

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA MECÂNICA

CÍCERO LUIZ ALVES ZANETTE

**DESENVOLVIMENTO DE UM AMBIENTE DE SIMULAÇÃO PARA
UM ROBÔ BÍPEDE CONTROLADO POR CONTRAPESO COM
MOVIMENTO LINEAR**

SÃO CARLOS
2021

CÍCERO LUIZ ALVES ZANETTE

**DESENVOLVIMENTO DE UM AMBIENTE DE SIMULAÇÃO PARA
UM ROBÔ BÍPEDE CONTROLADO POR CONTRAPESO COM
MOVIMENTO LINEAR**

Trabalho de Conclusão de Curso
apresentado ao Departamento de
Engenharia Mecânica da Universidade
Federal de São Carlos, para obtenção
do título de Bacharel em Engenharia
Mecânica.

Orientador: Prof. Dr. João Vitor de
Carvalho Fontes

SÃO CARLOS
2021



FUNDAÇÃO UNIVERSIDADE FEDERAL DE SÃO CARLOS

COORDENAÇÃO DO CURSO DE ENGENHARIA MECÂNICA - CCEMec/CCET

Rod. Washington Luís km 235 - SP-310, s/n - Bairro Monjolinho, São Carlos/SP, CEP 13565-905

Telefone: (16) 33519703 - <http://www.ufscar.br>

DP-TCC-FA nº 29/2021/CCEMec/CCET

Graduação: Defesa Pública de Trabalho de Conclusão de Curso

Folha Aprovação (GDP-TCC-FA)

FOLHA DE APROVAÇÃO

CÍCERO LUIZ ALVES ZANETTE

DESENVOLVIMENTO DE UM AMBIENTE DE SIMULAÇÃO PARA UM ROBÔ BÍPEDE CONTROLADO POR CONTRAPESO COM MOVIMENTO LINEAR

Trabalho de Conclusão de Curso

Universidade Federal de São Carlos – Campus São Carlos

São Carlos, 02 de dezembro de 2021

ASSINATURAS E CIÊNCIAS

Cargo/Função	Nome Completo
Orientador	João Vitor de Carvalho Fontes
Membro da Banca 1	Luis Antonio Oliveira Araújo
Membro da Banca 2	Vitor Ramos Franco



Documento assinado eletronicamente por **Vitor Ramos Franco, Docente**, em 02/12/2021, às 11:48, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Joao Vitor de Carvalho Fontes, Docente**, em 02/12/2021, às 11:49, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Luis Antonio Oliveira Araujo, Docente**, em 02/12/2021, às 11:50, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site <https://sei.ufscar.br/autenticacao>, informando o código verificador **0539203** e o código CRC **107585AA**.

AGRADECIMENTOS

Gostaria de agradecer aos meus pais que fizeram o possível e o impossível para que eu pudesse chegar até aqui e aos meus amigos que me ajudaram a manter a sanidade durante o período de isolamento no qual esse trabalho foi escrito. Agradeço também ao professor Alexandre Tácito Malavolta por ter me oferecido a primeira oportunidade de ingressar no ambiente de pesquisas e me apresentar à área de simulações, ao professor Mariano Eduardo Moreno por ter me apresentado à programação por blocos já em um contexto de uma aplicação real, e ao meu orientador, professor João Vitor de Carvalho Fontes, que foi extremamente calmo durante minhas intermináveis dúvidas e sempre buscou ter certeza de que eu tinha realmente entendido a explicação, recebeu minhas mensagens altas horas da noite com algum avanço no programa e comemorou comigo cada parte concluída, isso tudo sem nunca termos nos encontrado pessoalmente.

RESUMO

Robôs bípedes são máquinas capazes de executar algumas tarefas consideravelmente complicadas para outros robôs, como subir escadas e ultrapassar pequenos obstáculos, isso ocupando uma área de suporte relativamente pequena. Geralmente, esse tipo de máquina é controlada utilizando o chamado movimento pendular, no qual o robô balança de um lado para o outro para manter seu centro de massa dentro da área de apoio. Isso impõe que o planejamento da trajetória do robô leve em conta também o seu equilíbrio o que torna o projeto mais complexo. No presente trabalho, buscou-se construir um ambiente de simulação em SIMULINK para um robô cujo equilíbrio fosse feito separadamente da passada através de um contrapeso, tornando os dois movimentos independentes. O ambiente desenvolvido foi considerado bem-representativo através de uma simulação quase-estática, porém a elaboração do controle mostrou-se mais complexa do que o planejado e não foi possível implementá-la. Ainda assim o programa permitirá que futuros trabalhos nessa área possam ser desenvolvidos de maneira mais célere devido à não necessidade de se construir um protótipo.

Palavras-chave: Robô bípede. Simulação. SIMULINK.

ABSTRACT

Bipedal robots are machines capable of performing some tasks that are considerably complicated for other robots, such as climbing stairs and overcoming small obstacles, while occupying a relatively small support area. Generally, this type of machine is controlled using a so-called pendulum movement, in which the robot swings from side to side to maintain its center of mass within the support area. This requires that the robot trajectory planning also take into account its balance, which makes the project more complex. In the present work, we sought to build a simulation environment in SIMULINK for a robot whose balance was done separately from the stride through a counterweight, making the two movements independent. The developed environment was considered well-representative through a quasi-static simulation, but the development of the control proved to be more complex than planned and it was not possible to implement it, even so, the program will allow that future works in this area can be developed more quickly due to the no need to build a prototype.

Keywords: Bipedal robot. Simulation. SIMULINK.

LISTA DE FIGURAS

Figura 1 – Exemplo de um robô bípede	23
Figura 2 – Exemplo de um <i>Passive-Dynamic Walker</i>	24
Figura 3 – Robô bípede controlado por propulsores	25
Figura 4 – Exemplo de robô bípede com 8 GDL's	27
Figura 5 – Robô desenvolvido por Oliveira e Araujo (2020)	28
Figura 6 – Visualização da geometria do robô	34
Figura 7 – DCL do robô mono-apoiado	36
Figura 8 – DCL do sistema guia e contrapeso	37
Figura 9 – Visualização de α e y	39
Figura 10 – Visualização do robô no ambiente de simulação	41
Figura 11 – Variação do ângulo do pé esquerdo com o tempo	43
Figura 12 – Fases da movimentação	44
Figura A-1–Código geral da simulação	49
Figura A-2–Código do subsistema "Perna esquerda"	50
Figura A-3–Código do subsistema "Perna direita"	51
Figura A-4–Código do subsistema "Guia e contrapeso"	52
Figura A-5–Código geral da simulação de controle	53
Figura A-6–Código do subsistema "Controle"	54

LISTA DE QUADROS

Quadro 1 – Blocos utilizados na simulação	31
---	----

LISTA DE TABELAS

Tabela 1 – Condições de contorno de posição para cada junta em cada fase, em rad	35
Tabela 2 – Massas relativas às forças presentes no DCL da Figura 7, em kg	35
Tabela 3 – Massas relativas às forças presentes no DCL da Figura 8, em kg	37
Tabela 4 – Condições de contorno de posição do contrapeso em cada fase, em mm	39
Tabela 5 – Coeficientes da Equação 1 para $q_{q.d}$	42
Tabela 6 – Coeficientes da Equação 1 para $q_{q.e}$	42
Tabela 7 – Coeficientes da Equação 1 para $p_{contrapeso}$	42

LISTA DE ABREVIATURAS E SIGLAS

GDL	Grau de liberdade
DCL	Diagrama de corpo livre
CG	Centro de gravidade
CAD	Computer-aided design

LISTA DE SÍMBOLOS

q	Posição angular
t	Tempo
$a_0 \dots a_5$	Coefficientes polinomiais
$q_{q.d}$	Posição angular do quadril direito
$q_{c.d}$	Posição angular da coxa direita
$q_{q.e}$	Posição angular do quadril esquerdo
$q_{c.e}$	Posição angular da coxa esquerda
F	Força
F_P	Força exercida pela perna levantada
F_{CE}	Força exercida pela parte esquerda da cintura
F_{CD}	Força exercida pela parte direita da cintura
M	Momento
M_S	Momento exercido pelos servomotores acoplados à perna de apoio
m	Massa
m_P	Massa da perna levantada
m_{CE}	Massa da parte esquerda da cintura
m_{CD}	Massa da parte direita da cintura
m_S	Massa de dois servos
g	Aceleração da gravidade
F_{GE}	Força exercida pela parte esquerda da guia
F_{GD}	Força exercida pela parte direita da guia
F_{CCP}	Força exercida pelo conjunto do contrapeso
m_{GE}	Massa da parte esquerda da guia
m_{GD}	Massa da parte direita da guia

m_c	Massa do carrinho da guia
M_{GE}	Momento exercido pela parte esquerda da guia
M_{GD}	Momento exercido pela parte direita da guia
M_{CCP}	Momento exercido pelo conjunto do contrapeso
m_{cp}	Massa do contrapeso
p	Posição linear
$p_{contrapeso}$	Posição linear do contrapeso

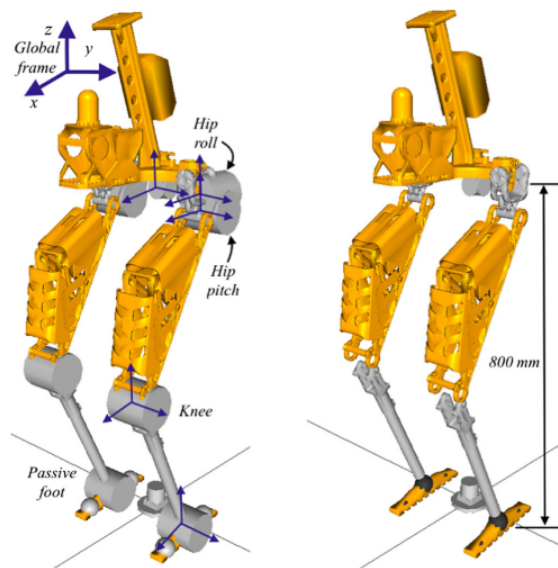
SUMÁRIO

1	INTRODUÇÃO	23
1.1	OBJETIVOS	25
2	FUNDAMENTAÇÃO TEÓRICA	27
2.1	GDL's E EQUILÍBRIO	27
2.2	MÉTODO DE CONTROLE	28
2.3	PLANEJAMENTO DA TRAJETÓRIA	29
3	DESENVOLVIMENTO	31
3.1	REVISÃO DO PROJETO MECÂNICO	31
3.2	MONTAGEM DO CÓDIGO	31
3.3	MOVIMENTAÇÃO QUASE-ESTÁTICA	35
3.4	SIMULAÇÃO COMPLETA	39
4	RESULTADOS	41
4.1	AMBIENTE DE SIMULAÇÃO	41
4.2	PLANEJAMENTO DE TRAJETÓRIA	41
4.3	MOVIMENTAÇÃO QUASE-ESTÁTICA	42
4.4	CONTROLE	43
5	CONCLUSÃO	45
5.1	CONTRIBUIÇÕES	45
5.2	DIFICULDADES	45
5.3	TRABALHOS FUTUROS	45
	REFERÊNCIAS	47
	APÊNDICE A – CÓDIGOS DO SIMULINK	49
	APÊNDICE B – CÓDIGO DE INICIALIZAÇÃO	55
	APÊNDICE C – CÓDIGOS USADOS NO PLANEJAMENTO DE TRAJETÓRIA	57

1 INTRODUÇÃO

Robôs bípedes são cada vez mais objetos de estudo, seja por acadêmicos ou por empresas particulares. Isso se deve ao fato de esse tipo de robô demonstrar uma capacidade excepcional na execução de algumas tarefas como subir escadas e ultrapassar pequenos obstáculos, fazendo isso com uma área de suporte relativamente pequena (OTT et al., 2012). Além disso, esses robôs tem condições de se adaptar a ambientes frequentados por humanos por serem morfologicamente similares (LUO et al., 2019). A Figura 1 apresenta um dentre os diversos exemplos existentes desse tipo de máquina. Dentre as diferentes dificuldades no desenvolvimento de um robô bípede, a estabilidade do robô, o planejamento da passada, o controle combinado da passada e da estabilização e os GDL's (graus de liberdade) das pernas podem ser considerados os principais desafios (JAMISOLA; ROBERTS, 2021). Dessa forma, é importante que sejam feitos estudos nas diferentes possíveis formas de se controlar e estabilizar essa máquina.

