serem produzidas com preenchimento parcial, e o próprio processo não garante isometria nas propriedades mecânicas. Outro ponto importante a se considerar são efeitos dinâmicos, que não foram simulados neste trabalho devido à complexidade do projeto e a ausência de um computador com um software adequado para realizar esse estudo. Com isso, adota-se um fator de 4 para a comparação entre os materiais mais comuns, como mostra a Tabela 2.4. Note que o valor máximo de tensão, mesmo após a aplicação do fator de segurança, permanece em um valor inferior ao do uso comum desses materiais para impressão 3D.





2.18(a): Extremidade da perna.



2.18(b): Concentração de tensão no ponto de fixação.

Figura 2.18: Resultados da simulação para a extremidade da perna.

Tensão Retirada da Simulação (MPa)	Tensão de Escoamento (MPa)		
	ABS	PETG	PLA
25,8	38	51	66

Tabela 2.4: Comparativo da tensão de escoamento de materiais comumente utilizados em impressão 3D.

## 3 Eletrônica

### 3.1 Eletrônica Original

A eletrônica embarcada do robô e seu funcionamento está descrita no diagrama da Figura 3.1. O sistema de atuação do robô possui como principais componentes o computador central e duas placas de circuito impresso (PCB) customizadas: a *MasterBoard* e a *MicroDriver*. Tanto o design como os arquivos de fabricação dessas placas estão disponíveis de forma aberta no *site* do projeto. Todo o controle e planejamento de trajetória do robô é realizado em um computador externo que se comunica com o robô através de uma rede *Wifi*, que possui uma certa latência de comunicação, ou através de um cabo de rede, que não possui quase nenhuma latência, mas é tem a desvantagem de precisar manter um cabo conectado ao robô constantemente. No computador central estão os controladores de movimento das pernas, que recebem os dados dos sensores do robô e geram os torques de referência para as juntas necessários para executar um determinado movimento.

A função da placa *MasterBoard* é coletar as informações dos sensores e transmitir essas informações para o computador central que controla o robô. Os sensores são: uma *IMU* (Unidade Inercial de Medição, composta por um giroscópio, um acelerômetro e um magnetômetro) na base do robô, e todos os sensores de posição, velocidade e torque dos motores. A *MasterBoard* recebe os torques, posições e velocidades de referência para as juntas, determinados pelo computador central, e transmite essas informações via SPI para as placas *MicroDriver*, responsáveis pela atuação dos motores. Cada placa *MicroDriver* pode atuar até dois motores e o controle de torque é realizado através do ajuste da corrente no motor.

O design da placa *MicroDriver* é baseado em uma placa de desenvolvimento disponibilizada comercialmente, produzida pela Texas Instruments, chamada *Launchpad F28069M with DRV8305 Booster Packs*. Os desenvolvedores do projeto fizeram uma nova versão desse driver de motor brushless para reduzir consideravelmente o seu tamanho e caber na base do robô, mantendo o seu formato mais compacto e leve.

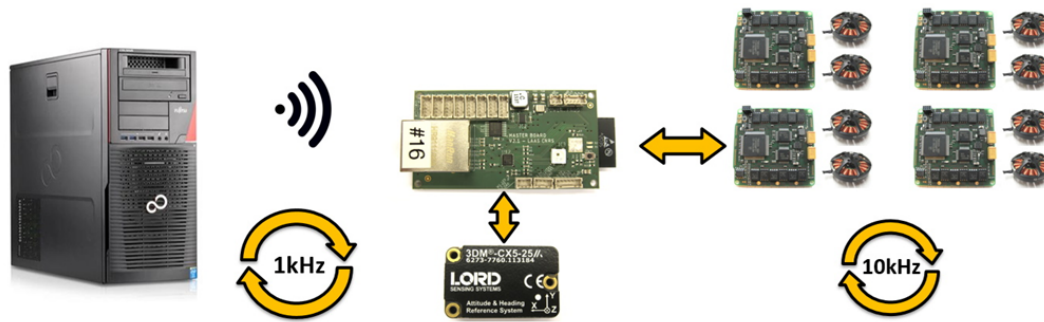


Figura 3.1: Diagrama da eletrônica do robô Solo. Adaptado de *Github*<sup>1</sup>.

### 3.1.1 Alterações

A pandemia do COVID-19 causou uma falta mundial no estoque de diversos componentes eletrônicos. Para viabilizar a fabricação das placas eletrônicas embarcadas no robô, a *Master Board* e a *MicroDriver*, foram necessárias algumas alterações no projeto das placas.

Na placa *MicroDriver*, a alteração principal foi o conversor DCDC de alimentação, substituído por outro com pinagem compatível e com as mesmas especificações de tensão de entrada, saída e corrente máxima. O DCDC escolhido foi o TPS7A2533DRVR, da Texas Instruments, destacado na Figura 3.2.

Na placa *MasterBoard*, apresentada na Figura 3.3, foram trocados os conectores de programação do microcontrolador, de comunicação serial e de entrada/saída GPIO. O conversor DCDC de alimentação foi substituído pelo LT1129IST-3.3 e a placa, que antes permitia comunicação via cabo de rede e *WiFi*, foi mantida apenas com a comunicação *WiFi*, usando o microcontrolador ESP32. Além disso, foi adicionado um novo conector I2C para permitir a utilização de uma *IMU* mais barata e simples do que a proposta no projeto original do robô.

## 3.2 Controlador *ODrive*

Devido a atrasos de fabricação e problemas de comunicação entre as placas da eletrônica original, foi necessário buscar controladores de motor brushless alternativos, que estivessem disponíveis comercialmente para viabilizar os testes das pernas em tempo hábil para o projeto. Para isso, foi escolhido o controlador *Odrive*, apresentado na Figura 3.4, comercializado pela empresa *Odrive*. Esse controlador é amplamente utilizado em projetos de robôs móveis com rodas e drones, e o próprio fabricante disponibiliza tutoriais para

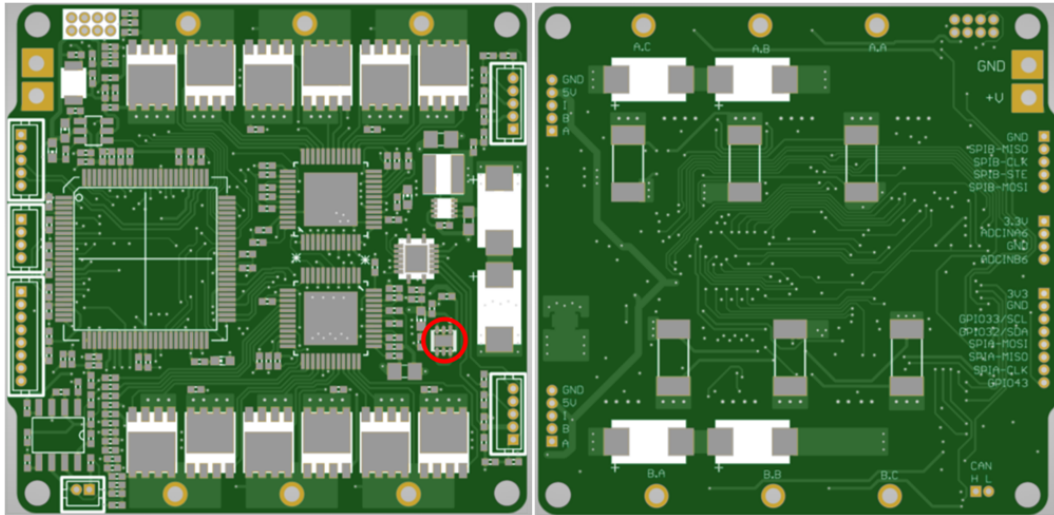


Figura 3.2: Placa MicroDriver atualizada.

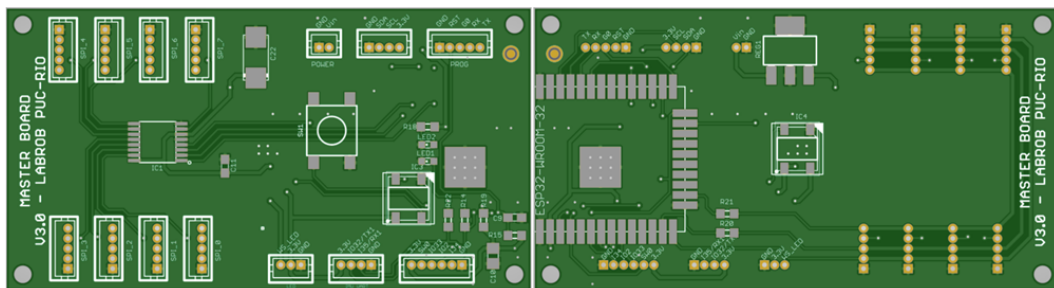


Figura 3.3: Placa Masterboard atualizada.

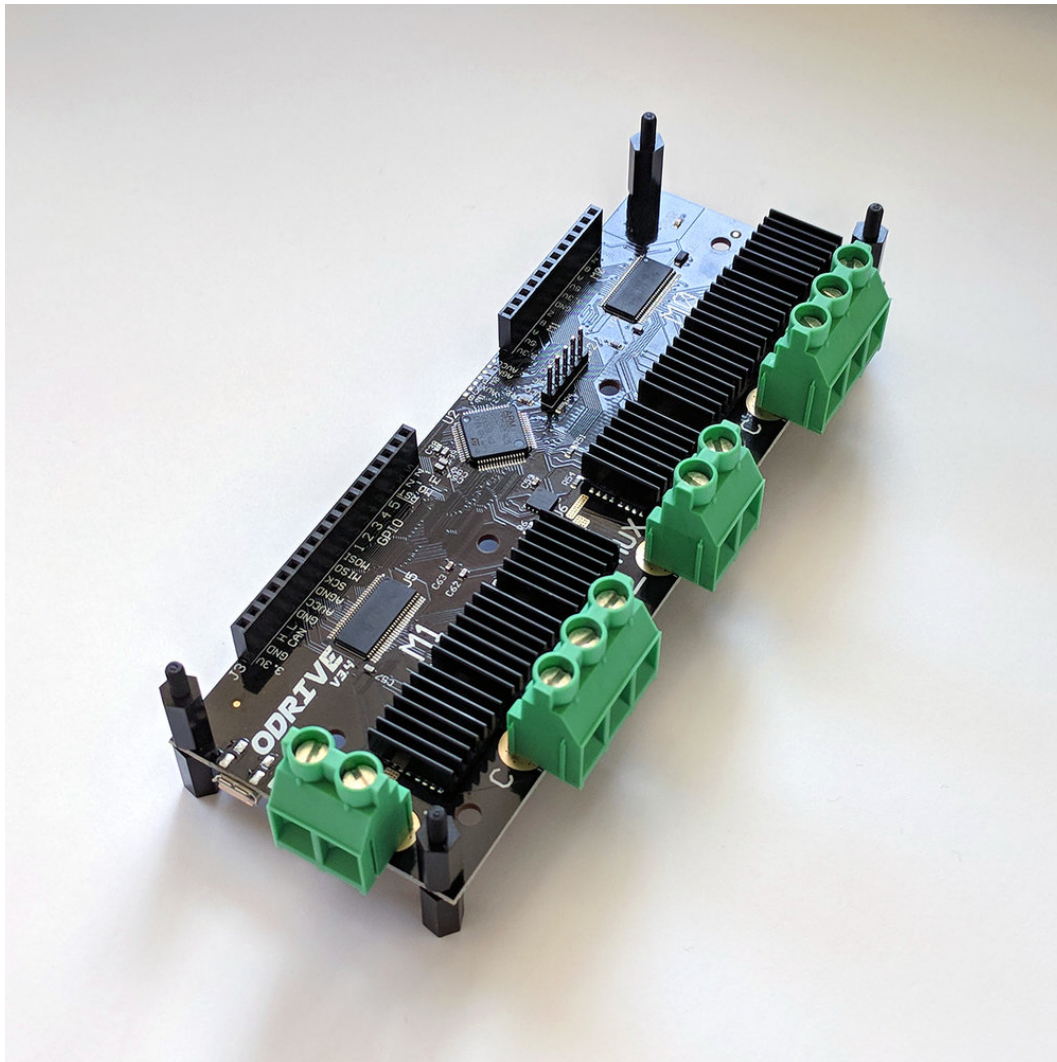
utilização, e um fórum de dúvidas no *Discord*.

O controlador *Odrive* possui dois canais, M0 e M1, que possibilitam controlar até dois motores de maneira simultânea. Além disso, possui uma entrada *micro-USB* que permite o envio de comandos e modificações de suas configurações por meio de um computador. A placa possui ainda entradas para conexão com *encoders* e com outras placas que podem ser usadas para enviar comandos, como um *Arduino*, por exemplo. Além disso, é possível realizar controle de torque, por meio da leitura e regulação de corrente, e controle de posição e velocidade, através da leitura de um *encoder*.

Nesse sentido, o controlador *Odrive* se assemelha a placa *MicroDrive*, tanto por ser uma placa que atende até dois motores, como por realizar os mesmos tipo de controle: posição, velocidade e torque. A única desvantagem é o seu tamanho, que é mais que o dobro da placa original. A sua utilização na versão final do robô exigiria um reajuste do projeto mecânico da base.

Para verificar a viabilidade de uso do *Odrive*, ao fabricante disponibiliza duas tabelas, uma de motores, mostrando seus desempenhos e características<sup>2</sup>,

<sup>2</sup>Tabela de Motores compatíveis <https://docs.google.com/spreadsheets/d/12vzz7XVEK6YNI0qH0jAz51F5VUpC-1JEs3mmkWP1H4Y/edit#gid=0>

Figura 3.4: Controlador *Odrive*.

e outra de *encoders*, com informações e tipos compatíveis<sup>3</sup>. Baseado nelas, se observa que o tanto o motor *brushless outrunner Antigravity MN4004 KV400* quanto o *encoder* óptico E2 do *Solo* são compatíveis.

### 3.2.1 Configuração

A partir do modelo *Odrive* 3.5, se realiza a configuração necessária para o motor e para o *encoder* do projeto, conforme mostrado na Tabela 3.1. Em relação ao motor *brushless outrunner Antigravity MN4004 KV400*, ele é configurado como um motor de Alta-Corrente, com 12 pares de polo (24 polos), as correntes de calibragem e máxima foram configuradas para permanecer abaixo do limite do motor, pois excesso de corrente poderia levar o motor a aquecer muito durante a operação, chegando a afetar negativamente seu

<sup>3</sup>Tabela de *Encoder* compatíveis: <https://docs.google.com/spreadsheets/d/10BDwYrBb5zUPZLrhL98ezZbg94tUsZcdTuwiVNgVqpU/edit#gid=0>

funcionamento. Além disso, como constante de torque, a própria empresa *Odrive*, orienta configurar como 8,27 dividido pelo valor da constante de velocidade do motor, disponibilizado no seu *datasheet*.

Para o *encoder* óptico E2, é necessário configurar o valor de CPR (counts per revolution), definido no manual do *Odrive* como quatro vezes o número de pulsos por volta. Como o *encoder* gera 5.000 pulsos por revolução, o valor do parâmetro CPR deve ser configurado para 20.000. Apesar do *encoder* possuir um pino de *Index* (Indicador de posição angular), desativa-se o seu uso no *Odrive* por não ser necessário.

Por último, são configurados os parâmetros gerais da placa em si, das correntes que atuam sobre ela e foi ativado o resistor de frenagem. Esse procedimento de configuração tem que ser sintonizado até que o procedimento de calibração dos motores seja executado sem erros. A Tabela 3.1 mostra os parâmetros finais obtidos para o *Odrive*.

Motor	Pares de Polo	12	
	Constante de Torque	8.27/300	N.m
	Corrente Limite	7.0	A
	Corrente de Calibragem	5.0	A
	Máxima Voltagem	5.0	V
	Tipo de Motor	Alta-Corrente	
Encoder	CPR	20000	
	Bandwith	3000	
	Uso do Index	Desativado	
Geral	Velocidade Limite	50.0	Turns/s
	Resistor de Freio	Ativado	
	Corrente Máxima	60.0	A
	Corrente Negativa Máxima	-10.0	A

Tabela 3.1: Configuração.

Com a configuração e a calibragem feitas, realizaram-se testes de controle de posição, velocidade e torque. Nos testes de posição<sup>4</sup>, foram utilizados com valores de referência na sequência: 0,25, 0.50, 0.75, 1, 5/6, 2/3, 1/3 e 0, onde a unidade da posição angular desejada é em *turns*, ou seja, voltas do motor. Para o teste de velocidade<sup>5</sup>, foram testados diferentes valores de velocidades em sentidos diferentes e foi verificado que o parâmetro de limite de velocidade configurado é sempre respeitado, mesmo se for dado um comando fora dos limites. Para testar o torque, foi utilizada uma peça impressa presa ao eixo do motor, que realizava força sobre uma balança, como mostra a Figura 3.5. O

<sup>4</sup>Testes de posição com o *Odrive*: <https://www.youtube.com/shorts/BD9NWPc8rhY>

<sup>5</sup>Testes de velocidade com o *Odrive*: <https://www.youtube.com/watch?v=-9gz2znvB6w>



valor do peso medido pela balança deveria ser o valor do comando de torque dividido pelo raio da peça.



Figura 3.5: Estrutura experimental para testes de controle de torque dos motores.

Durante os testes com o controlador, foram sintonizados de forma empírica os ganhos para o controlador PID (proporcional, integral e derivativo) de posição e velocidade, como mostrado na Tabela 3.2.

Ganhos:					
P:	50.00	I:	0.00	D:	0.02

Tabela 3.2: Ganhos do controlador PID.

## 4

### Fabricação e montagem

Com o novo projeto pronto (Figura 4.1), foi realizada a construção definitiva do robô. As seções a seguir apresentam o procedimento de montagem dos componentes principais do sistema.

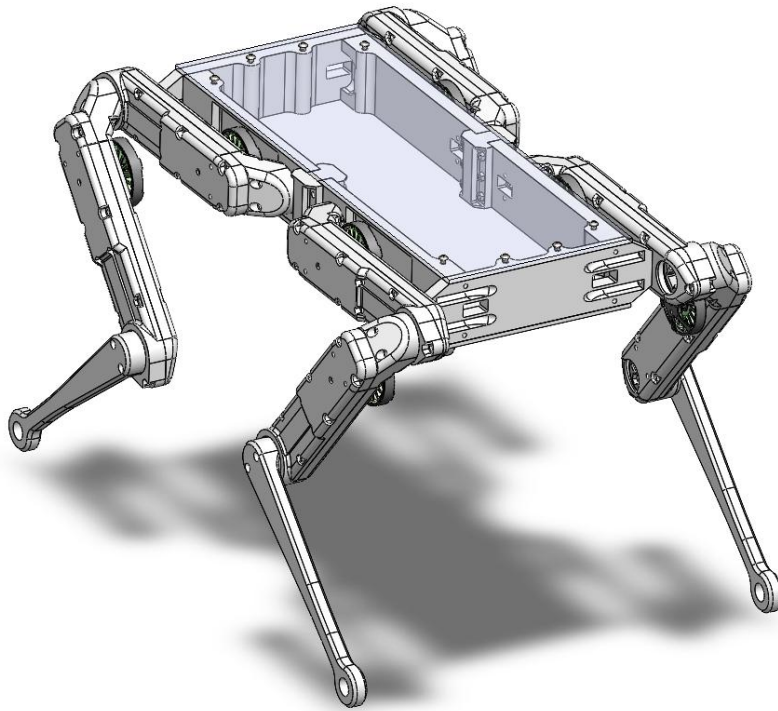


Figura 4.1: Projeto final.

#### 4.1

##### Motor

Para cada motor, faz-se necessário trocar o eixo original por um mais apropriado ao projeto. O primeiro passo é retirar o colar do motor (lado esquerdo na Figura 4.2), assim, podendo retirar o colar de latão e as moscas que prendem o eixo ao estator. Tirando o colar, as peças retiradas são separadas para serem colocadas de volta depois.

Em seguida, conforme mostra a Figura 4.3, separa-se o rotor do estator para então o encaixá-lo numa peça de cilindro de alumínio vazado. Com isso, há um apoio para a retirada do eixo, podendo usar tanto uma prensa ou um martelo, e ainda evitando o risco de flexão da deformação da peça ao usar a região de entrada do estator como apoio.





Figura 4.2: Retirada do colar, anel e moscas.

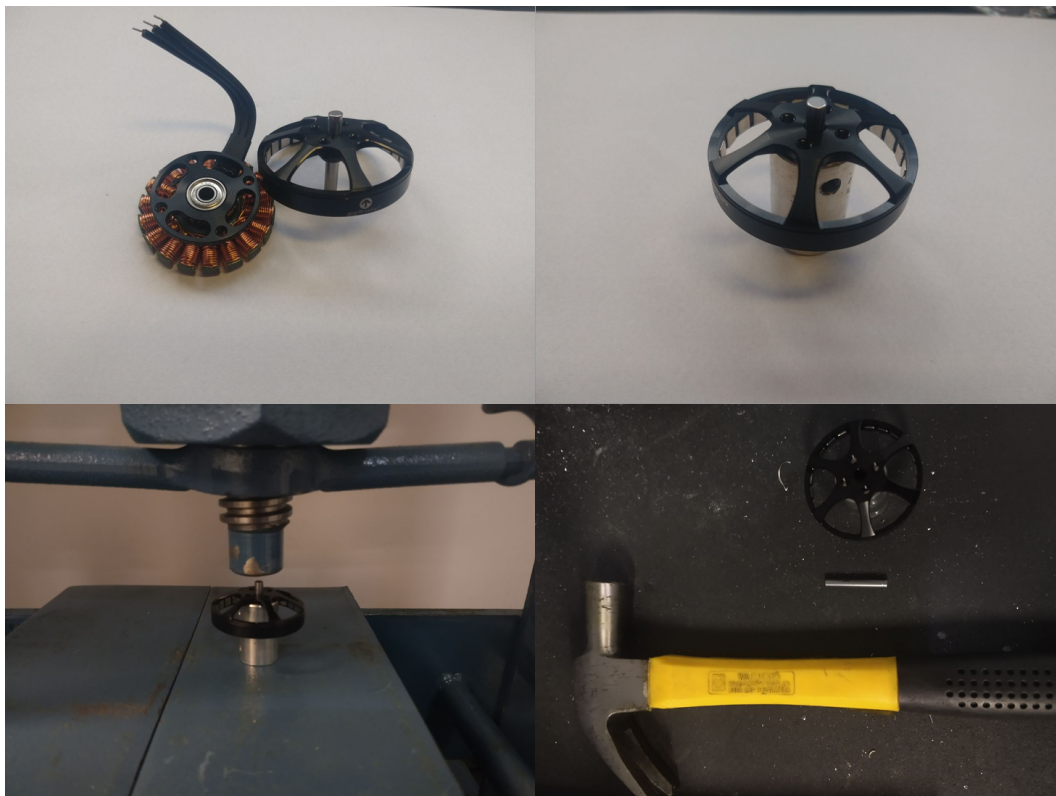


Figura 4.3: Retirada do eixo antigo.

Após o processo, reúne-se o rotor, estator, as moscas, o anel de latão, o pinhão e o novo eixo usinado para a montagem (lado esquerdo da Figura 4.4). O novo eixo deve ser colocado primeiro no rotor e em seguida preso pelas moscas. Em seguida, monta-se o estator, o anel de latão e o pinhão, respectivamente (lado direito da Figura 4.4). Após a extensão dos fios, o motor fica pronto para ser colocado no módulo de atuação.



Figura 4.4: Montagem do Motor com novo eixo.

## 4.2 Corpo

Para a fabricação das peças impressas, são utilizados as impressoras do modelo *Ender 3 v2* e *Ultimaker S3*. Como material, o *ABS* é usado para as polias, enquanto para a estrutura se usa tanto *ABS* quanto *PETG*, materiais já verificados que suportam as tensões previstas de uso.

Com as peças prontas, para a montagem do módulo de atuação, primeiro são inseridos os rolamentos da base e da tampa do módulo. Os passos seguintes são a fixação do motor, a montagem da transmissão de polias e correias, a montagem e fixação do disco óptico e, por último, o posicionamento do *encoder* fixado por pinos de latão, como mostra a Figura 4.5.

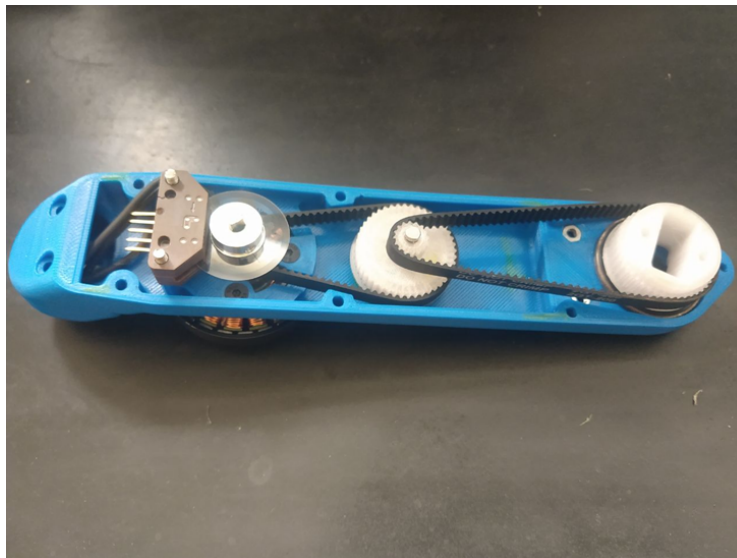


Figura 4.5: Montagem do módulo de atuação.

Com os módulos de atuação montados, é realizada a montagem da perna, se encaixando dois módulos sequencialmente, como mostra a Figura 4.6. O

último elo é o único diferente, sem o módulo de atuação, somente um elo rígido para o contato com o terreno.



Figura 4.6: Montagem de uma perna

Como pode ser visto na Figura 4.7, as quatro pernas são fixadas no corpo do robô, cada uma presa em dois pontos das laterais, com os fios direcionados para parte interna. Para finalizar, seria necessário montar a eletrônica interna completa e o fechamento com as tampas de policarbonato, no entanto, como um *driver* de motor diferente precisou ser utilizado, a montagem da eletrônica interna ficou como trabalho futuro. Os testes foram feitos com os *Odrives* na parte externa da base. A Figura 4.8 mostra o robô completo apoiado sobre uma caixa, demonstrando como ele ficaria de pé.