Figura 1 – Exemplo de um robô bípede



Fonte: Luo et al. (2019)

Kuindersma et al. (2016) estudaram a otimização de algoritmos para o funcionamento de robôs bípedes em situações de catástrofes onde esse tipo de máquina pode conseguir acessar locais que outras máquinas não alcançam e/ou que seriam muito arriscados para humanos irem. Já Moreno, Figueiredo e Pons (2018) mostram a possibilidade de uso, para reabilitação, de um tipo específico de robô bípede, o exoesqueleto.

Há várias formas de se construir esse tipo de robô e a mais utilizada é inserindo servomotores nas juntas para que o robô consiga se movimentar. Entretanto, há alguns estudos que buscam uma forma mais eficiente de movimentação que não necessite de tantos atuadores ou que esses atuadores realizem o mínimo possível de esforços. Nessa linha estão os chamados *Passive-Dynamic Walkers* que são robôs bípedes que caminham

sem a necessidade de qualquer atuação em um plano inclinado ou com um mínimo de atuação em locais planos (COLLINS et al., 2005). Um exemplo pode ser visto na Figura 2.

Figura 2 – Exemplo de um *Passive-Dynamic Walker*



Fonte: Collins et al. (2005)

Dentre as opções para o controle da inclinação do robô a mais comum é a utilização do movimento pendular no qual o deslocamento do centro de massa é feito “jogando” a cintura de um lado para o outro. Choi et al. (2007) usaram um método baseado no centro de massa e ponto de momento zero para desenvolver um controle que possibilitasse a chamada coordenação de corpo inteiro. Dessa forma foi possível obter uma alta robustez contra distúrbios. Já Ott et al. (2012) usaram sensores de torque nas juntas do robô para fazer o sensoramento, ao invés da abordagem utilizando o ponto de momento zero.

Outras formas não convencionais para melhorar o equilíbrio também foram propostas. Uma delas é a apresentada por Kim et al. (2021) na qual são usados propulsores para auxiliar no equilíbrio durante a passada (que utiliza movimento pendular), como mostrado na Figura 3, mas também permitem que o robô possa voar, andar de skate, e fazer *slackline*.

Todas as estratégias para estabilização e equilíbrio anteriormente apresentadas levam em consideração o planejamento da passada para fazer o controle de estabilidade. Na tentativa de tornar esses problemas independentes, Bragaia e Araujo (2020) propuseram uma outra solução para o controle da inclinação do robô através do uso de um contrapeso em forma de cauda (esse trabalho foi posteriormente estendido por Oliveira e Araujo (2020)). Assim o controle da passada e o da inclinação podem ser trabalhados separadamente, já que não há necessidade do movimento pendular, possibilitando que as técnicas de controle

Figura 3 – Robô bípede controlado por propulsores



Fonte: Kim et al. (2021)

e as de planejamento de trajetória sejam estudadas isoladamente e também testadas separadamente em um modelo real ou mesmo em uma simulação.

Dessa forma, seguindo e expandindo a linha de pesquisa de Oliveira e Araujo (2020), o presente trabalho procurou desenvolver e validar um ambiente de simulação, utilizando MATLAB e SIMULINK, de um robô bípede cujo controle de inclinação seja feito através de uma nova abordagem: utilizando um contrapeso com movimento linear. Ainda que a forma de movimentação do contrapeso seja diferente da originalmente apresentada por Oliveira e Araujo (2020), pequenas alterações podem ser feitas para se adaptar o modelo para a simulação com a cauda.

1.1 OBJETIVOS

O presente trabalho tem como objetivo geral verificar e validar o controle de estabilidade de um robô bípede utilizando um contrapeso como fator de balanceamento dos momentos do corpo.

Para atingir tal objetivo, os objetivos específicos são listados abaixo:

- Revisão do projeto mecânico buscando melhor estabilidade;
- Implementação do ambiente de simulação de um robô bípede no qual o usuário possa testar lógicas de controle e planejamento de trajetória;
- Atualização e implementação do planejamento de trajetória em relação a estudada por Oliveira e Araujo (2020);

- Realização de movimentação quase-estática do robô no ambiente de simulação a fim de validar o modelo comparando os resultados da simulação com os resultados do modelo analítico estático;
- Elaboração do controle de posição do contrapeso.

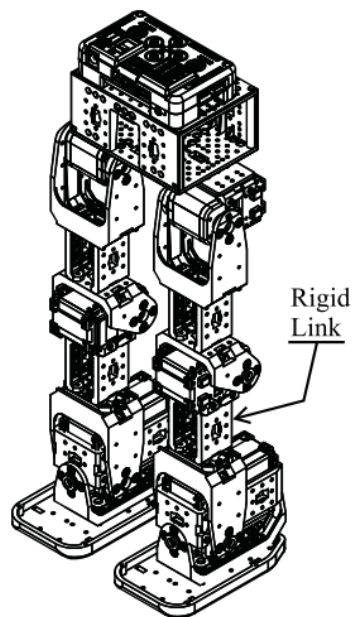
2 FUNDAMENTAÇÃO TEÓRICA

Há características que devem ser consideradas para se projetar e construir um robô bípede, sendo algumas delas o número de Graus de Liberdade, a forma de equilíbrio, o método de controle e o planejamento da trajetória (JAMISOLA; ROBERTS, 2021). Neste capítulo, são apresentados os conceitos básicos destas características e algumas das considerações realizadas durante o desenvolvimento do trabalho.

2.1 GDL's E EQUILÍBRIO

Os GDL's podem ser definidos como as movimentações geradas por cada junta. Em robôs clássicos, cada junta gera apenas 1 GDL, sendo as juntas rotativas, para movimento angular, ou prismáticas, para movimento linear (SICILIANO et al., 2009). Especificamente para robôs bípedes, o número de GDL's pode variar dependendo da aplicação e do objetivo. Os robôs com 8 GDL's são os mais encontrados na literatura já que imitam o número de graus de liberdade das pernas humanas. Um exemplo desse tipo de robô é o desenvolvido por Sarkar, Kishore e Dutta (2014) que pode ser visto na Figura 4.

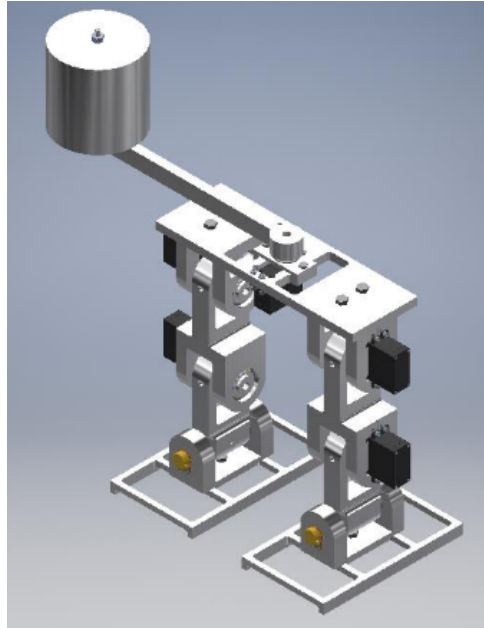
Figura 4 – Exemplo de robô bípede com 8 GDL's



Fonte: Sarkar, Kishore e Dutta (2014)

Nesses casos, a utilização de 8 graus de liberdade é necessária por conta do tipo de equilíbrio utilizado no projeto, que faz o robô realizar um movimento pendular. Porém, ao se alterar o tipo de equilíbrio, torna-se possível diminuir o número de GDL's, como mostrado por Bragaia e Araujo (2020) e Oliveira e Araujo (2020) que utilizaram um contrapeso com movimentação angular para realizar o equilíbrio do robô, como mostrado na Figura 5. Dessa forma foi proposta a ideia de realizar o equilíbrio utilizando um contrapeso com movimentação linear.

Figura 5 – Robô desenvolvido por Oliveira e Araujo (2020)



Fonte: Oliveira e Araujo (2020)

2.2 MÉTODO DE CONTROLE

A forma mais intuitiva de se equilibrar algo é posicioná-lo de forma que todas as suas forças e torques se cancelem em um ponto de equilíbrio. Porém, quando se fazem análises dinâmicas, é importante considerar também as forças de reação que irão aparecer durante a movimentação. Essas forças são importantes a ponto de existirem métodos de controle completamente baseados nelas e alguns mecanismos bem conhecidos como rodas de reação.

Um dos problemas de controle mais estudados são os pêndulos invertidos por conta de sua natureza instável e não-linear (NEVES; ANGELICO, 2021). Um dos tipos de pêndulos invertidos existentes é o que utiliza rodas de reação para se equilibrar, como o construído por Aghaei et al. (2021). Eles utilizaram o método lagrangiano para modelar o pêndulo e, após a linearização, desenvolveram dois controladores, um PID e um baseado em espaço de estados.

Neves e Angelico (2021) utilizaram uma roda de reação para equilibrar seu robô que possuía somente uma roda de apoio e portanto era também instável em sua movimentação lateral. Eles também utilizaram uma abordagem lagrangiana para fazer a modelagem e controle ótimo LQR.

Já Trentin et al. (2020) fizeram um estudo de um pêndulo invertido utilizando duas rodas de reação, cujo controle poderia ser feito utilizando um PID simples, e desenvolveram mais dois tipos de controlador para essa configuração, um PD não linear e um *sliding mode controller* (SMC).

Em uma abordagem mais prática, Belascuen e Aguilar (2019) estudaram formas de otimizar a construção de um pêndulo invertido, desde o projeto mecânico até a escolha dos parâmetros do motor elétrico utilizado.