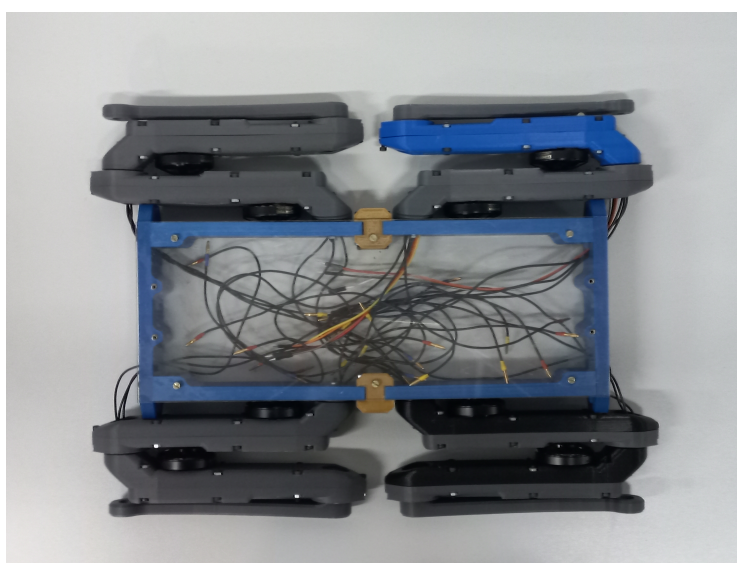


Figura 4.7: Corpo montado.





Figura 4.8: Robô apoiado em caixa.

### 4.3

#### Estimativa de Custo

Durante o processo da construção e de compras de material, é importante separar e estimar o custo básico para a construção do robô, sem incluir peças adicionais para troca e manutenção. Essa conta é importante para validar a acessibilidade do projeto como plataforma experimental, umas das suas principais motivações.

Os primeiros gastos principais são de compra de componentes, como os motores, os sensores, as correias e os rolamentos, importantes para o funcionamento dos módulos de atuação. A Tabela 4.1 mostra o custo de todos os componentes. Os itens marcados em azul são compras internacionais e têm seus valores influenciados pela cotação do dólar no momento da compra. Além disso, vale ressaltar que o *Encoder* tem seu preço influenciado pela quantidade de compra, podendo ficar mais caro se comprado em menores quantidades.

Para as peças impressas (Tabela 4.2), foi utilizado a estimativa de custo de impressão de cada componente, considerando o uso de máquina e o volume de material. No entanto, é importante ressaltar que esses custos podem variar dependendo do modelo da impressora 3D, e é importante ajustar corretamente os parâmetros de impressão para cada modelo, para que se evite a necessidade de reimpressão, gerando gastos extras.

Custo de compras			
Peça	Qtd	Preço	Total
<i>Antigravity</i> MN4004 KV400/ 2 peças	4	R\$ 743,12	R\$ 2.972,48
<i>Encoder</i> E2-5000-157-IE-D-D-B	8	R\$ 301,03	R\$ 2.408,28
150-2GT-6 - 2 correias	4	R\$ 40,63	R\$ 162,52
200-2GT-6 - 3 correias	3	R\$ 35,00	R\$ 105,00
Rolamento 8x4x2mm - 10 Peças	3	R\$ 46,47	R\$ 139,41
Rolamento 25x32x4mm - 2 Peças	8	R\$ 15,31	R\$ 122,48
Rolamento axial 10x24x2/ 10 peças	1	R\$ 39,08	R\$ 39,08
Custo Final:			R\$ 5.949,25

Tabela 4.1: Custo de compras

Peças Impressas			
Peças	Qtd	Cst. Unitário	Cst. Total
Base Atuador	8	R\$ 12,40	R\$ 99,20
Tampa Atuador	8	R\$ 5,68	R\$ 45,44
Perna	4	R\$ 4,54	R\$ 18,16
Polia Motor	8	R\$ 0,15	R\$ 1,20
Polia Meio	8	R\$ 0,79	R\$ 6,32
Polia Final	8	R\$ 1,30	R\$ 10,40
Conexão Lateral	2	R\$ 4,62	R\$ 9,24
Lateral	2	R\$ 10,30	R\$ 20,60
Frente	4	R\$ 13,34	R\$ 53,36
Custo Total:			R\$ 263,92

Tabela 4.2: Custos de impressão 3D.

O custo dos componentes da eletrônica original estão apresentados na Tabela 4.3. Devido à sua complexidade, a produção e a montagem das placas *MicroDriver* foram realizadas por uma empresa nacional especializada nessa área. Para gravar o código embarcada na placa *MicroDriver*, é necessário o uso do gravador TMS320-XDS100-V3, cujo custo também está na tabela. Em "componentes diversos", relatam-se um conjunto de itens menores necessários para a montagem das placas e outros itens de baixo valor, que podem ser vistos na página do *Github*<sup>1</sup>. O custo da eletrônica alternativa com o controlador *Odrive* não foi realizado.

Para materiais básicos (Tabela 4.4), foram incluídos o custo de matéria-prima para usinagem mais eixos e parafusos. Para as peças que foram usinadas por empresas, está incluído também o custo do serviço. O alto valor dos eixos dos motores é porque se trata de um serviço de usinagem que necessita grande precisão e de ferramentas adequadas para o tamanho da peça.

Com a soma de todos os custos, é possível estimar o custo total para

<sup>1</sup>Link de Eletrônica [https://github.com/open-dynamic-robot-initiative/open\\_robot\\_actuator\\_hardware/tree/master/electronics](https://github.com/open-dynamic-robot-initiative/open_robot_actuator_hardware/tree/master/electronics)

Eletrônica	
Componente	Custo
<i>MicroDriver</i> (Fabricação + Montagem)	R\$ 5.226,26
<i>Flasher</i> - TMS320-XDS100-V3	R\$ 289,15
Cabos Hirose	R\$ 119,70
<i>Masterboard</i>	R\$ 96,78
Componentes Diversos	R\$ 721,75
Total	R\$ 6.453,64

Tabela 4.3: Custo da eletrônica original.

Material	
Peças	Custo
Material de Eixo e Usinagem	R\$ 640,00
Chapas de Policarbonato	R\$ 40,00
Eixos de Latão	R\$ 27,80
Parafusos	R\$ 100,00
Total	R\$ 807,80

Tabela 4.4: Custo de materiais mecânicos.

a construção do projeto sendo por volta de R\$ 13.474,61, como mostra a Tabela 4.5. Como comparação, utiliza-se o preço do robô *Spot*, da empresa *Boston Dynamics*, divulgado no artigo do *Washington Post*[17], que na época da publicação era de USD \$ 74.500, o que aproximado para reais é R\$ 372.500, sendo cerca de 27,6 vezes o custo do projeto.

Custo Total Estimado	
Tipo	Custo
Eletrônica	R\$ 6.453,64
Compras	R\$ 5.949,25
Material	R\$ 807,80
Impressos	R\$ 263,92
Total	R\$ 13.474,61

Tabela 4.5: Custo total estimado do projeto.

## 5

### Modelagem

Robôs quadrúpedes são modelados como robôs de base flutuante [7], ou seja, o deslocamento da base acontece devido às forças nos pontos de contato, com restrições cinemáticas de velocidade nula da extremidade da para quando em contato com o solo, como ilustra a Figura 5.1.

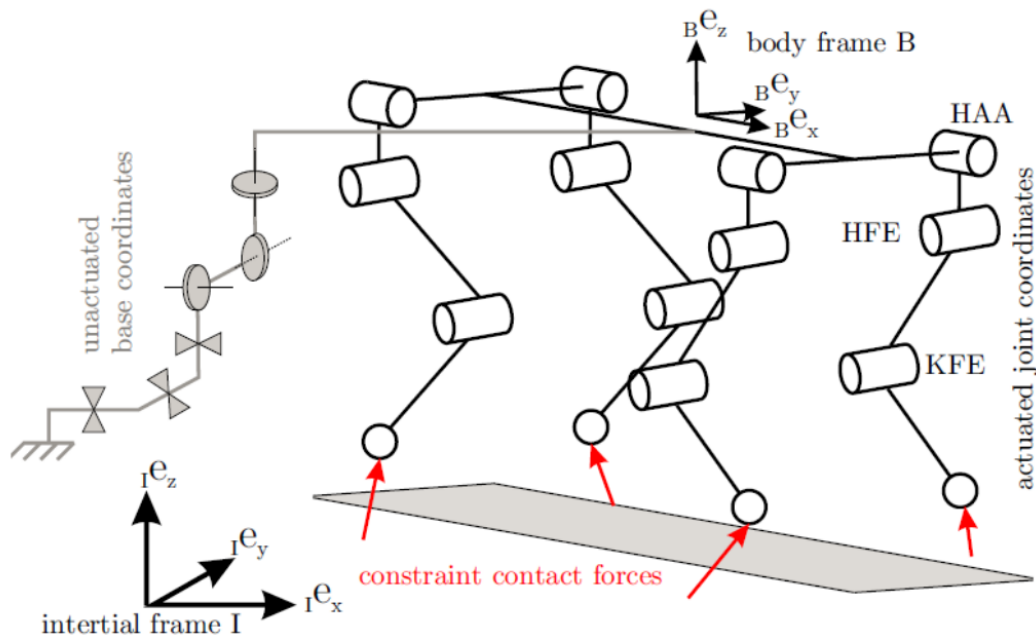


Figura 5.1: Representação do modelo dinâmico de um robô com pernas.

#### 5.1

##### Cinemática

Devido à complexidade da modelagem de um robô quadrúpede completo, este trabalho apresenta somente a modelagem de uma perna do robô, para o estudo de sua movimentação. Nesse caso, considere que a perna está ao ar livre e a base do robô é fixa, fazendo com a perna opere como um manipulador de dois graus de liberdade.

Para a modelagem cinemática, se utiliza o esquemático da Figura 5.2, onde  $D$  é a distância entre a primeira junta até o centroide da base do robô,  $L$  é o comprimento dos dois elos da perna,  $J_1$  e  $J_2$  são as juntas de rotação,  $\theta_1$  e  $\theta_2$  são os deslocamentos das juntas, e  $P$  é o ponto final da perna, definido em relação à base do robô.

A partir do diagrama proposto, obtém-se a Equação 5-1, que relaciona a posição do ponto  $P$  com as posições angulares das juntas,  $\theta_1$  e  $\theta_2$ .

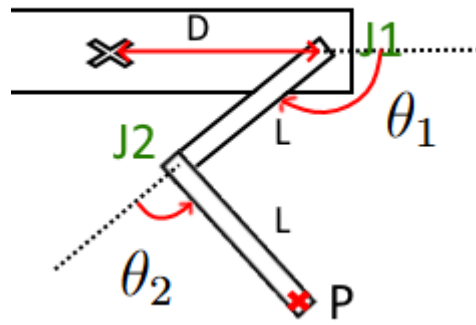


Figura 5.2: Configuração cinemática da perna do robô.

$$P(\theta_1, \theta_2) = \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} D + L * \cos(\theta_1) + L * \cos(\theta_1 + \theta_2) \\ L * \sin(\theta_1) + L * \sin(\theta_1 + \theta_2) \end{bmatrix} \quad (5-1)$$

Na Equação 5-2, define-se a matriz Jacobiana, que relaciona as velocidades das juntas,  $\dot{\theta}_1$  e  $\dot{\theta}_2$ , com a velocidade da extremidade da pata, dada por  $V(\theta_1, \theta_2)$ .

$$V(\theta_1, \theta_2) = \begin{bmatrix} -L * \sin(\theta_1) - L * \sin(\theta_1 + \theta_2) & -L * \sin(\theta_1 + \theta_2) \\ L * \cos(\theta_1) + L * \cos(\theta_1 + \theta_2) & L * \cos(\theta_1 + \theta_2) \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \quad (5-2)$$

## 5.2 Dinâmica

Para o equacionamento da dinâmica da perna do robô, se utiliza as massas dos elos ( $M_1$  e  $M_2$ ), seus momentos de inércias ( $I_1$  e  $I_2$ ), a posição do centro de massa ao longo do seu comprimento ( $X_{cm1}$  e  $X_{cm2}$ ), e os efeitos de gravidade sobre cada elo, conforme mostra a Figura 5.3.

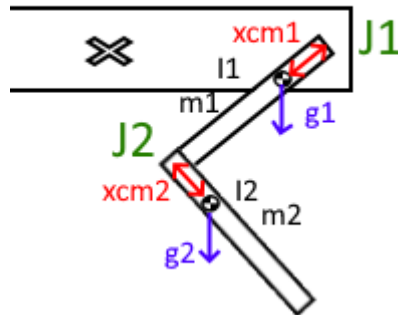


Figura 5.3: Termos Dinâmico

O equacionamento envolve a aplicação das Leis de Newton-Euler para obter a relação entre o torque  $\tau$  nas juntas atuadas e o movimento do robô,



como mostra a Equação 5-3 [18]. Nela, a matriz  $M(q)$  é a matriz de massa,  $C(q, \dot{q})$  são os termos referentes aos efeitos centrífugos e de Coriolis, e  $G(q)$  são os termos gravitacionais.

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) \quad (5-3)$$

No entanto, no caso do módulo de atuação, existem outros elementos para se considerar, como a redução e os elementos dinâmicos do motor. Para isso, adapta-se a Equação 5-4 [19], onde  $\tau_m$  é o torque do motor,  $I_m$  e  $B_m$  são, respectivamente, a inércia e o coeficiente de atrito viscoso do motor, e  $r$  é a relação de transmissão.

$$\tau_m = I_m\ddot{q}_m + B_m\dot{q}_m + \frac{1}{r}(M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q)) \quad (5-4)$$

No caso do motor Antigravity MN4004 300KV, para simplificação do equacionamento, assume-se que seus fatores dinâmicos são desprezíveis devido ao seu baixo peso e sua aplicação no projeto ser em velocidade baixas. Com isso, a equação dinâmica fica da forma:

$$\tau_m = \frac{1}{r}(M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q)) \quad (5-5)$$

### 5.3

#### Parâmetros

Para a obtenção dos valores dos parâmetros para a modelagem, foram utilizadas as informações geradas a partir dos modelos tridimensionais no *SolidWorks*, que consegue estimar massa, centro de massa, inércias e outros parâmetros dos modelos que compõem o desenho do robô. A partir disso, se convertem os valores para *SI* para a sua utilização na modelagem, como mostra a Tabela 5.1.