Percebe-se que o pêndulo invertido controlado por rodas de reação é bastante estudado e possui uma vasta bibliografia, porém, no caso do presente trabalho a forma de controle escolhida não foi uma roda de reação, mas um contrapeso com movimento linear. A força de reação desse contrapeso será considerada para realizar a estabilização do robô bípede.

2.3 PLANEJAMENTO DA TRAJETÓRIA

O planejamento de trajetória tem como finalidade a geração de uma referência para o sistema de controle que deverá atuar para manter os movimentos dentro do planejado (SICILIANO et al., 2009). Há duas formas principais de se planejar uma trajetória, o espaço das juntas, que trabalha com os ângulos das juntas determinados, e o espaço das tarefas, que trabalha com a posição do efetuador final determinada. No presente trabalho, devido a não necessidade de um posicionamento preciso dos pés do robô e a simplicidade do movimento foi utilizada a abordagem utilizando as juntas.

Dentro dessa abordagem há ainda duas técnicas para determinação da trajetória, a movimentação ponto-a-ponto e a movimentação através de uma sequência de pontos. Essa última é utilizada quando há a necessidade de um controle maior da movimentação durante sua execução (SICILIANO et al., 2009), como esse não é o caso do presente trabalho, foi utilizada a movimentação ponto-a-ponto.

Nesse tipo de movimentação a máquina deve partir de uma configuração inicial e chegar a uma configuração final predeterminadas em um certo espaço de tempo, não importando o caminho que a extremidade faça no percurso (SICILIANO et al., 2009). Para isso utiliza-se um polinômio de quinta ordem como mostrado na Equação 1, onde $q(t)$ é o ângulo de determinada junta, $a_5 \dots a_0$ são coeficientes a serem definidos após a imposição de condições de contorno e t é o tempo.

$$q(t) = a_5t^5 + a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0 \quad (1)$$

Nesse caso, as condições de contorno são as posições, velocidades e acelerações angulares iniciais e finais que cada junta irá experimentar.

3 DESENVOLVIMENTO

A elaboração da simulação foi dividida em quatro grandes partes: revisão do projeto mecânico, montagem do código usando programação em blocos através do SIMULINK, simulação quase-estática com movimentos lentos e sem uma lógica de controle definida e simulação completa com lógica de controle definida.

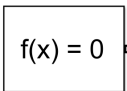

3.1 REVISÃO DO PROJETO MECÂNICO

Durante o início do presente projeto, foi feita uma simulação do robô original, com cauda, proposto por Bragaia e Araujo (2020) e verificou-se que a cauda fazia com que o robô tombasse para trás quando ela estivesse perpendicular com a cintura. Além disso, verificou-se que o comprimento necessário dessa cauda para que fosse possível equilibrar o robô acabaria sendo desproporcionalmente grande. Por conta disso, duas mudanças importantes foram feitas em relação ao modelo original: a mudança da geometria da cintura do robô, retirando os apoios para o servomotor que faria a movimentação da cauda, e a adição de um subconjunto contendo uma guia linear com seu respectivo bloco e um contrapeso. Esse subconjunto, diferentemente do restante do robô, foi modelado diretamente no SIMULINK utilizando o bloco *Brick solid*, já que se tratava de uma geometria simples.

3.2 MONTAGEM DO CÓDIGO








Para a montagem do código, foi utilizado o programa SIMULINK e a biblioteca Simscape - Multibody, além de alguns blocos de uso geral. Todos os blocos utilizados e suas respectivas funções podem ser vistos no Quadro 1.

Quadro 1 – Blocos utilizados na simulação

Blocos do SIMULINK	Descrição
 <p>Solver Configuration</p>	Define as configurações do <i>solver</i> utilizadas na simulação.
 <p>World Frame</p>	<i>Frame</i> de referência.


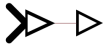





Continua na próxima página

Quadro 1 – Continuação da página anterior

Blocos do SIMULINK	Descrição
 <p data-bbox="293 452 437 510">Mechanism Configuration</p>	<p data-bbox="552 394 1316 474">Define as configurações físicas utilizadas na simulação, como direção e módulo da gravidade.</p>
 <p data-bbox="300 676 421 734">Rigid Transform</p>	<p data-bbox="552 618 1316 698">Utilizada para definir as posições relativas entre as peças na montagem.</p>
 <p data-bbox="277 922 437 958">Brick Solid</p>	<p data-bbox="552 864 1157 900">Gera um sólido em forma de paralelepípedo.</p>
 <p data-bbox="271 1137 453 1173">Spherical Solid</p>	<p data-bbox="552 1039 1316 1164">Gera um sólido em forma de esfera. Utilizado nos pés do robô para permitir a utilização do bloco <i>Spatial contact force</i>.</p>
 <p data-bbox="293 1370 421 1406">File Solid</p>	<p data-bbox="552 1281 1316 1361">Gera um sólido baseado em uma geometria importada de um arquivo CAD.</p>
 <p data-bbox="284 1572 437 1608">6-DOF Joint</p>	<p data-bbox="552 1482 1316 1608">Junta que permite todos os 6 graus de liberdade. Usada para permitir a movimentação do robô em relação ao <i>World frame</i>.</p>
 <p data-bbox="271 1809 453 1845">Revolute Joint</p>	<p data-bbox="552 1751 1268 1787">Junta que permite um grau de liberdade de rotação.</p>

Continua na próxima página

Quadro 1 – Continuação da página anterior

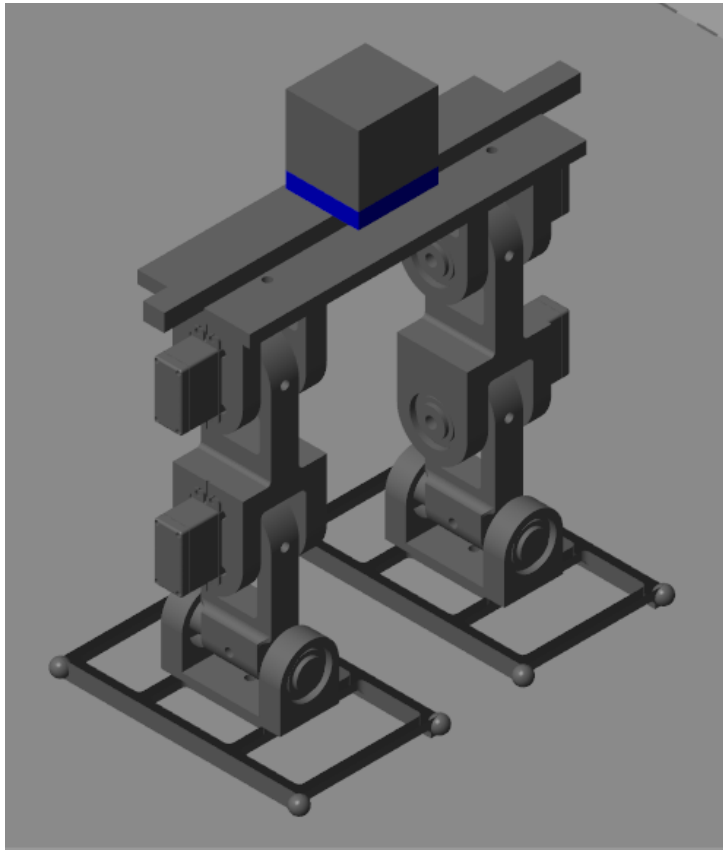
Blocos do SIMULINK	Descrição
 <p data-bbox="379 456 496 517">Prismatic Joint</p>	<p data-bbox="627 416 1385 454">Junta que permite um grau de liberdade de translação.</p>
 <p data-bbox="347 636 533 712">Simulink-PS Converter</p>	<p data-bbox="627 613 1390 696">Converte os sinais de controle do SIMULINK em sinais físicos para os blocos do <i>Simscape Multibody</i>.</p>
 <p data-bbox="347 909 533 936">MATLAB Function</p>	<p data-bbox="627 837 1390 920">Permite implementar um código do MATLAB dentro do SIMULINK.</p>
 <p data-bbox="357 1144 507 1171">Simscape Bus</p>	<p data-bbox="627 1061 1390 1144">Junta os sinais em uma única ligação. Utilizado para deixar o ambiente de simulação mais limpo.</p>
 <p data-bbox="379 1346 496 1395">Clock</p>	<p data-bbox="627 1308 1129 1346">Fornece o tempo atual da simulação.</p>
 <p data-bbox="352 1592 523 1619">Infinite Plane</p>	<p data-bbox="627 1509 1390 1592">Um plano infinito. Utilizado como superfície por onde o robô poderia caminhar.</p>
 <p data-bbox="363 1794 517 1854">Spatial Contact Force</p>	<p data-bbox="627 1733 1390 1816">Simula as forças de contato entre uma superfície esférica e uma superfície plana.</p>

Fonte: próprio autor

A geometria principal do robô foi obtida através do modelo CAD desenvolvido por Oliveira e Araujo (2020) e pode ser vista na Figura 6.

A programação da simulação foi dividida em 6 subsistemas: as pernas esquerda e

Figura 6 – Visualização da geometria do robô



Fonte: próprio autor

direita, os contatos entre ambas as pernas e o plano infinito, o conjunto guia e contrapeso e o planejamento de trajetória. Todos esses subconjuntos foram interligados através da cintura do robô e das portas de atuação das juntas. Os códigos são mostrados no Apêndice A. O código geral da simulação pode ser visto na Figura A-1.

Os subsistemas “Perna esquerda”, “Perna direita” e “Guia e contrapeso” são mostrados nas Figuras A-2, A-3 e A-4, respectivamente.

A movimentação do robô foi dividida em 6 fases, cada uma com duração de 5 segundos, para a simulação quase-estática:

- Fase 1: movimentação do contrapeso do centro do robô para a posição de equilíbrio;
- Fase 2: levantamento da perna direita
- Fase 3: movimentação para trás da perna esquerda
- Fase 4: movimentação do contrapeso para a nova posição de equilíbrio
- Fase 5: alinhamento da perna direita, o que faz com que a perna esquerda levante do solo

- Fase 6: alinhamento da perna esquerda levando o robô de volta a sua configuração inicial.

Para o planejamento de trajetória, foi utilizada a Equação 1 e as condições de contorno mostradas na Tabela 1, na qual q_{q-d} é o ângulo da junta do quadril direito, q_{c-d} é o ângulo da junta da coxa direita, q_{q-e} é o ângulo da junta do quadril esquerdo e q_{c-e} é o ângulo da junta do quadril esquerdo. As velocidades e acelerações não foram mostradas, pois ambas valem zero tanto no início quanto no fim do movimento para todas as juntas.