Parâmetros				
Medidas	SolidWorks		Em S.I	
	Valores	Unidade	Valores	Unidade
D	190	mm	0,19	m
L	160	mm	0,16	m
$X_{cm1}$	77,53	mm	77,53E-03	m
$X_{cm2}$	63,87	mm	63,87E-03	m
$M_1$	181	g	0,181	Kg
$M_2$	35,71	g	35,71E-03	Kg
$I_1$	1,63E+06	g.mm <sup>2</sup>	1,63E-03	kg.m <sup>2</sup>
$I_2$	2,58E+05	g.mm <sup>2</sup>	2,58E-04	kg.m <sup>2</sup>

Tabela 5.1: Tabela de Parâmetros

## 6 Simulação

### 6.1

#### ***Machines in Motion Laboratory***

A *Machines in Motion Laboratory* (MIM) disponibiliza no seu repositório do *GitHub*<sup>1</sup> uma biblioteca que contém programas ligados a controle e simulação de robôs quadrúpedes, baseados na biblioteca *Pinocchio*<sup>2</sup>, e também códigos que planejam e simulam trajetórias. Além disso, possui os modelos tridimensionais inclusos, como o próprio *Solo*, base desse projeto.

A partir destas bibliotecas, realizam-se estudos dinâmicos e de planejamento de trajetória para futura aplicação no projeto real.

#### 6.1.1

##### ***Kino-Dynamic Trajectory Optimization***

A *Kino-Dynamic Trajectory Optimization* é um dos programas presentes na biblioteca do MIM Lab, cuja função é otimização de trajetórias para robôs com pernas, podendo ser aplicado em modelos já existente de robôs, como o *Solo*. Para isso, é preciso configurar alguns parâmetros da trajetória desejada, como a posição dos pontos de contato das pernas do robô, o padrão de marcha, a posição inicial e final da base, a duração da trajetória, entre outros parâmetros. O programa também mostra a previsão da trajetória sendo executada pelo robô em um simulador dinâmico aberto, o *RaiSim*<sup>3</sup>.

#### 6.1.2

##### ***Stable Walking***

O teste realizado com *Kino-Dynamic Trajectory Optimization* foi o planejamento e a simulação de trajetória de caminhada estaticamente estável. Nela, sempre há no mínimo 3 pernas apoiadas no chão, com o centro de massa projetando-se dentro da região entre os 3 apoios, conforme ilustra a Figura 6.1.

A Tabela 6.1 mostra o padrão de marcha para esse tipo de caminhada, com a sequência de contatos das patas: os quadrados brancos indicam a pata erguida e os quadrados pretos indicam as patas em contato com chão. Um

---

<sup>1</sup>Repositório da *Machines in Motion Laboratory* <https://github.com/machines-in-motion>

<sup>2</sup>Biblioteca para computação de modelos dinâmicos para robôs ou estruturas rígidas articuladas.

<sup>3</sup>Site da *RaiSim*: <https://raisim.com/>

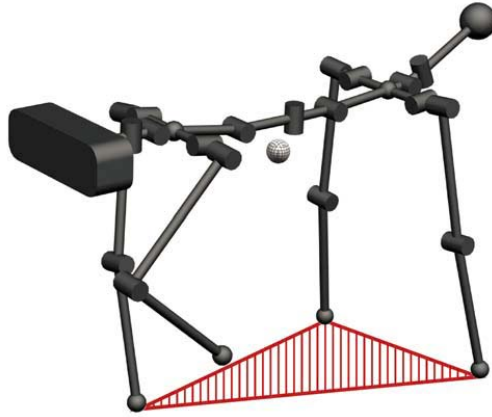


Figura 6.1: Ilustração de uma trajetória estaticamente estável com um robô quadrúpede.

ciclo termina após o movimento de todas as patas. As siglas  $LH$ ,  $RF$ ,  $RH$  e  $LF$ , representam respectivamente as seguintes as patas: esquerda traseira (*Left-Hind*), direita dianteira (*Right-Forward*), direita traseira (*Right-Hind*) e esquerda dianteira (*Left-Forward*).

Stable-Walking Gait									
LH									
RF									
RH									
LF									

Tabela 6.1: Stable walking gait.

Após a definição do tipo de caminhada, define-se as posições das patas ao longo do tempo conforme as Tabelas 6.2, 6.3, 6.4 e 6.5. Elas servem para definir os momentos em que cada pata está em contato com o chão, a duração de cada fase de contato, e as posições e orientações (representada usando quatérnios) da pata em relação ao ponto inicial do robô.

Pata	Momento		Posição			Orientação			
	$T_x$	$T_{x+1}$	x	y	z	w	qx	qy	qz
LH	0,00	1,00	-0,20	0,15	0,0	1	0	0	0
	1,25	6,00	-0,10	0,15	0,0	1	0	0	0
	6,25	11,00	0,00	0,15	0,0	1	0	0	0
	11,25	16,00	0,10	0,15	0,0	1	0	0	0

Tabela 6.2: Perna traseira esquerda.

Pata	Momento		Posição			Orientação			
	Tx	Tx+1	x	y	z	w	qx	qy	qz
RF	0,00	2,25	0,20	-0,15	0,0	1	0	0	0
	2,50	7,25	0,30	-0,15	0,0	1	0	0	0
	7,50	12,25	0,40	-0,15	0,0	1	0	0	0
	12,50	16,00	0,50	-0,15	0,0	1	0	0	0

Tabela 6.3: Pata direita frontal.

Pata	Momento		Posição			Orientação			
	Tx	Tx+1	x	y	z	w	qx	qy	qz
RH	0,00	3,50	-0,20	-0,15	0,0	1	0	0	0
	3,75	8,50	-0,10	-0,15	0,0	1	0	0	0
	8,75	13,50	0,00	-0,15	0,0	1	0	0	0
	13,75	16,00	0,10	-0,15	0,0	1	0	0	0

Tabela 6.4: Perna direita traseira.

Pata	Momento		Posição			Orientação			
	Tx	Tx+1	x	y	z	w	qx	qy	qz
LF	0,00	4,75	0,20	0,15	0,0	1	0	0	0
	5,00	9,75	0,30	0,15	0,0	1	0	0	0
	10,00	14,75	0,40	0,15	0,0	1	0	0	0
	15,00	16,00	0,50	0,15	0,0	1	0	0	0

Tabela 6.5: Pata esquerda dianteira.

### 6.1.3 Resultado

O resultado da trajetória otimizada pela *Kino-Dynamic Trajectory Optimization* está disponível no vídeo da simulação, disponível em: <https://www.youtube.com/watch?v=cr5BtWdN88I>.

A Figura 6.2 mostra *snapshots* do robô executando a trajetória no *RaiSim*. No primeiro quadro, ele está movendo a pata direita dianteira, no segundo quadro está movendo direita traseira, no terceiro a esquerda frontal e no último ele começa a mover a esquerda traseira.

Na Figura 6.3 mostra o gráfico do movimento das patas do robô durante a execução da trajetória, com a linha azul mostrando a posição obtida na simulação e o laranja a posição desejado. No eixo X, o eixo de direção da caminhada, o robô consegue seguir a trajetória desejada corretamente, enquanto no eixo Z, movimento vertical, e no Y, movimento lateral, há a presença de pequenas oscilações, mas na ordem de milímetros que são comuns com a vibração do robô.

A Figura 6.4 mostra a movimentação do centro da base do robô ao longo da trajetória. No eixo X, é visível o deslocamento para frente do robô, com as

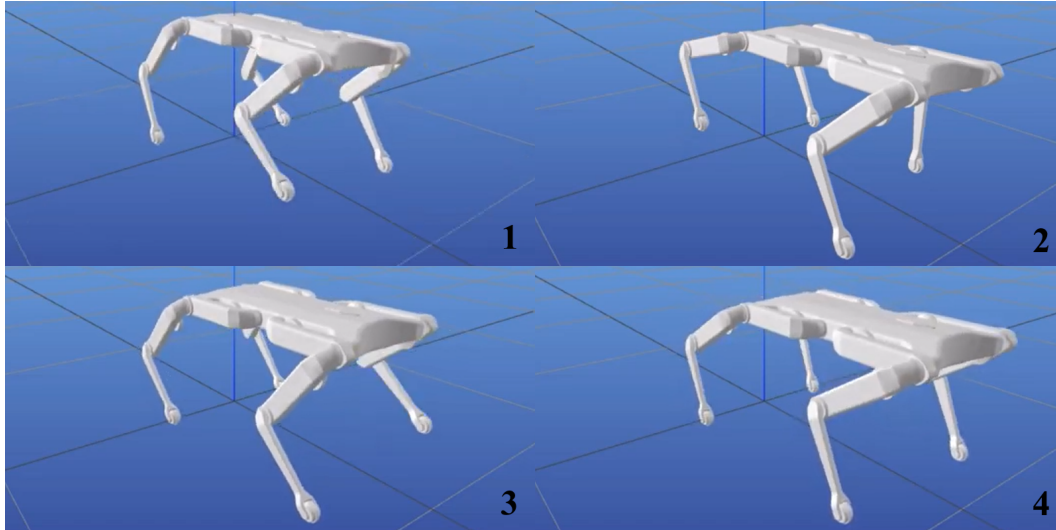
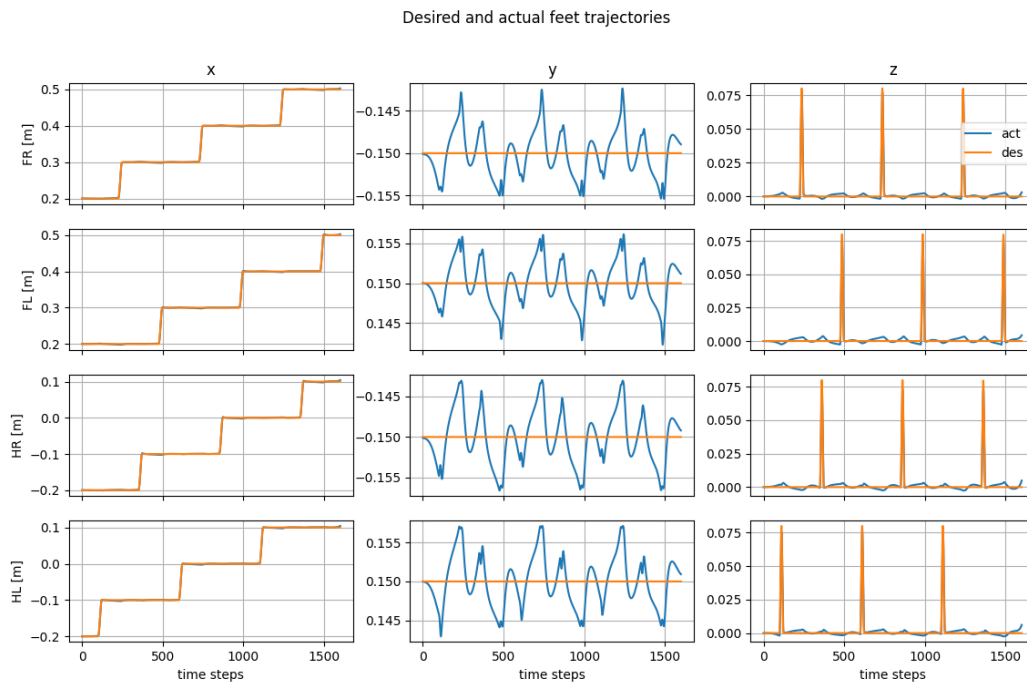
Figura 6.2: Movimento *Stable Walking*.

Figura 6.3: Posições das patas do robô ao longo da trajetória.

oscilações sendo consequências do próprio movimento das patas que afetam a dinâmica dele. No eixo Y, que mostra o movimento lateral, notam-se oscilações de até 20 mm, que também coincidem como o movimento das patas, repetindo um padrão de cada ciclo da marcha para manter a estabilidade. O movimento no eixo Z demonstra uma oscilação esperada de altura a cada passo.

Para minimizar as oscilações, é possível fazer melhor estudo dos parâmetros da otimização, no entanto, o resultado gerado está dentro do esperado. O próximo passo seria o teste dessa trajetória em outros simuladores, como o *PyBullet* ou o *Gazebo*.

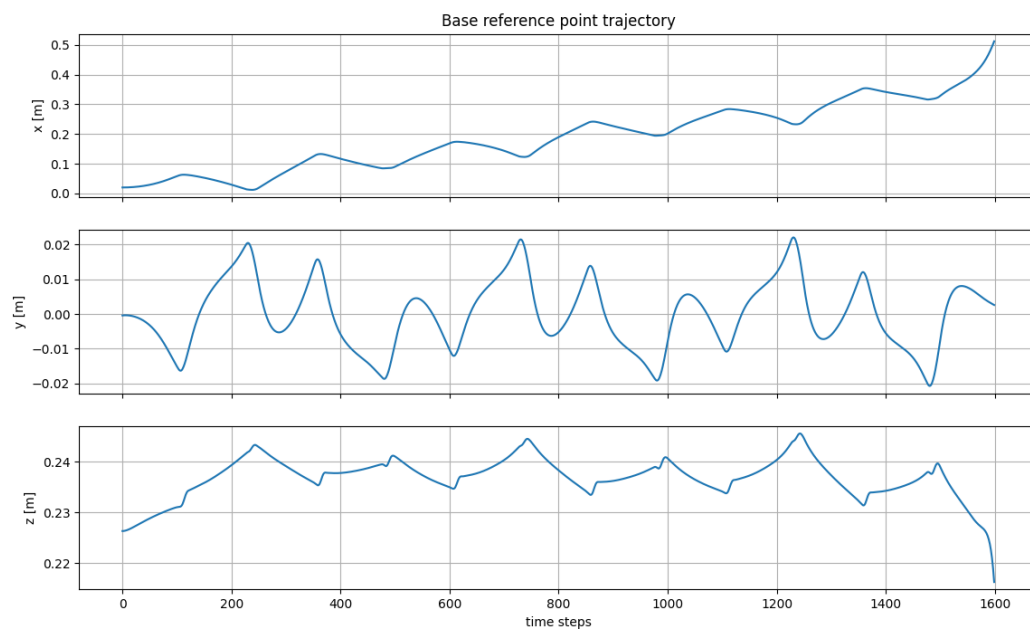


Figura 6.4: Deslocamento.