Tabela 1 – Condições de contorno de posição para cada junta em cada fase, em rad

Fase	q_{q-d} (0)	q_{q-d} (5)	q_{c-d} (0)	q_{c-d} (5)	q_{q-e} (0)	q_{q-e} (5)	q_{c-e} (0)	q_{c-e} (5)
1	0	0	0	0	0	0	0	0
2	0	$-\pi/6$	0	$\pi/6$	0	0	0	0
3	$-\pi/6$	$-\pi/6$	$\pi/6$	$\pi/6$	0	$-\pi/6$	0	$\pi/6$
4	$-\pi/6$	$-\pi/6$	$\pi/6$	$\pi/6$	$-\pi/6$	$-\pi/6$	$\pi/6$	$\pi/6$
5	$-\pi/6$	0	$\pi/6$	0	$-\pi/6$	$-\pi/6$	$\pi/6$	$\pi/6$
6	0	0	0	0	$-\pi/6$	0	$-\pi/6$	0

Fonte: próprio autor

3.3 MOVIMENTAÇÃO QUASE-ESTÁTICA

Para simular a movimentação quase-estática, foi necessário realizar um equilíbrio de forças no robô quando mono-apoiado a fim de encontrar a massa do contrapeso necessária para equilibrá-lo. O DCL (diagrama de corpo livre) do robô nessa condição pode ser visto na Figura 7, no qual F_P é a força exercida pela perna levantada, F_{CE} é a força exercida pela parte esquerda da cintura, F_{CD} é a força exercida pela parte direita da cintura e M_S é o momento realizado pelos servomotores presos à perna de apoio. As distâncias dos CG's relativos a cada força e as massas, apresentadas na Tabela 2, foram obtidas através do modelo CAD.

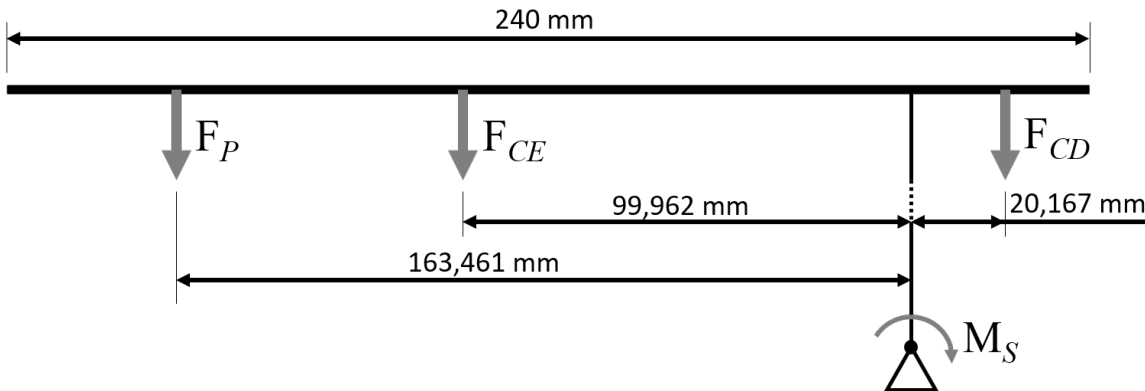
Tabela 2 – Massas relativas às forças presentes no DCL da Figura 7, em kg

m_P	m_{CE}	m_{CD}	m_S
1,385	0,386	0,077	0,144

Fonte: próprio autor

Utilizando esses dados, foi possível calcular os momentos no apoio como mostram as Equações 2, 3, 4 e 5. A distância da força que gera M_S foi obtida também através do CAD e vale 40,412 mm.

Figura 7 – DCL do robô mono-apoiado



Fonte: próprio autor

$$F_P = m_p g$$

$$F_P = 13,59 \text{ N}$$

$$M_P = F_P \cdot 0,163$$

$$M_P = 2,22 \text{ Nm} \quad (2)$$

$$F_{CE} = m_{ce} g$$

$$F_{CE} = 3,79 \text{ N}$$

$$M_{CE} = F_P \cdot 0,1$$

$$M_{CE} = 0,379 \text{ Nm} \quad (3)$$

$$F_{CD} = m_{cd} g$$

$$F_{CD} = 0,755 \text{ N}$$

$$M_{CD} = F_P \cdot 0,0202$$

$$M_{CD} = 0,0153 \text{ Nm} \quad (4)$$

$$F_S = m_s g$$

$$F_S = 1,41 \text{ N}$$

$$M_S = F_P \cdot 0,0404$$

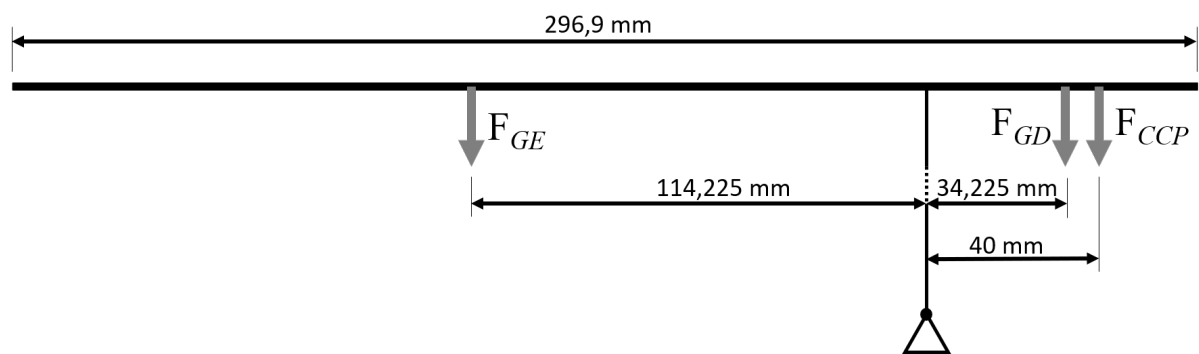
$$M_S = 0,057 \text{ Nm} \quad (5)$$

Fazendo a somatória de momentos em torno do apoio e adotando o sentido de giro positivo como sendo o anti-horário, foi possível obter o momento atuante no apoio, devido ao robô, como mostrado na Equação 6.

$$\begin{aligned}\Sigma M &= M_P + M_{CE} - M_{CD} - M_S \\ \Sigma M &= 2,53 \text{ Nm}\end{aligned}\quad (6)$$

Além das momentos gerados pelo robô, há também os momentos gerados pelo sistema guia e contrapeso. O DCL desse sistema está mostrado na Figura 8, na qual F_{GE} é força exercida pela parte esquerda da guia, F_{GD} é a força exercida pela parte direita da guia e F_{CCP} é força exercida pelo conjunto do contrapeso.

Figura 8 – DCL do sistema guia e contrapeso



Fonte: próprio autor

O tamanho da guia foi escolhido de modo que a força exercida pelo contrapeso ficasse exatamente sobre a borda da cintura do robô, ou seja, a 120 mm do centro da cintura. Como o carrinho da guia linear mede 56,9 mm x 52 mm, escolheu-se fazer o contrapeso nas mesmas medidas, assim, para que seu centro de gravidade ficasse no ponto desejado foi necessário usar uma guia de 296,9 mm.

As distâncias dos CG's relativos a cada força foram obtidas através do modelo CAD. As massas, apresentadas na Tabela 3, foram calculadas com os seguintes dados de catálogos: a massa linear da guia vale 1,2 kg/m e a massa do carrinho vale 0,19 kg.

Tabela 3 – Massas relativas às forças presentes no DCL da Figura 8, em kg

m_{GE}	m_{GD}	m_c
0,274	0,082	0,19

Fonte: próprio autor

Com esses dados foi possível calcular os momentos relativos à guia e ao contrapeso atuantes no apoio, como mostram as Equações 7, 8 e 9.

$$\begin{aligned}
F_{GE} &= m_{ge}g \\
F_{GE} &= 2,69 \text{ N} \\
M_{GE} &= F_{GE} \cdot 0,114 \\
M_{GE} &= 0,31 \text{ Nm}
\end{aligned} \tag{7}$$

$$\begin{aligned}
F_{GD} &= m_{gd}g \\
F_{GD} &= 0,8 \text{ N} \\
M_{GD} &= F_{GD} \cdot 0,034 \\
M_{GD} &= 0,027 \text{ Nm}
\end{aligned} \tag{8}$$

$$\begin{aligned}
F_{CCP} &= m_{ccp}g \\
F_{CCP} &= (9,81m_{cp} + 1,864) \text{ N} \\
M_{CCP} &= F_{CCP} \cdot 0,04 \\
M_{CCP} &= (0,392m_{cp} + 0,075) \text{ Nm}
\end{aligned} \tag{9}$$

Fazendo a somatória de momentos em torno do apoio e adotando o sentido de giro positivo como sendo o anti-horário foi possível obter o momento atuante no apoio, devido a guia e contrapeso, como mostrado na Equação 10.

$$\begin{aligned}
\Sigma M &= M_{GE} - M_{GD} - M_{CCP} \\
\Sigma M &= (0,2 - 0,392m_{cp}) \text{ Nm}
\end{aligned} \tag{10}$$

Finalmente, usando as Equações 6 e 10, foi possível fazer o equilíbrio de momentos no apoio mostrado da Equação 11, obtendo-se, assim, o valor do contrapeso.

$$\begin{aligned}
\Sigma M &= 0 \\
m_{cp} &= 6,96 \text{ kg}
\end{aligned} \tag{11}$$

Como esse tipo de movimentação não possui um sistema de controle para realizar o equilíbrio do robô, foi necessário fazer um planejamento de trajetória para o contrapeso. Obedecendo as 6 fases propostas, o contrapeso só irá se movimentar nas fases 1 e 4, que são as fases na qual o robô está bi-apoiado. As condições de contorno, utilizadas junto da Equação 1, podem ser vistas na Tabela 4. As acelerações e velocidades não foram mostradas, pois ambas valem zero tanto no início quanto no final de todas as movimentações.