## 7

### Testes Experimentais

Devido aos problemas encontrados na eletrônica, detalhados na Seção 3, não foi possível realizar testes experimentais com o robô completo. Para validação experimental, foram realizados testes de funcionamento dos módulos de atuação e da perna em si, a partir de comandos simples e de trajetórias pré-definidas.

#### 7.1

##### Módulo de atuação

Como componente essencial para o robô, o módulo de atuação precisa ter seu funcionamento verificado. Para isso, foi montada uma bancada de testes, apresentada na Figura 7.1, com o módulo conectado a um *ESC* (*Electronic Speed Controller*) comercial para controle de velocidade. O *ESC* é conectado a uma fonte de tensão e a um *Arduino Nano*, com um potenciômetro. Acoplado ao módulo, se coloca a ponta da perna.

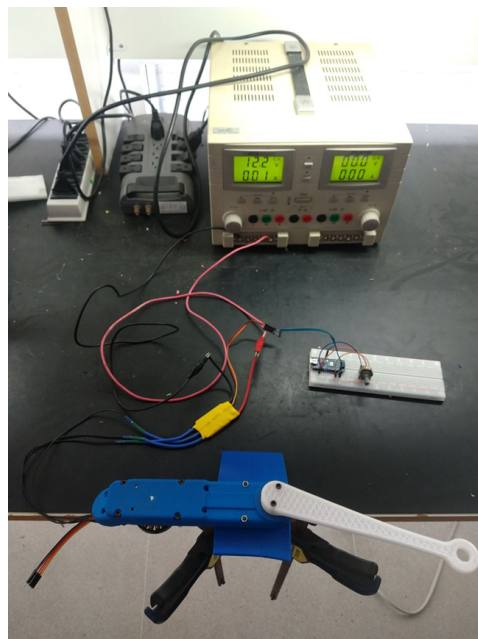


Figura 7.1: Bancada para teste do módulo de atuação da perna do robô Solo.

A partir de um *Arduino Nano*, o potenciômetro é utilizado para enviar comandos de velocidade para o *ESC* que, por sua vez, aciona o motor com uma velocidade proporcional ao comando do potenciômetro. Como primeiro teste, verificou-se o funcionamento do módulo para diferentes referências de velocidade. Em seguida, testou-se a sua reação a travamento, aplicando

manualmente uma força oposta ao movimento do motor, para verificar se existe deslizamento das correias.

No primeiro teste<sup>1</sup> (Figura 7.2), o módulo apresentou uma boa resposta a variação da velocidade pelo potenciômetro. Foi notado um ruído alto, o que pode ser indício de perda de energia e atrito em algum ponto da montagem.

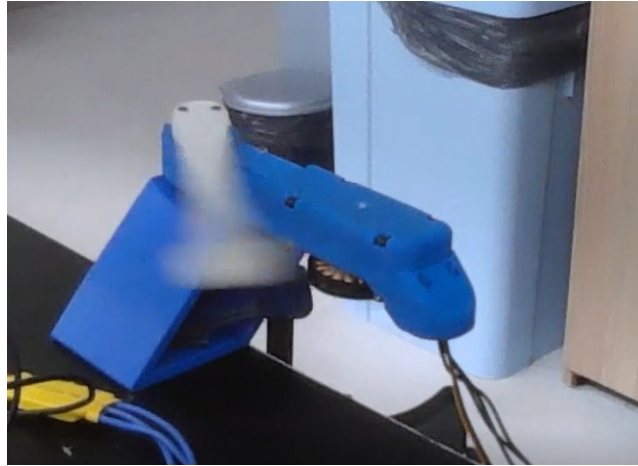


Figura 7.2: Primeiro teste com o módulo de atuação.

No segundo teste<sup>2</sup>, ao encontrar um bloqueio (Figura 7.3), o motor demonstrou ficar travado com o elo final da bancada, apresentando as mesmas oscilações, o que foi possível notar através da fixação de um pedaço de papel no rotor no motor. O mesmo ruído do teste anterior foi notado e para altas velocidades, existe deslizamento no sistema de transmissão.

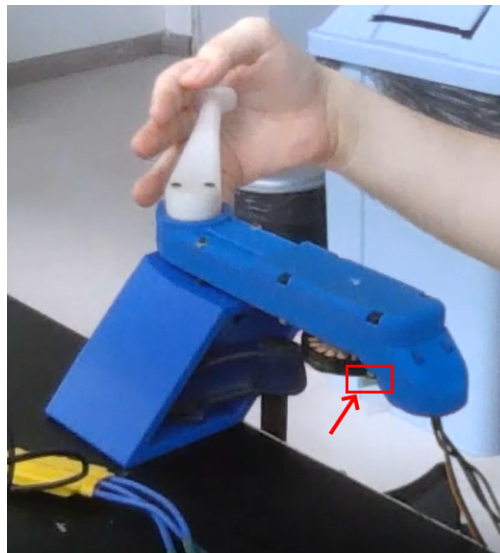


Figura 7.3: Segundo teste com o módulo de atuação.

<sup>1</sup>Teste de Transmissão [https://www.youtube.com/shorts/UKmH\\_p1naq0](https://www.youtube.com/shorts/UKmH_p1naq0)

<sup>2</sup>Teste de Travamento <https://www.youtube.com/shorts/7ftibnrLWYk>



## 7.2

### Testes de uma perna com o *Odrive*

Com o *Odrive* configurado e os ganhos dos controladores definidos, foram realizados testes de controle de movimento com uma perna, validando a montagem mecânica, eletrônica e o controle de uma perna. Todos os testes foram gravados e se encontram disponíveis para visualização no Youtube, os links estão nas notas de rodapé ao longo do texto.

#### 7.2.1

##### Matlab

Durante os testes com a perna, foram realizadas algumas trajetórias, tanto planejadas no espaço das juntas, como planejadas no espaço cartesiano, ou seja, para a extremidade da pata.

Para a pré-visualização da trajetória planejada e a verificação da sua viabilidade no robô real, foi utilizada a biblioteca *Robotics System Toolbox* do *Matlab*. Essa biblioteca permite definir um robô através do seu URDF (*Universal Robot Description File*), um arquivo no formato XML que descreve todos os elos do robô, as suas posições, suas características dinâmicas, e suas juntas. A biblioteca permite ainda a visualização de cada elo do robô a partir do seu desenho CAD no formato STL. Uma vez que o robô foi definido, é possível configurar qualquer posição arbitrária de juntas para visualização. A Figura 7.4 mostra o modelo completo do robô Solo no Matlab.

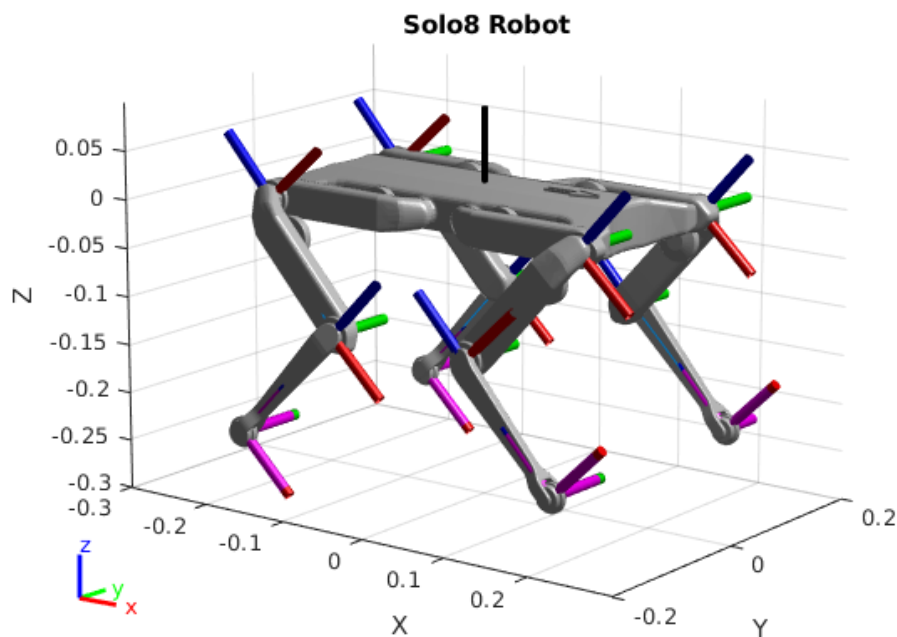


Figura 7.4: Modelo do robô Solo no Matlab.

Além disso, a biblioteca possui funções já implementadas de cinemática direta e inversa, que permitem a fácil visualização de uma trajetória proposta no espaço de tarefas, ou seja, do efector-final.

### 7.2.2

#### Comandos básicos

Os testes foram iniciados com comandos simples usando a própria interface do *Odrive*, utilizando a bancada da Figura 7.5. O controlador *Odrive* utilizado para este teste possui dois canais, o que permitiria controlar dois motores na mesma placa. No entanto, devido a um mau funcionamento apresentado no Canal 0 da placa, foi necessário o uso de dois controladores *Odrives* para o controle de uma perna completa, cada um ligado a um módulo de atuação. Ambas as placas estão ligadas a um computador, responsável pelo envio dos comandos.



Figura 7.5: Bancada para teste de comandos.

O primeiro teste a ser feito em cada elo atuado é o de calibração dos motores<sup>3</sup>. O procedimento de calibração é realizado pelo *Odrive* e consiste em fazer o motor realizar uma pequena rotação no sentido horário e depois retornar ao ponto original.

<sup>3</sup>Calibração da Perna <https://www.youtube.com/watch?v=7d9CLBnxFcc>

Em seguida, são realizados comandos com diferentes posições de referência para ambos os módulos, encerrando com o comando de retorno a posição inicial<sup>4</sup>. Durante o teste, foi possível verificar que o controlador de posição conseguiu levar o motor para a posição desejada sem *overshoot*, e os ganhos do controlador estavam satisfatórios.

### 7.2.3 Trajetória Simples

Após a validação do controle de posição da perna com a placa *Odrive*, o próximo passo é criar um código com comandos que façam a perna seguir uma trajetória. Para minimizar a interferência e melhorar a sincronia entre os módulos, os dois controladores são conectados a mesma fonte de tensão, como mostra a Figura 7.6.

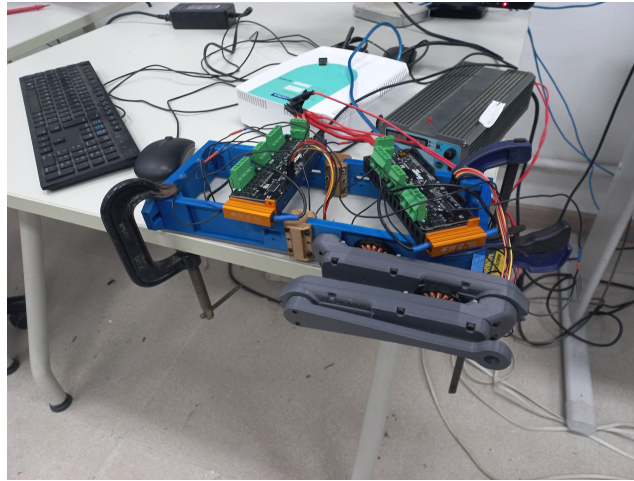


Figura 7.6: Bancada para teste de trajetórias.

Para este teste, a trajetória da perna foi definida no espaço das juntas, ou seja, diretamente em posições angulares para os motores. A posição inicial da perna é ela completamente recuada na horizontal, conforme Figura 7.7.



Figura 7.7: Posição inicial na horizontal.

O movimento é feito com um incremento linear, partindo de 0° até que o primeiro módulo e o segundo módulo tenham atingido respectivamente uma

---

<sup>4</sup>Teste de Controle de Posição <https://www.youtube.com/watch?v=6vXwIEhn3pM>

rotação de  $45^\circ$  e  $90^\circ$ , efetuando em seguida o retorno para a posição zero também linearmente. A trajetória proposta foi testada a partir da visualização no Matlab<sup>5</sup>, como mostra a Figura 7.8. A linha em azul mostra a trajetória executada pela pata do robô, computada a partir da cinemática direta da perna.

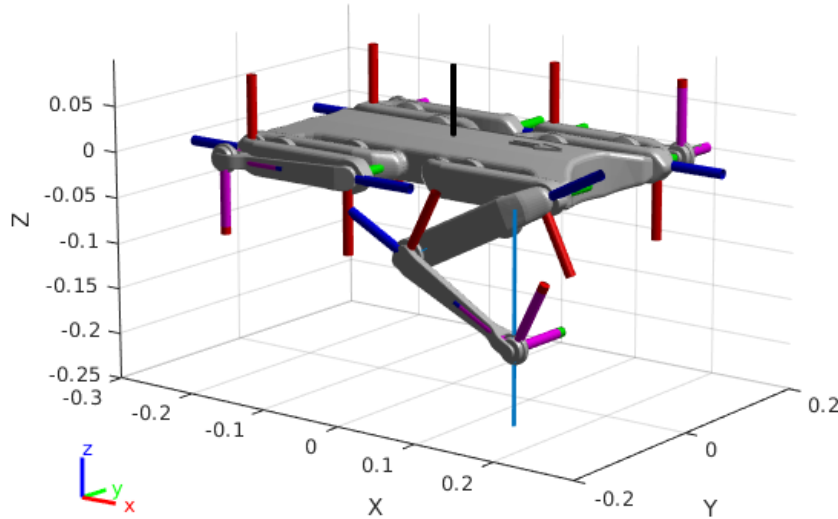


Figura 7.8: Teste de visualização da trajetória simples no Matlab.

Para o teste com o robô real, foi utilizado um código em *Python*, mas visando uma futura implementação em *C++* para maior eficiência. O *script* começa realizando a calibração dos motores e depois envia comandos para atualização de posição desejada com intervalos de 0,5 segundos, também gravando a leitura do *encoder* a cada etapa. A Figura 7.9 mostra dois momentos da realização deste teste.

A partir dos dados gravados durante a trajetória, é possível realizar uma análise gráfica dos resultados. A Figura 7.10 mostra a comparação entre as posições desejadas das juntas e as posições medidas pelo *Odrive*. Nota-se que a trajetória feita pelos módulos possuem uma defasagem em relação ao desejado, mas explicável pelo espaço de tempo que leva para o código em *Python* enviar a posição desejada para o *Odrive* e a baixa frequência de envio das posições. Além disso, existe uma imperfeição gerada por um breve período transiente no início, quando começa o movimento, e na metade da operação, quando ocorre inversão de direção, algo melhor observável na Figura 7.11.

Usando a cinemática direta da perna do robô, é possível comparar também a trajetória realizada pela pata do robô. A Figura 7.12 mostra a comparação da trajetória real (em azul) com a desejada (em laranja), é possível observar que a trajetória executada pela pata está muito próxima

<sup>5</sup>Trajetoária simples no Matlab <https://youtu.be/5bmWSyn-1NY>



7.9(a): Descendo.



7.9(b): Ponto mais baixo.

Figura 7.9: Fotos do Vídeo da trajetória simples.

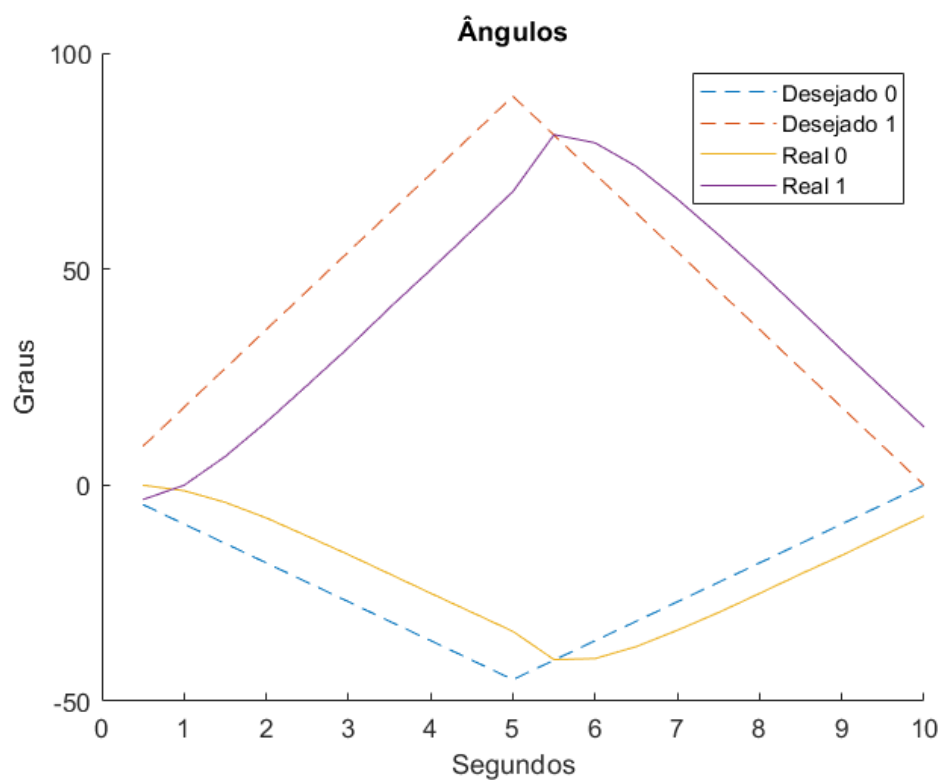


Figura 7.10: Variação de ângulos na trajetória simples.

da trajetória desejada, mas apresenta um pequeno deslocamento vertical de aproximadamente 10 mm. Na direção do eixo X, as diferenças horizontais vistas na Figura 7.13 podem ser consideradas ruídos, já que mostram pequenos erros



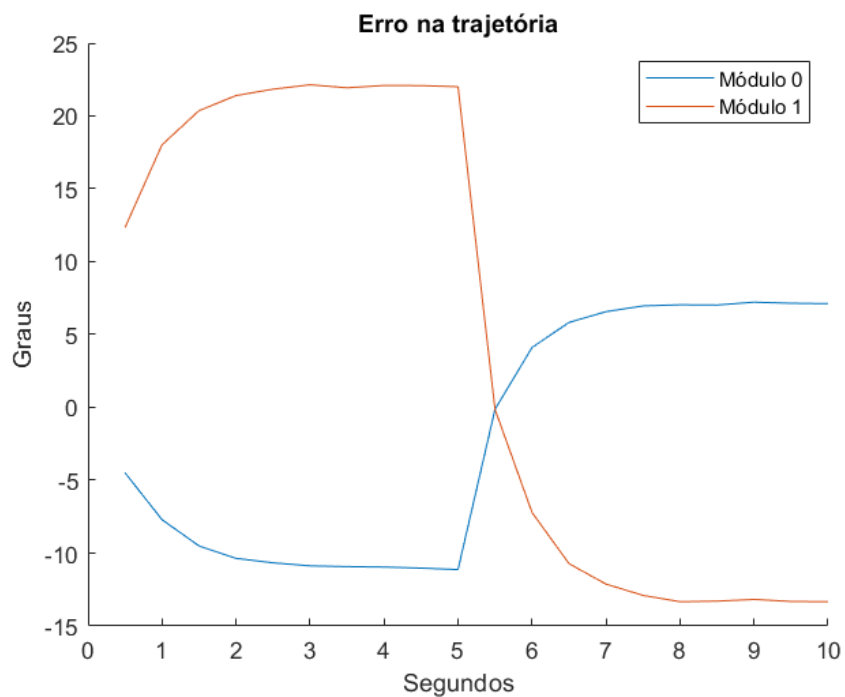


Figura 7.11: Gráfico do erro na trajetória das juntas.

de menos de 1,0 mm.

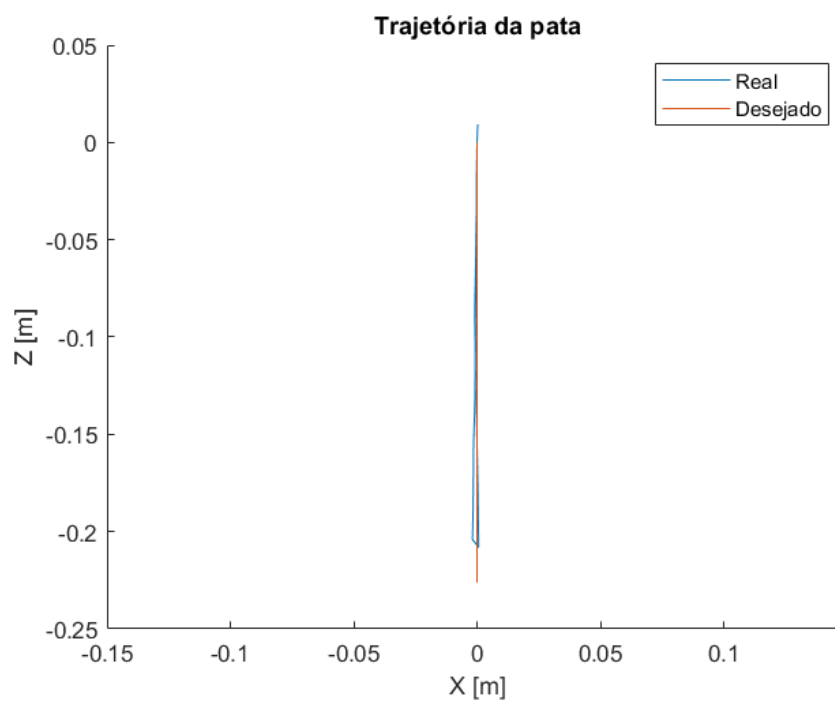


Figura 7.12: Movimentação da pata durante a trajetória simples.

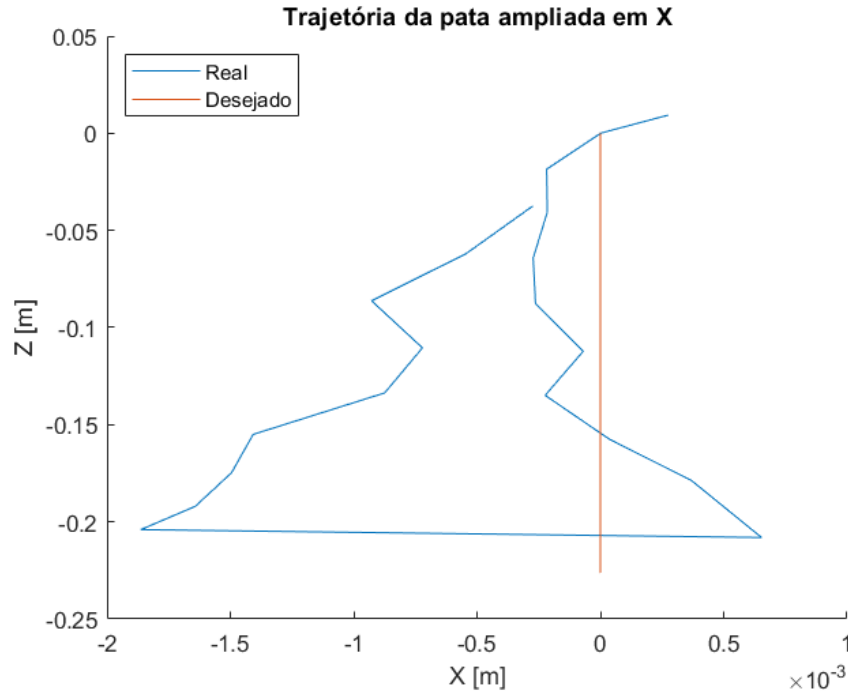


Figura 7.13: Gráfico ampliado de trajetória simples.

#### 7.2.4

##### Trajétória vertical no espaço cartesiano

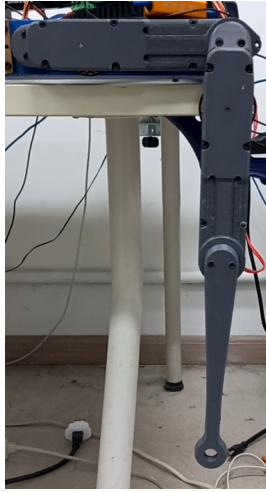
A partir do código utilizado para a trajetória simples, foi implementado um código de cinemática inversa para testar trajetórias planejadas para a pata do robô no espaço cartesiano. Este teste é importante também para verificar se há queda de desempenho quando é preciso computar a cinemática inversa a cada iteração para gerar a posição desejada das juntas. Para ficar compatível com o modelo cinemático inverso, a calibragem da perna foi realizada com ela esticada para baixo, conforme a Figura 7.14(a). Para reduzir o custo computacional, a cinemática inversa foi implementada de forma analítica, não sendo necessário o uso de um *solver* numérico.

A primeira trajetória consiste em um movimento vertical da pata, representada como um sinal senoidal de meio-ciclo com amplitude de 0,1 m no eixo Z, partindo da posição em que a primeira junta está com  $-45^\circ$  e segunda junta com  $90^\circ$ <sup>6</sup>, como mostra a Figura 7.14(b). Esta movimentação é muito similar a trajetória da perna durante uma caminhada em baixa velocidade. Durante o experimento<sup>7</sup> (Figura 7.15), foram gravadas as posições reais e desejadas das juntas por 800 s.

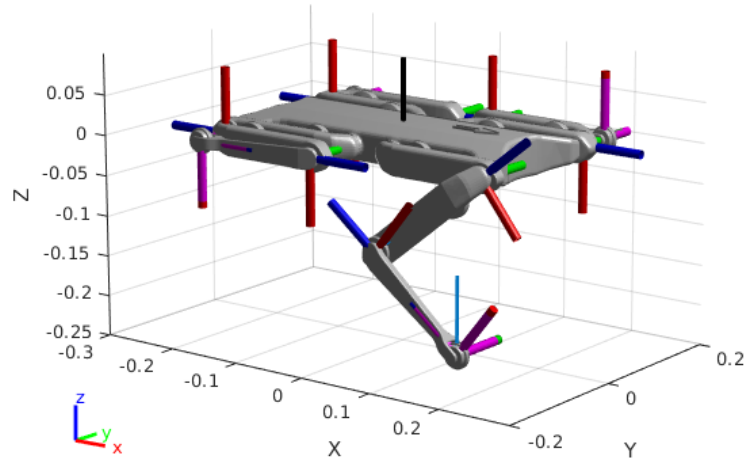
A Figura 7.16 mostra a comparação entre as posições desejadas das juntas e as posições medidas pelo *Odrive*. Nota-se que uma defasagem mais explícita

<sup>6</sup>Trajétória em Z no *Matlab* [https://youtu.be/e3\\_x5TbmqTM](https://youtu.be/e3_x5TbmqTM)

<sup>7</sup>Teste de trajetória vertical <https://www.youtube.com/shorts/r5MyAMVaiRQ>



7.14(a): Posição inicial na vertical.