Tabela 4 – Condições de contorno de posição do contrapeso em cada fase, em mm

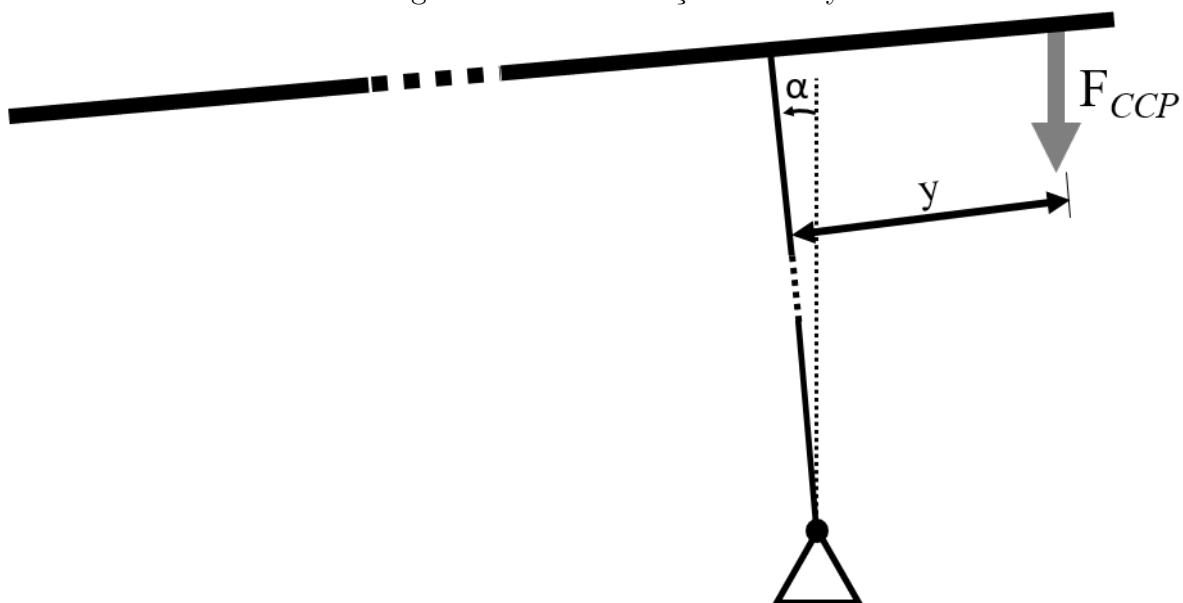
Fase	$p_{contrapeso} (0)$	$p_{contrapeso} (5)$
1	0	120
2	120	120
3	120	120
4	120	-120
5	-120	-120
6	-120	-120

Fonte: próprio autor

3.4 SIMULAÇÃO COMPLETA

Para ser possível controlar o robô de forma dinâmica, a primeira coisa feita foi encontrar a FT correspondente. Utilizando os mesmos DCL's mostrados nas Figuras 7 e 8, porém considerando o robô inclinado para a esquerda o que gera um ângulo α positivo, como mostrado na Figura 9, e a Equação 12, foi possível obter a relação mostrada na Equação 13, na qual α é o ângulo e Y é a posição do contrapeso. Para esse cálculo foi considerada uma mola torcional com k_t de 30 Nm localizada no pé do robô, ou seja, no ponto de giro, e um momento de inércia de massa, obtido através do modelo CAD, de $0,4283 \text{ kgm}^2$. Essa mola torcional foi inserida para absorver os erros de aproximação que ocorrem durante a simulação, pois como esse é um problema naturalmente instável, qualquer erro de aproximação pode fazer o robô cair para algum lado.

Figura 9 – Visualização de α e y



Fonte: próprio autor

$$\Sigma M = I\ddot{\alpha} \quad (12)$$

$$Y(0,6994s^2 - 26,39) = \alpha(0,4283s^2 + 16,77) - 3,048 \quad (13)$$

A fim de desenvolver um controle utilizando essa equação foi necessário linearizá-la, ou seja, remover o termo independente. Esse termo aparece na equação devido a uma linearização anterior, na qual os cossenos são aproximados para 1. Assim a FT obtida para o sistema pode ser vista na Equação 14. Pode-se perceber que essa FT é instável, ou seja, apresenta polos no semiplano direito do lugar das raízes.

$$\frac{Y}{\alpha} = \frac{0,4283s^2 + 16,77}{0,6994s^2 - 26,39} \quad (14)$$

Essa abordagem foi feita utilizando como base as posições de equilíbrio para cada ângulo dado, porém, ao se movimentar, o contrapeso exerce forças de reação no robô, de forma que, se o robô estivesse caindo para a direita, o contrapeso se moveria para a esquerda, porém isso geraria uma força de reação para a direita o que faria o robô inclinar ainda mais, gerando um *loop* no qual o robô nunca conseguiria se equilibrar. Por conta disso, foi proposta a utilização dessas forças de reação para o controle do robô.

Nesse caso, devido a uma maior complexidade na modelagem, foi utilizado uma abordagem na qual o contrapeso se moveria dependendo de α e $\dot{\alpha}$ da seguinte forma: se $\alpha > 0$ e $\dot{\alpha} > 0$ ou se $\alpha < 0$ e $\dot{\alpha} < 0$ seria utilizado um PID agressivo para gerar forças de reação suficiente para equilibrar o robô e se $\alpha > 0$ e $\dot{\alpha} < 0$ ou se $\alpha < 0$ e $\dot{\alpha} > 0$ seria utilizado um PID menos agressivo para levar o contrapeso de volta a posição de equilíbrio. O código desse modelo pode ser visto nas Figuras A-5 e A-6.

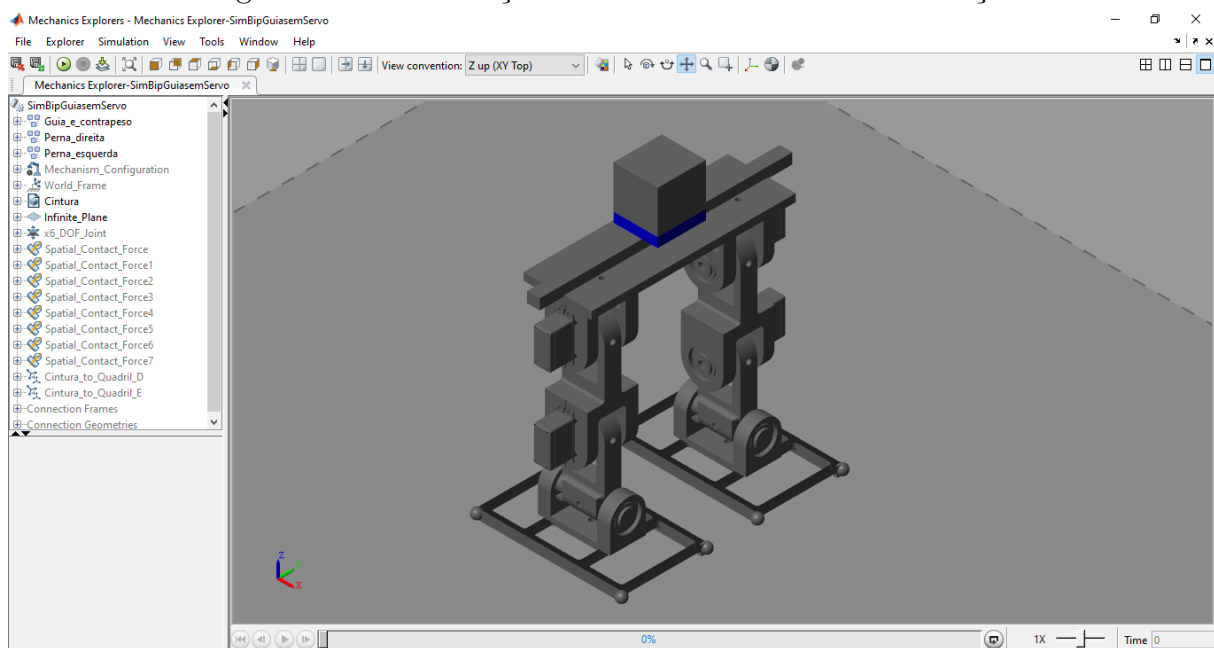
4 RESULTADOS

Os resultados do presente trabalho podem ser divididos em quatro partes: o ambiente de simulação funcional, o planejamento de trajetória, a movimentação quase-estática e o controle.

4.1 AMBIENTE DE SIMULAÇÃO

Os códigos desenvolvidos para o ambiente de simulação já foram apresentados no Capítulo 3. Como o pacote utilizado para a implementação dos códigos foi o *SimMechanics* a visualização 3D do robô é possível, como mostrado na Figura 10.

Figura 10 – Visualização do robô no ambiente de simulação



Fonte: próprio autor

Para que fosse mais fácil alterar os valores de alguns parâmetros da simulação sem precisar fazer isso manualmente em cada bloco, um código de inicialização, que pode ser visto no Apêndice B, foi utilizado.

4.2 PLANEJAMENTO DE TRAJETÓRIA

Os coeficientes obtidos para os ângulos q_{q-d} e q_{q-e} podem ser vistos nas Tabelas 5 e 6, respectivamente. Os coeficientes para os ângulos q_{c-d} e q_{c-e} são os mesmos de q_{q-d} e q_{q-e} , respectivamente, porém com os sinais invertidos, por isso não serão mostrados. Os coeficientes da movimentação do contrapeso são apresentados na Tabela 7. Esses coeficientes são relativos à Equação 1.

Utilizando essas equações foi possível implementar a movimentação das pernas do robô. Essa movimentação foi utilizada tanto na parte quase-estática quanto na parte de controle. Os códigos usados na geração das trajetória são apresentados no Apêndice C.

Tabela 5 – Coeficientes da Equação 1 para q_{q-d}

Fase	a_0	a_1	a_2	a_3	a_4	a_5
1	0	0	0	0	0	0
2	0	0	0	$-\pi/75$	$\pi/250$	$-\pi/3125$
3	$-\pi/6$	0	0	0	0	0
4	$-\pi/6$	0	0	0	0	0
5	$-\pi/6$	0	0	$\pi/75$	$-\pi/250$	$\pi/3125$
6	0	0	0	0	0	0

Fonte: próprio autor

Tabela 6 – Coeficientes da Equação 1 para q_{q-e}

Fase	a_0	a_1	a_2	a_3	a_4	a_5
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	$-\pi/75$	$\pi/250$	$-\pi/3125$
4	$-\pi/6$	0	0	0	0	0
5	$-\pi/6$	0	0	0	0	0
6	$-\pi/6$	0	0	$\pi/75$	$-\pi/250$	$\pi/3125$

Fonte: próprio autor

Tabela 7 – Coeficientes da Equação 1 para $p_{contrapeso}$

Fase	a_0	a_1	a_2	a_3	a_4	a_5
1	0	0	0	$48/5$	$-72/25$	$144/625$
2	120	0	0	0	0	0
3	120	0	0	0	0	0
4	120	0	0	$-96/5$	$144/25$	$-288/625$
5	-120	0	0	0	0	0
6	-120	0	0	0	0	0

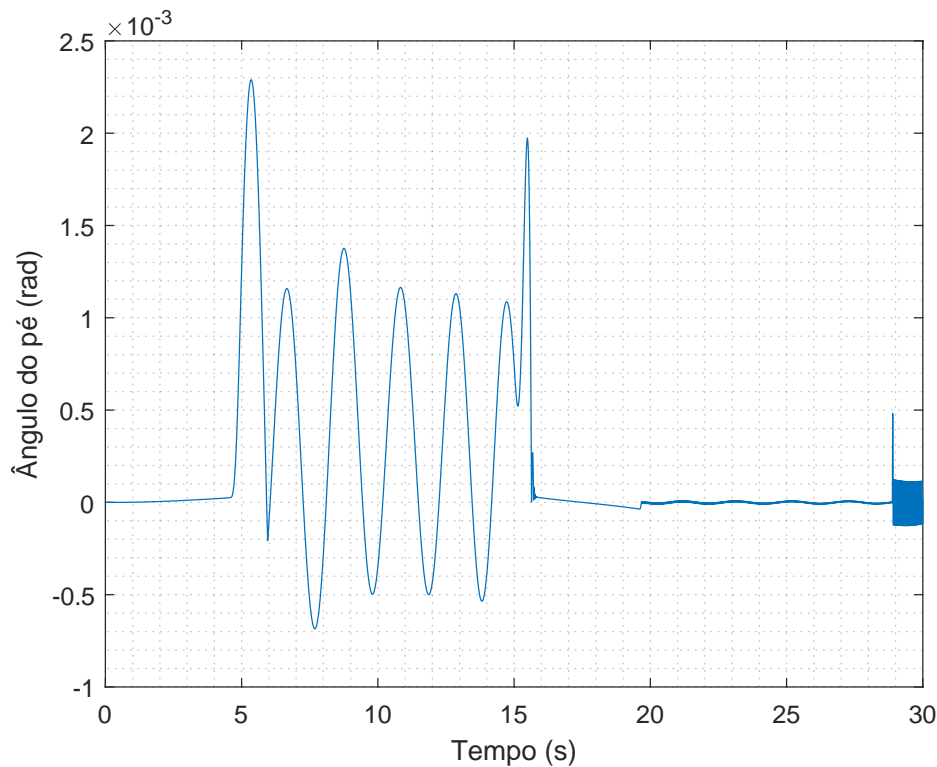
Fonte: próprio autor

4.3 MOVIMENTAÇÃO QUASE-ESTÁTICA

O objetivo de fazer uma simulação com movimentos lentos foi o de validar o modelo utilizado, ou seja, para que o ambiente de simulação fosse considerado bem-representativo, ao se aplicar as mesmas características encontradas no equilíbrio estático o robô não deveria tombar.