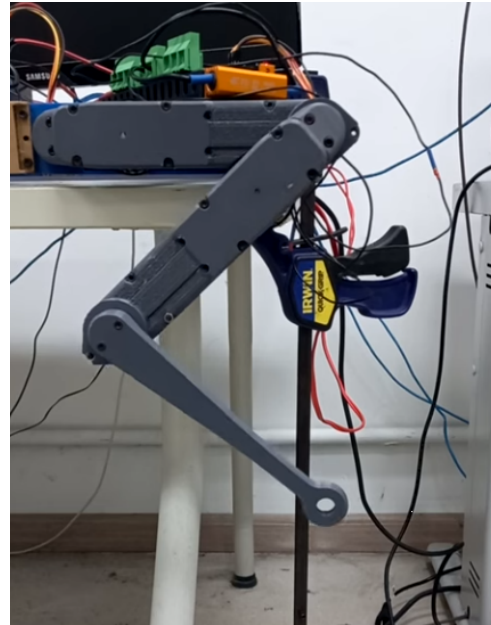


7.14(b): Visualização da trajetória vertical no Matlab.

Figura 7.14: Calibração e visualização da trajetória vertical.



7.15(a): Subindo.



7.15(b): Descendo.

Figura 7.15: Fotos do Vídeo da trajetória vertical.

do que em relação vista na trajetória simples, mas que pode ser explicada pelo tempo gasto no cálculo da cinemática inversa e pela latência de comunicação entre o computador e o *Odrive*. Além disso, o período que deveria ser de pausa acaba sendo usado para o módulo se ajustar, o que indica necessidade de ajuste do ganho proporcional, para uma resposta mais rápida.

Na Figura 7.17, é possível perceber que o erro de posição angular para essa trajetória foi considerável. No módulo 0, o erro varia entre  $-15^\circ$  até  $10^\circ$ , enquanto o módulo 1 tem uma variação ainda maior, entre de  $-25^\circ$  até  $35^\circ$ .



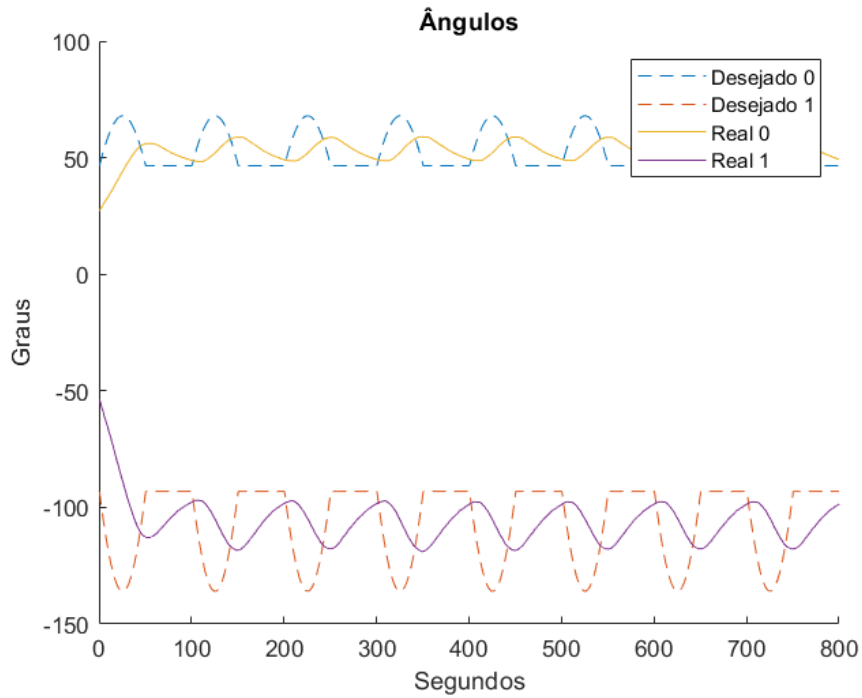


Figura 7.16: Variação de ângulos na trajetória vertical.

Esses erros podem indicar a necessidade de reajuste dos ganhos do controlador.

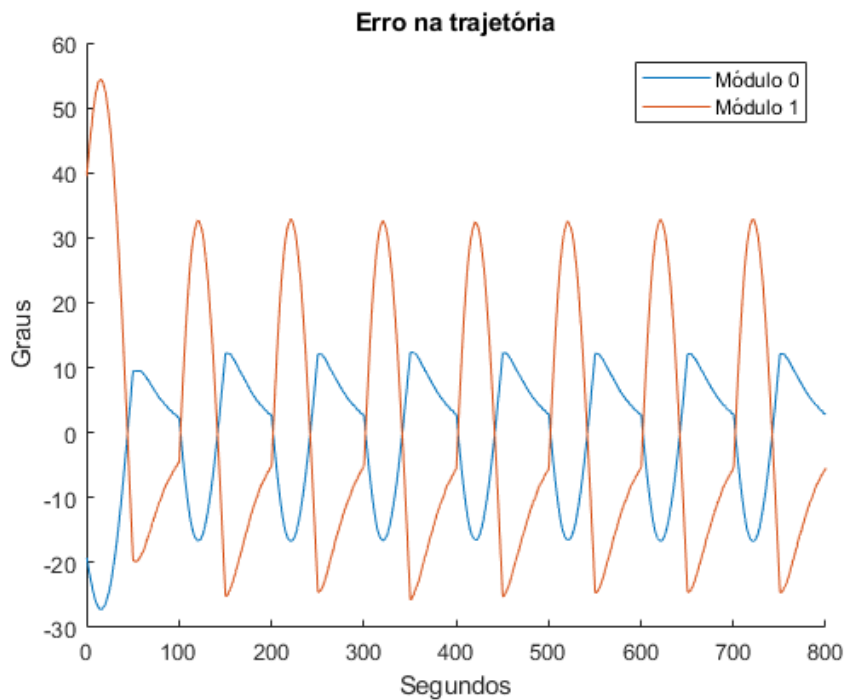


Figura 7.17: Gráfico do erro na trajetória vertical.

Com aplicação da cinemática direta nos dados medidas das juntas, gera-se a Figura 7.18, que mostra a trajetória da pata. Por causa dos erros na trajetória das juntas, nota-se um movimento mais para baixo que o esperado

no eixo Z. Uma forma de melhorar esse controle é atuar em torque nas juntas, ao invés de posição, pois poderia ser feita a compensação da gravidade e os ganhos do controlador poderiam ser menores. Não houve erro significativo no movimento ao longo do eixo X, como mostra a Figura 7.19.

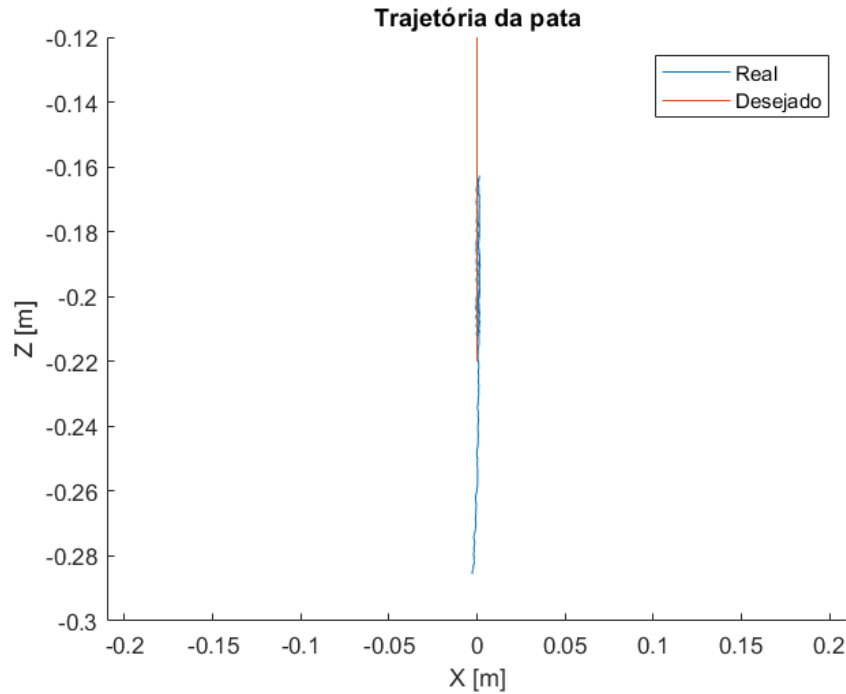


Figura 7.18: Gráfico da trajetória vertical.

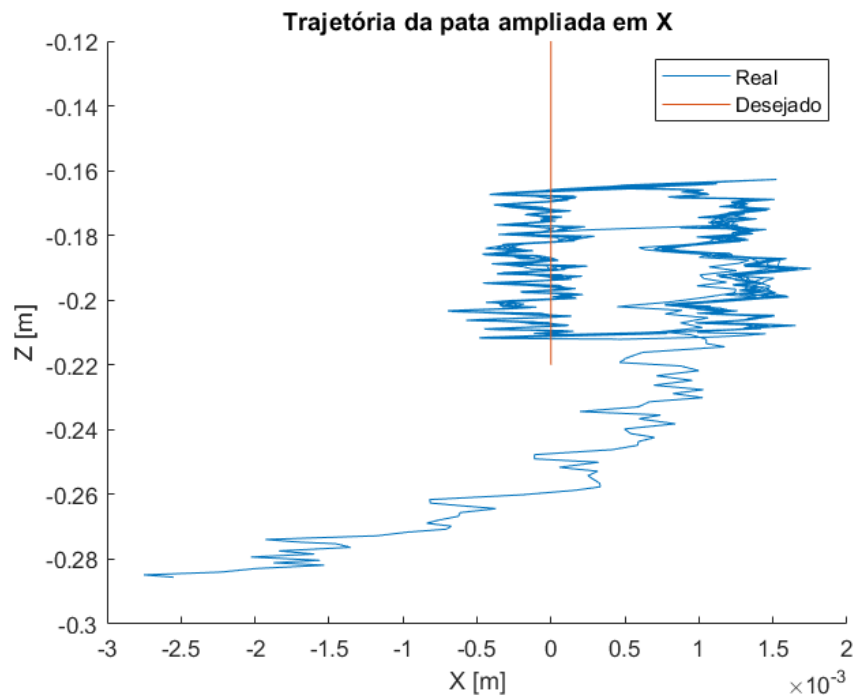


Figura 7.19: Gráfico ampliado da trajetória vertical.

Nota-se que, apesar dessa trajetória ter a mesma direção que a da trajetória simples, acabou tendo um erro consideravelmente maior que a anterior. Esta diferença pode ser explicada pelo cálculo de cinemática inversa que teve que ser feito, e pelo fato da computação dos ângulos ser realizada numa frequência muito maior, 100 Hz, o que deixa mais clara a influência do tempo de resposta do *Odrive*.

### 7.2.5

#### Trajétória simulando um passo no espaço cartesiano

Essa trajetória simula o movimento de um passo para frente, mas como o corpo do robô está parado, a trajetória inclui também o movimento da pata no eixo X. A trajetória é composta por um arco para frente, com um movimento linear horizontal de volta ao ponto original<sup>8</sup>, como mostra a Figura 7.20. A calibração foi realizada a partir da posição vertical da pata (Figura 7.14(a)).

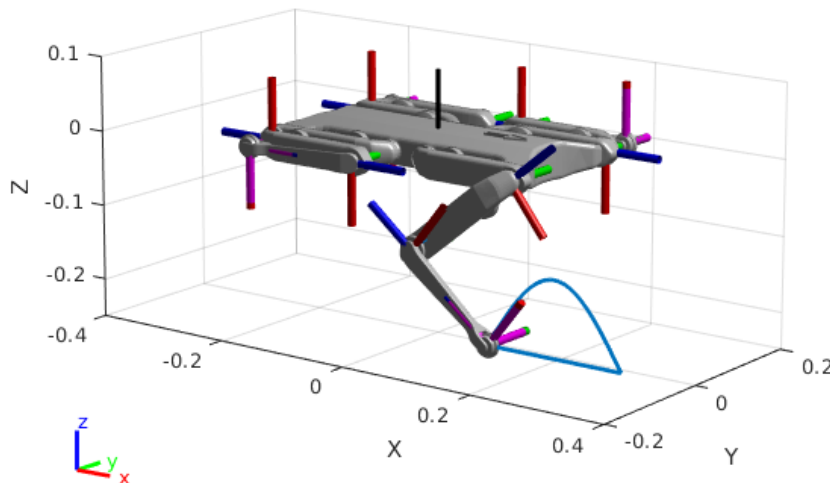


Figura 7.20: Trajetória simulando um passo para o a base do robô parada.

Esse teste<sup>9</sup> também teve duração de 800 segundos, com frequência de comunicação em 100 Hz, onde a Figura 7.21 mostra os movimentos do avanço 7.21(a) e recuo 7.21(b).

A Figura 7.10 mostra a comparação entre as posições desejadas das juntas e as posições medidas pelo *Odrive* para essa trajetória. Nota-se que o problema de defasagem também se repete e há uma maior discrepância entre a trajetória desejada e real em relação aos outros experimentos, por ser mais complexo e exigir locomoção nos dois eixos. Na Figura 7.22, observa-se que o erro varia de  $-30^\circ$  até  $40^\circ$  no módulo 0 e de  $-60^\circ$  até  $35^\circ$  no módulo 1.

<sup>8</sup>Trajétória de passo no *Matlab* [https://youtu.be/qZIfm0JKd\\_c](https://youtu.be/qZIfm0JKd_c)

<sup>9</sup>Teste de trajetória de passo [https://www.youtube.com/shorts/4l\\_5fzjiRNw](https://www.youtube.com/shorts/4l_5fzjiRNw)



7.21(a): Avanço.



7.21(b): Recuo.

Figura 7.21: Fotos do Vídeo da trajetória de passo.

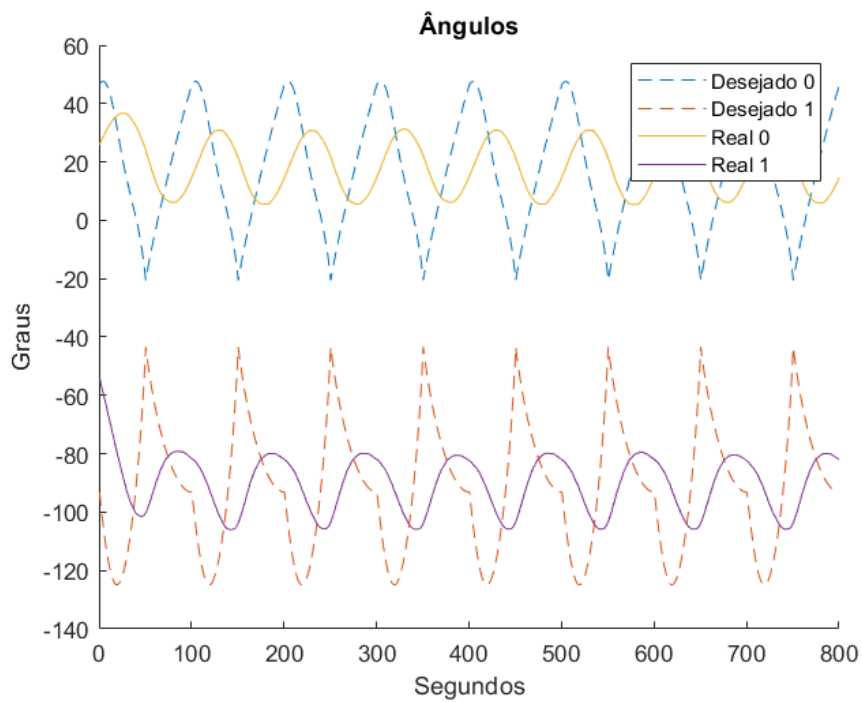


Figura 7.22: Variação de ângulos na trajetória de passo

A Figura 7.24 mostra a comparação entre o movimento real e o desejado para a pata do robô. Como foi dito anteriormente, nota-se uma grande disparidade entre o real e desejado, tanto no eixo Z quanto no X, de aproximadamente 5 cm. A primeira parte do movimento, um semi-ciclo de senoide, acaba sendo realizado com uma amplitude bastante inferior e com seu pico antecedendo

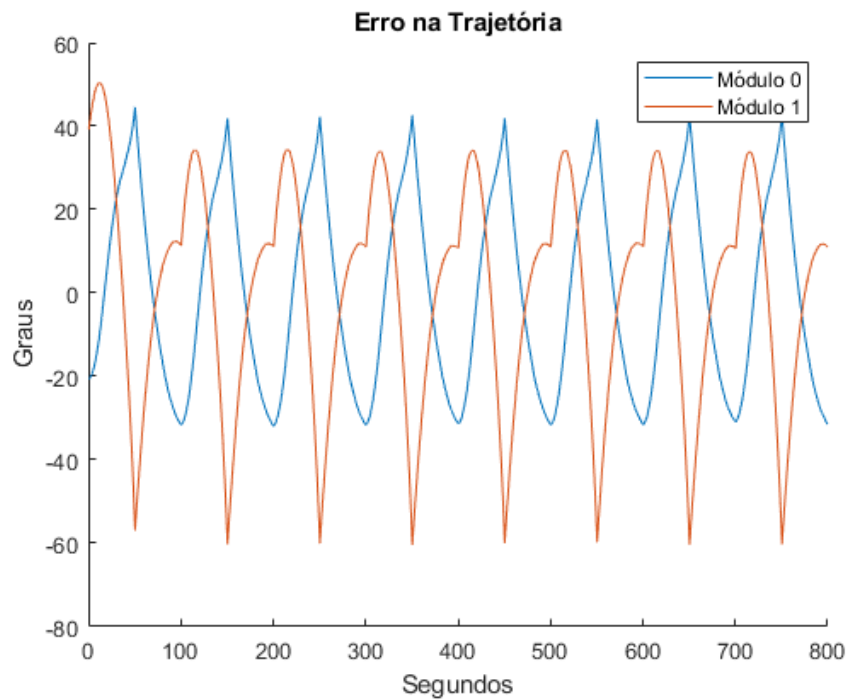


Figura 7.23: Gráfico do erro na trajetória de passo.

o desejado, enquanto a parte horizontal do movimento, que deveria ser reto, virou uma curva que tangencia o movimento desejado.

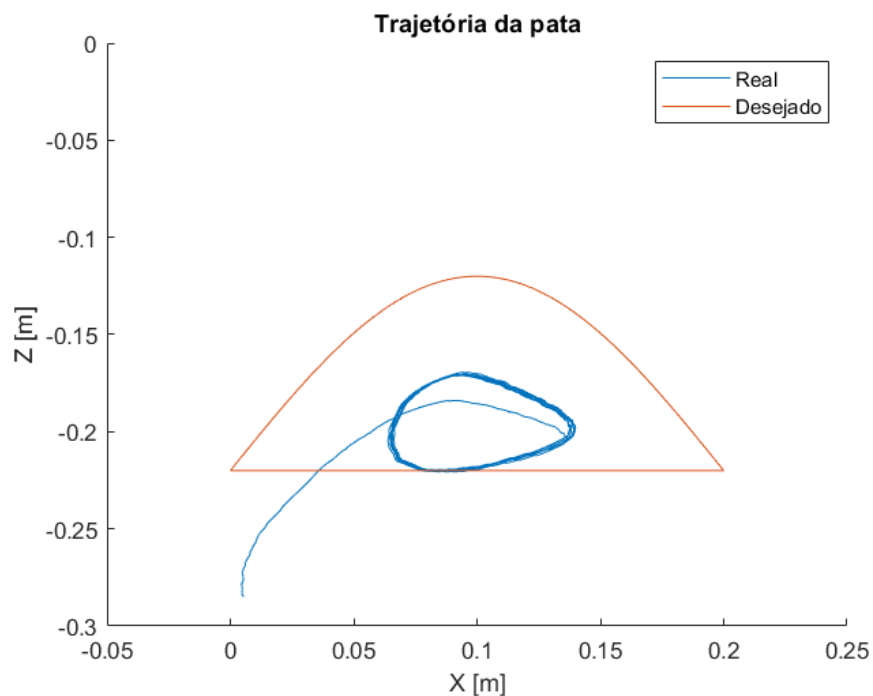


Figura 7.24: Gráfico da trajetória do passo executada pela perna do robô.

As possíveis razões para essa diferença são mesmas analisadas para a trajetória vertical.

## 8

### Conclusão

Este trabalho mostrou a construção e montagem do robô quadrúpede *Solo*, conseguindo usar alternativas de componentes mais baratos nacionais, além de ter sua estrutura modular e perna validada. Apesar do problema na eletrônica original, que deixa para ser um desafio a ser resolvido, houve sucesso na implementação do *Odrive*, deixando em aberto uma oportunidade de adaptação do sistema de controle para usar esse controlador alternativo. Durante os testes experimentais, foi verificada a necessidade de ajustes no sistema de controle, usando a linguagem *C++* para melhor eficiência e um ajuste melhor dos ganhos do controlador PID. Os próximos passos consistem em testes de controle de trajetória da pata usando um controle de torque, para compensar efeitos dinâmicos, e o teste com o robô completo realizando a trajetória de caminhada estaticamente estável testada em simulação.

## Referências Bibliográficas

- [1] Roland Siegwart, Illah R. Nourbakhsh, and Davide Scaramuzza. *Introduction to Autonomous Mobile Robots*. The MIT Press, 2nd edition, 2011.
- [2] MARC RAIBERT, KEVIN BLANKESPOOR, GABRIEL NELSON, ROB PLAYTER. Bigdog, the rough-terrain quadruped robot. *IFAC Proceedings Volumes*, 41:10822–10825, 2008.
- [3] Hae-Won Park, Patrick M. Wensing, and Sangbae Kim. Jumping over obstacles with mit cheetah 2. *Robotics and Autonomous Systems*, 136:103703, 2021.
- [4] Marco Hutter, Christian Gehring, Dominic Jud, Andreas Lauber, C. Dario Bellicoso, Vassilios Tsounis, Jemin Hwangbo, Karen Bodie, Peter Fankhauser, Michael Bloesch, Remo Diethelm, Samuel Bachmann, Amir Melzer, and Mark Hoepflinger. Anymal - a highly mobile and dynamic quadrupedal robot. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 38–44, 2016.
- [5] ANYBotics. Anymal x — the world's first ex-proof legged robot, 2021. Available at: <https://www.anybotics.com/anymal-x-the-worlds-first-ex-proof-legged-robot/>.
- [6] Xiaomi. Explore the possibilities with xiaomi cyberdog | xiaomi academy, 2021. Available at: [https://www.youtube.com/watch?v=X\\_uAqYVK6xY](https://www.youtube.com/watch?v=X_uAqYVK6xY).
- [7] Marco Hutter, Roland Siegwart, Thomas Stastny, Konrad Rudin, and Michael Blösch. Robot dynamics lecture notes, 2017. Available at: [https://ethz.ch/content/dam/ethz/special-interest/mavt/robotics-n-intelligent-systems/rsl-dam/documents/RobotDynamics2017/RD\\_HS2017script.pdf](https://ethz.ch/content/dam/ethz/special-interest/mavt/robotics-n-intelligent-systems/rsl-dam/documents/RobotDynamics2017/RD_HS2017script.pdf). Accessed: April 2021.
- [8] Qing Shi, Junhui Gao, Shengjie Wang, Xiaolong Quan, Guanglu Jia, Qiang Huang, and Toshio Fukuda. Development of a small-sized quadruped robotic rat capable of multimodal motions. *IEEE Transactions on Robotics*, pages 1–17, 2022.
- [9] Lei Jiang, Bo Su, Peng Xu, Qindan Deng, Lindong Mu, and Yunfeng Jiang. Research on cheetah bionic quadruped robot technology. In *2020 3rd International Conference on Unmanned Systems (ICUS)*, pages 1078–1082, Nov 2020.

- [10] V. S. Medeiros, E. Jelavic, M. Bjelonic, R. Siegwart, M. A. Meggiolaro, and M. Hutter. Trajectory optimization for wheeled-legged quadrupedal robots driving in challenging terrain. *IEEE Robotics and Automation Letters*, 5(3):4172–4179, 2020.
- [11] Marko Bjelonic, Prajish K. Sankar, C. Dario Bellicoso, Heike Vallery, and Marco Hutter. Rolling in the deep - hybrid locomotion for wheeled-legged robots using online trajectory optimization. *CoRR*, abs/1909.07193, 2019.
- [12] Marco Hutter, Remo Diethelm, Samuel Bachmann, Péter Fankhauser, Christian Gehring, Vassilios Tsounis, Andreas Lauber, Fabian Guenther, Marko Bjelonic, Linus Isler, Hendrik Kolvenbach, Konrad Meyer, and Mark Hoepflinger. Towards a generic solution for inspection of industrial sites. In Marco Hutter and Roland Siegwart, editors, *Field and Service Robotics*, pages 575–589. Springer International Publishing, Cham, 2018.
- [13] C. Gehring, P. Fankhauser, L. Isler, R. Diethelm, S. Bachmann, M. Potz, L. Gerstenberg, and M. Hutter. Anymal in the field: Solving industrial inspection of an offshore hvdc platform with a quadrupedal robot. In Genya Ishigami and Kazuya Yoshida, editors, *Field and Service Robotics*, pages 247–260. Springer Singapore, Singapore, 2021.
- [14] Joonyoung Kim, Taewoong Kang, Dongwoon Song, and Seung-Joon Yi. Design and control of a open-source, low cost, 3d printed dynamic quadruped robot. *Applied Sciences*, 11:3762, 04 2021.
- [15] Facundo García-Cárdenas, Nelson Soberón, Oscar E. Ramos, and Ruth Canahuire. Charlotte: Low-cost open-source semi-autonomous quadruped robot. In *2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 281–286, April 2020.
- [16] F. Grimminger, A. Meduri, M. Khadiv, J. Viereck, M. Wüthrich, M. Naveau, V. Berenz, S. Heim, F. Widmaier, T. Flayols, J. Fiene, A. Badri-Spröwitz, and L. Righetti. An open torque-controlled modular robot architecture for legged locomotion research. *IEEE Robotics and Automation Letters*, 5(2):3650–3657, 2020.
- [17] Karen Heller. Spot is the \$74,500 robot dog of our dystopian dreams, 2021. Available at: [https://www.washingtonpost.com/lifestyle/style/spot-dog-robot-boston-dynamics/2021/08/06/81b2b780-f475-11eb-9068-bf463c8c74de\\_story.html/](https://www.washingtonpost.com/lifestyle/style/spot-dog-robot-boston-dynamics/2021/08/06/81b2b780-f475-11eb-9068-bf463c8c74de_story.html/).



- [18] Marco Antonio Meggiolaro. Controle de sistemas robóticos, parte 7 - dinâmica, 2020.
- [19] Peter Corke. Rigid body dynamics, 2017. Available at: <https://robotacademy.net.au/masterclass/rigid-body-dynamics/?lesson=406/>.