A Figura 11 mostra um gráfico da variação do ângulo do pé esquerdo do robô (pé de apoio durante as Fases 1 e 2) em relação ao tempo durante essa simulação. Como é possível ver por esse gráfico, a maior variação do ângulo da perna de apoio foi de $2,3 \times 10^{-3}$ radianos, o que confirma que a simulação é bem-representativa. A variação do ângulo entre 5 e 15 segundos se deve ao fato que, sendo um problema naturalmente instável, qualquer variação numérica que ocorra durante os cálculos da simulação pode acarretar na instabilidade do robô, dessa forma a mola torcional utilizada ajuda a mitigar esses efeitos

Figura 11 – Variação do ângulo do pé esquerdo com o tempo



Fonte: próprio autor

e gera a variação vista no gráfico. Já a variação de alta frequência nos últimos segundos acontece por que é nesse momento que esse pé encosta novamente no chão, sofrendo ação das forças de contato modeladas.

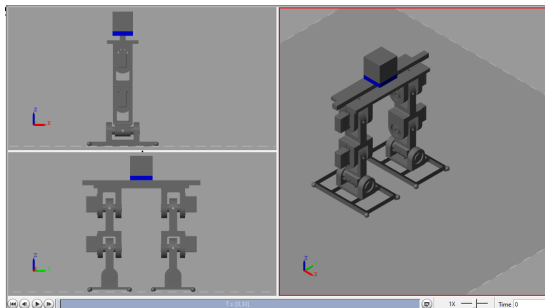
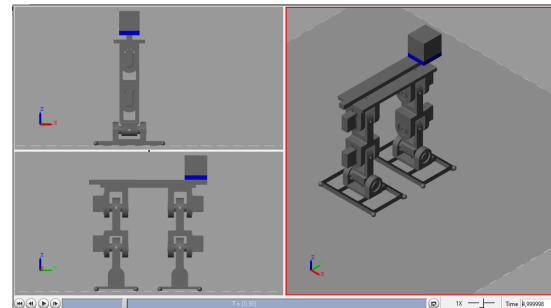
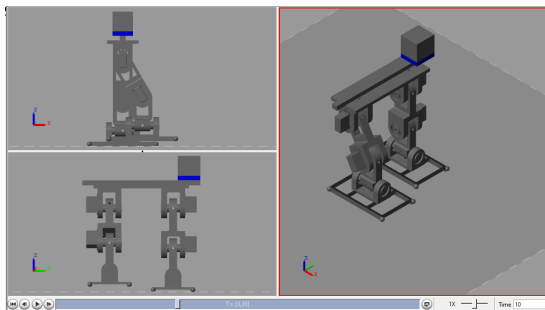
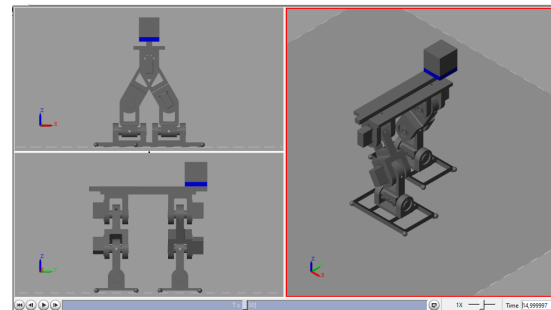
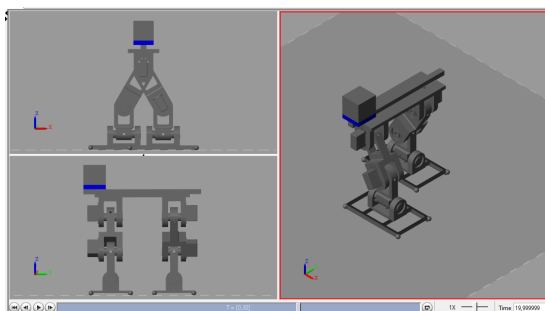
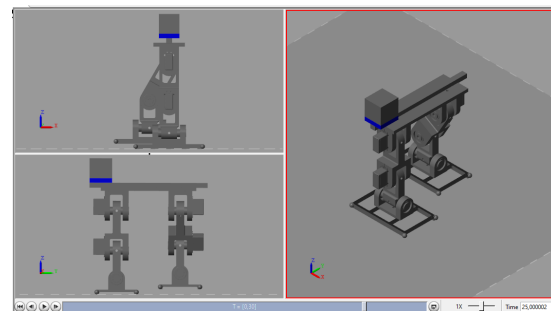
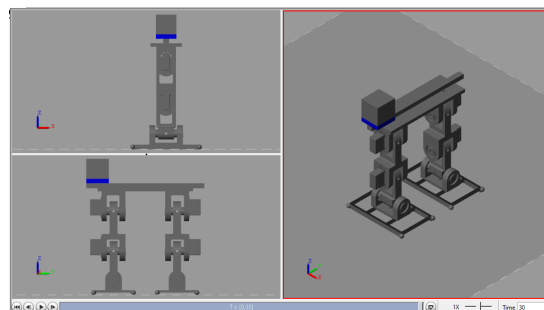
Todas as fases da movimentação quase-estática podem ser vistas na Figura 12 sendo a Fase 1 compreendida entre 0 s e 5 s, a Fase 2 entre 5 s e 10 s, a Fase 3 entre 10 s e 15 s, a Fase 4 entre 15 s e 20 s, a Fase 5 entre 20 s e 25 s e a Fase 6 entre 25 s e 30 s.

4.4 CONTROLE

Como já mencionado no Capítulo 3, a abordagem utilizando a FT mostrada na Equação 14 não funcionou devido a presença de forças de reação que agiam contra o equilíbrio do robô e não foram levadas em conta durante a modelagem. Ou seja, a tentativa de controle usando simplesmente a posição de equilíbrio estático do robô, baseada em cada ângulo, não se demonstrou estável.

Por conta desse problema, foi feita uma tentativa de controle utilizando as forças de reação para levar o robô de volta a uma posição mais ereta e o controle baseado na posição de equilíbrio para levar o contrapeso para a posição de equilíbrio. Porém, devido a complexidade e a instabilidade natural do sistema, não foi possível elaborar um modelo que englobasse esses dois tipos de controle. Ainda assim foram feitas tentativas usando algumas estimativas de valores de PID, porém não conseguiu-se um controle satisfatório.

Figura 12 – Fases da movimentação

(a) $t = 0$ s(b) $t = 5$ s(c) $t = 10$ s(d) $t = 15$ s(e) $t = 20$ s(f) $t = 25$ s(g) $t = 30$ s

Fonte: próprio autor

5 CONCLUSÃO

5.1 CONTRIBUIÇÕES

Dentre os resultados obtidos no presente trabalho, vale destacar aqueles que cumpriram os objetivos propostos e geraram material para futuros estudos.

O projeto mecânico originalmente proposto, que usava uma cauda, causava um desequilíbrio por conta do contrapeso estar fora da linha da cintura do robô. Assim o projeto mecânico foi alterado para a utilização de um controle usando contrapeso montado em uma guia linear de forma a resolver o problema de equilíbrio verificado.

Um ambiente de simulação foi desenvolvido como base para a implementação de estratégias de controle e planejamento da passada o que possibilitará que futuros trabalhos nesse mesmo tema avancem de maneira mais célere. Além disso, a trajetória utilizada por Oliveira e Araujo (2020) foi atualizada para poder ser utilizada com as restrições mecânicas presentes no projeto mecânico do robô.

Com a movimentação da passada implementada, esse ambiente de simulação foi validado através da movimentação quase-estática. Essa validação permite que se tenha uma forma de testar técnicas de controle e outros tipo de planejamento de trajetória sem a necessidade de se construir um protótipo.

Por fim, estratégias de controle para a movimentação dinâmica foram implementadas, entretanto não se obteve um resultado satisfatório.

5.2 DIFICULDADES

A maior dificuldade encontrada no decorrer do projeto foi o desenvolvimento do controle de inclinação durante a movimentação do robô devido à sua complexidade. Muitas tentativas foram realizadas, como considerar:

- a estabilidade estática do contrapeso e do peso do robô;
- o modelo dinâmico linearizado;
- uma combinação entre forças de reação e posição de equilíbrio.

Por conta disso, uma última estratégia foi utilizada com uma abordagem baseada em estimativas de valores de PID que se mostrou ineficaz e, portanto, foi abandonada. Ainda assim o ambiente necessário para a implementação desse controle foi desenvolvido e poderá ser utilizado posteriormente.

5.3 TRABALHOS FUTUROS

Futuros trabalhos nesse tema podem focar em alguns aspectos importantes a fim de ampliar o que foi desenvolvido nesse projeto:

- modelagem e controle do robô utilizando as forças de reação;

- utilização de outras técnicas de planejamento de trajetória.

Considerando os trabalhos executados, notou-se que as forças de reação realizam um importante papel na estabilização do robô bípede. Assim, ainda acredita-se que seja possível realizar o equilíbrio do robô utilizando estas forças.

A movimentação da passada foi feita de forma que não se considera sua força de movimentação para realizar o equilíbrio. Com o ambiente implementado, será possível estudar planejamentos que possam auxiliar no equilíbrio do robô.

REFERÊNCIAS

AGHAEI, V. T. et al. Modeling and simulation of the control performance of a reaction wheel pendulum. **SIU 2021 - 29th IEEE Conference on Signal Processing and Communications Applications, Proceedings**, p. 0–3, 2021.

BELASCUEN, G.; AGUILAR, N. Design, Modeling and Control of a Reaction Wheel Balanced Inverted Pendulum. **2018 IEEE Biennial Congress of Argentina, ARGENCON 2018**, IEEE, p. 1–9, 2019.

BRAGAIA, G. F.; ARAUJO, L. A. O. **Projeto conceitual de robô bípede**. Monografia (TCC) — Universidade Federal de São Carlos, São Carlos - SP, 2020.

CHOI, Y. et al. Posture/walking control for humanoid robot based on kinematic resolution of CoM Jacobian with embedded motion. **IEEE Transactions on Robotics**, v. 23, n. 6, p. 1285–1293, 2007. ISSN 15523098.

COLLINS, S. et al. Efficient bipedal robots based on passive-dynamic walkers. **Science**, v. 307, n. 5712, p. 1082–1085, 2005. ISSN 00368075.

JAMISOLA, R. S.; ROBERTS, R. G. An approach to drastically reduce the required legs DOFs for bipedal robots and lower-limb exoskeletons. **Robotica**, p. 1–15, 2021. ISSN 14698668.

KIM, K. et al. A bipedal walking robot that can fly , slackline , and skateboard. **Science Robotics**, v. 6, 2021.

KUINDERSMA, S. et al. Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. **Autonomous Robots**, v. 40, p. 429–455, 2016.

LUO, J. et al. Robust Bipedal Locomotion Based on a Hierarchical Control Structure. **Robotica**, v. 37, n. 10, p. 1750–1767, 2019. ISSN 14698668.

MORENO, J. C.; FIGUEIREDO, J.; PONS, J. L. Exoskeletons for lower-limb rehabilitation. In: COLOMBO, R.; SANGUINETI, V. (Ed.). **Rehabilitation Robotics**. [S.l.]: Elsevier Ltd., 2018. cap. 7, p. 89–99. ISBN 9780128119952.

NEVES, G. P.; ANGELICO, B. A. A discrete LQR applied to a self-balancing reaction wheel unicycle: Modeling, construction and control. **Proceedings of the American Control Conference**, American Automatic Control Council, v. 2021-May, p. 777–782, 2021. ISSN 07431619.

OLIVEIRA, G. C.; ARAUJO, L. A. O. **Desenvolvimento do projeto mecatrônico e construção de um protótipo de robô bípede com oito graus de liberdade: design e funcional**. São Carlos - SP, 2020.

OTT, C. et al. Hardware and Control Concept for an Experimental Bipedal Robot with Joint Torque Sensors. **Journal of the Robotics Society of Japan**, v. 30, n. 4, p. 378–382, 2012. ISSN 0289-1824.

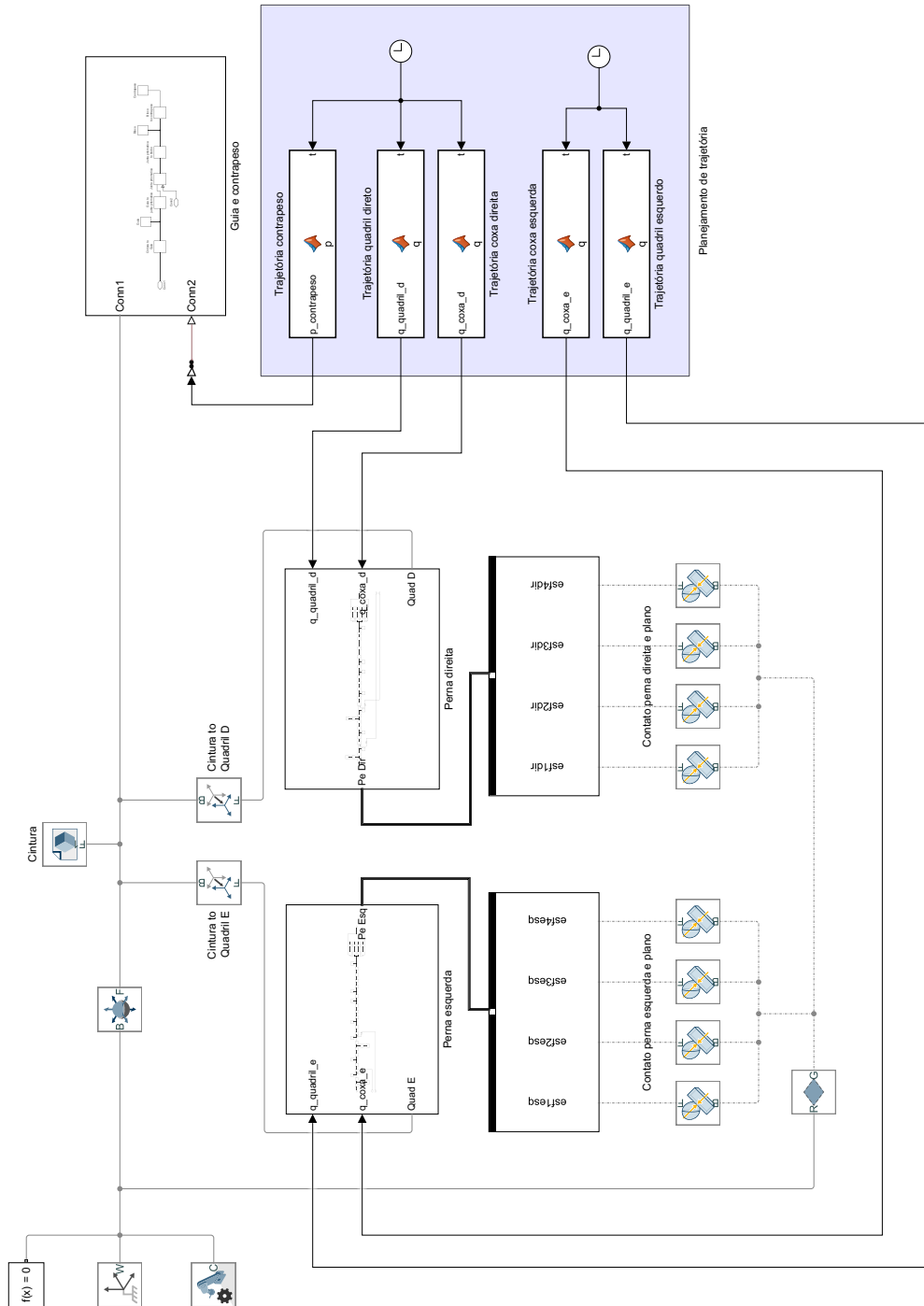
SARKAR, A.; KISHORE, N. N.; DUTTA, A. 8-DoF biped robot with compliant links. **2014 13th International Conference on Control Automation Robotics and Vision, ICARCV 2014**, IEEE, v. 2014, n. December, p. 1195–1200, 2014.

SICILIANO, B. et al. Trajectory planning. In: _____. **Robotics: Modelling, Planning and Control**. Londres: Springer London, 2009. cap. 4, p. 161–189. ISBN 9781846286414.

TRENTIN, J. F. S. et al. Inverted Pendulum Nonlinear Controllers Using Two Reaction Wheels: Design and Implementation. **IEEE Access**, v. 8, p. 74922–74932, 2020.

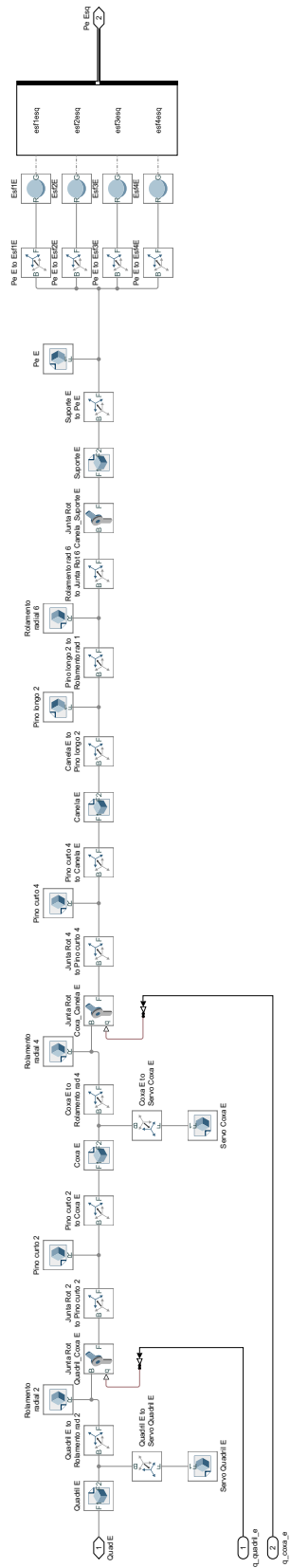
APÊNDICE A – Códigos do SIMULINK

Figura A-1 – Código geral da simulação



Fonte: próprio autor

Figura A-2 – Código do subsistema ”Perna esquerda”



Fonte: próprio autor

Figura A-4 – Código do subsistema “Guia e contrapeso”

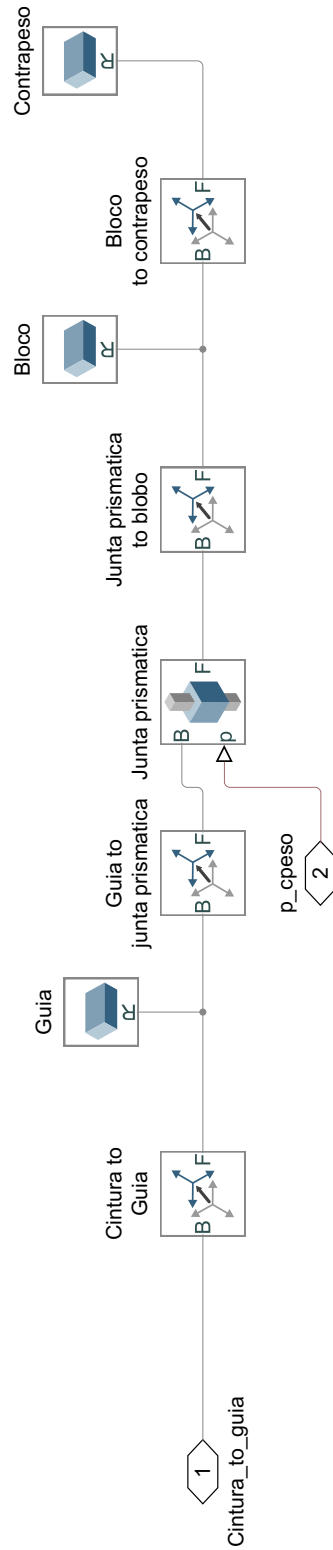
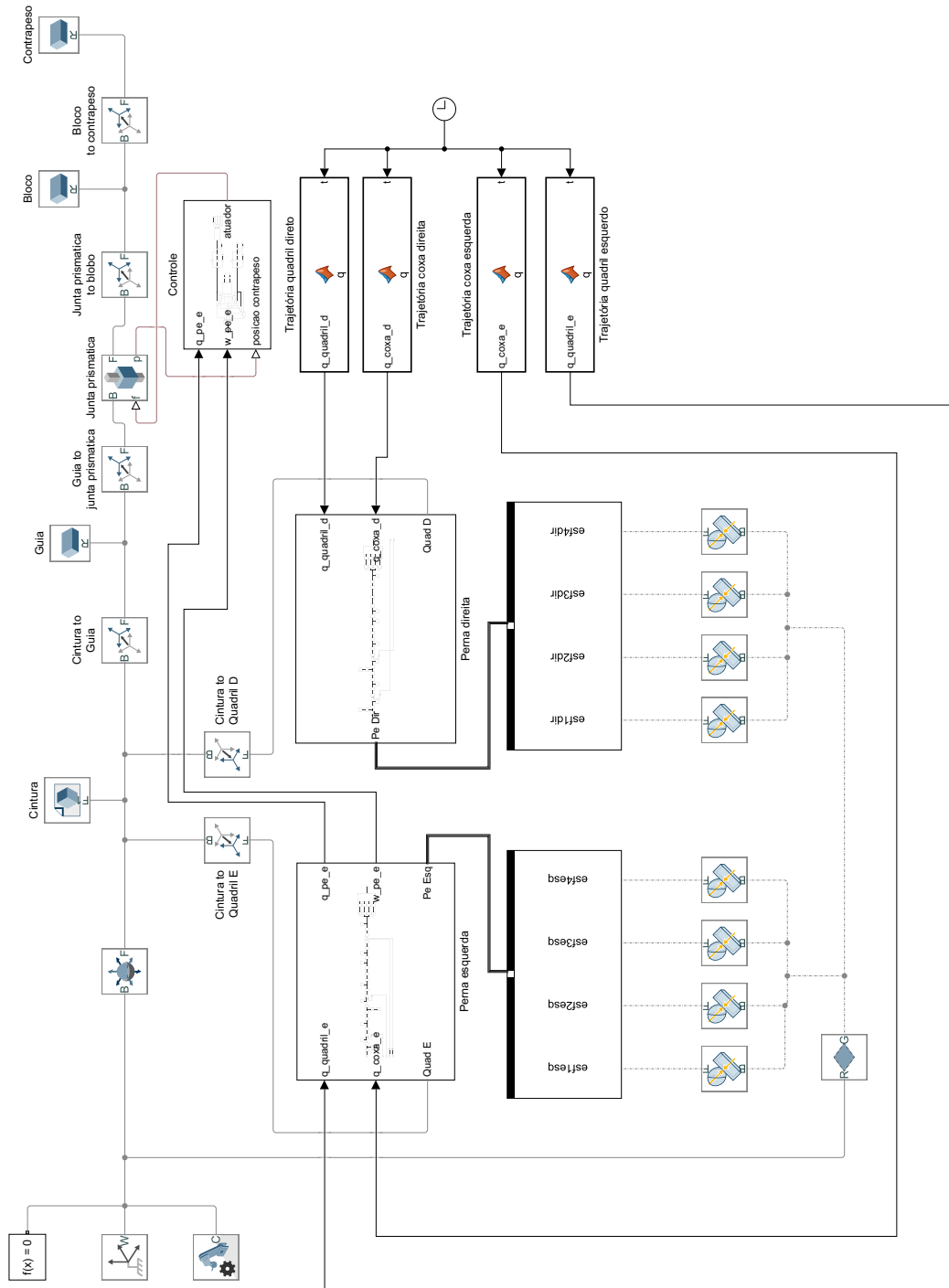
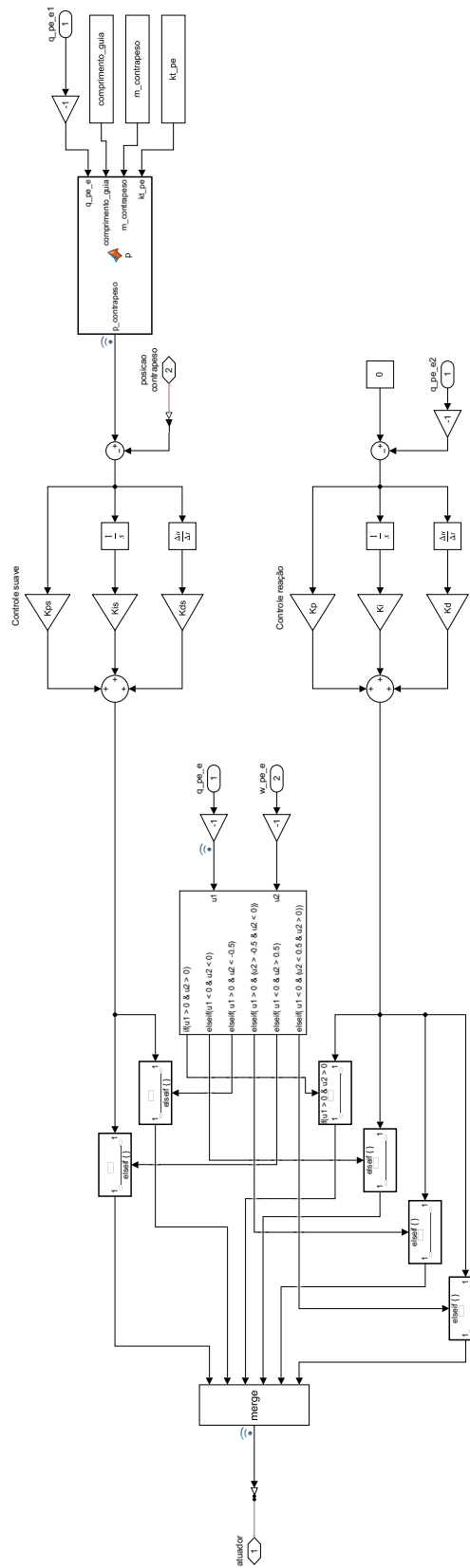


Figura A-5 – Código geral da simulação de controle



Fonte: próprio autor

Figura A-6 – Código do subsistema "Controle"



Fonte: próprio autor

APÊNDICE B – Código de inicialização

```

//Inicialização de variáveis
clear all
clc
//Posição inicial do robô em mm
//Essas posições são referentes ao frame "Frame1"da Cintura
x_i = 0;
y_i = -120; //valor para o robô centralizado na plataforma
z_i = 269; //valor para os pés do robô começarem tocando a plataforma
//Posição inicial das juntas em rad (caso queira usar em graus, utilize a função deg2rad)
//O valor 0 para todos os ângulos das pernas faz com que ambas fiquem completamente
retas
//Perna esquerda
ang_quadril_coxa_e = 0; //positivo inclina o elo para frente
ang_coxa_canela_e = 0; //positivo inclina o elo para frente
ang_canela_suporte_e = 0; //positivo gira o elo em sentido anti-horário, olhando de frente
//Perna direita
ang_quadril_coxa_d = 0; //positivo inclina o elo para trás
ang_coxa_canela_d = 0; //positivo inclina o elo para trás
ang_canela_suporte_d = 0; //positivo gira o elo em sentido anti-horário, olhando de frente
//Contrapeso e guia
//Posição inicial do contrapeso em mm
//O valor 0 deixa o contrapeso no centro do robô, valores positivos o levam para mais
próximo da perna esquerda
pos_contrapeso = 0;
//Dimensões do bloco em mm
comprimento_bloco = 56.9;
largura_bloco = 52;
espessura_bloco = 11.5;
//Dimensões da guia em mm
comprimento_guia = 296.9;
largura_guia = 15;
espessura_guia = 12.5;
//Dimensões do contrapeso em mm
comprimento_contrapeso = 56.9;
largura_contrapeso = 52;
espessura_contrapeso = 50;

```

```
//Constante de mola torcional do pé
kt_pe = 30; //N*m
//Densidades
d_cintura = 2700; //kg/m^3 Al 6061
d_quadril = 2700; //kg/m^3
d_coxa = 2700; //kg/m^3
d_canela = 2700; //kg/m^3
d_suporte = 2700; //kg/m^3
d_pe = 2700; //kg/m^3
d_pino_longo = 2700; //kg/m^3
d_pino_curto = 2700; //kg/m^3
//Massas
m_guia = 1.2; //kg/m
m_bloco = 0.19; //kg
m_contrapeso = 6.96; //kg
m_rolamento_radial = 0.018; //kg
massa_servo = 72; //g
```

APÊNDICE C – Códigos usados no planejamento de trajetória

PERNA DIREITA - QUADRIL

```
function q_quadriL_d = q(t)
if (t>=0) && (t<=5)
q_quadriL_d = 0;
elseif(t>5) && (t<=10)
q_quadriL_d = - (pi/3125)*(t-5)^5 + (pi/250)*(t-5)^4 - (pi/75)*(t-5)^3;
elseif (t>10) && (t<=15)
q_quadriL_d = -pi/6;
elseif (t>15) && (t<=20)
q_quadriL_d = -pi/6;
elseif (t>20) && (t<=25)
q_quadriL_d = (pi/3125)*(t-20)^5 - (pi/250)*(t-20)^4 + (pi/75)*(t-20)^3 - (pi/6);
else
q_quadriL_d = 0;
end
end
```

PERNA DIREITA - COXA

```
function q_coxa_d = q(t)
if (t>=0) && (t<=5)
q_coxa_d = 0;
elseif (t>5) && (t<=10)
q_coxa_d = (pi/3125)*(t-5)^5 - (pi/250)*(t-5)^4 + (pi/75)*(t-5)^3;
elseif (t>10) && (t<=15)
q_coxa_d = pi/6;
elseif (t>15) && (t<=20)
q_coxa_d = pi/6;
elseif (t>20) && (t<=25)
q_coxa_d = - (pi/3125)*(t-20)^5 + (pi/250)*(t-20)^4 - (pi/75)*(t-20)^3 + (pi/6);
else
q_coxa_d = 0;
end
end
```

PERNA ESQUERDA - QUADRIL

```
function q_quadriL_e = q(t)
if (t>=0) && (t<=10)
```

```

q_quadriLe = 0;
elseif (t>10) && (t<=15)
q_quadriLe = - (pi/3125)*(t-10)^5 + (pi/250)*(t-10)^4 - (pi/75)*(t-10)^3;
elseif (t>15) && (t<=20)
q_quadriLe = -pi/6;
elseif (t>20) && (t<=25)
q_quadriLe = -pi/6;
else
q_quadriLe = (pi/3125)*(t-25)^5 - (pi/250)*(t-25)^4 + (pi/75)*(t-25)^3 - (pi/6);
end
end

```

PERNA ESQUERDA - COXA

```

function q_coxa_e = q(t)
if (t>=0) && (t<=10)
q_coxa_e = 0;
elseif (t>10) && (t<=15)
q_coxa_e = (pi/3125)*(t-10)^5 - (pi/250)*(t-10)^4 + (pi/75)*(t-10)^3;
elseif (t>15) && (t<=20)
q_coxa_e = pi/6;
elseif (t>20) && (t<=25)
q_coxa_e = pi/6;
else
q_coxa_e = - (pi/3125)*(t-25)^5 + (pi/250)*(t-25)^4 - (pi/75)*(t-25)^3 + (pi/6);
end
end

```