

**FACULDADE DE ENGENHARIA MECÂNICA
UNIVERSIDADE FEDERAL DE UBERLÂNDIA**

RONALDO CÉSAR DE ALMEIDA

**APLICAÇÃO DE TÉCNICAS DE CONTROLE PARA ESTABILIZAÇÃO DA
CAMINHADA DE UM ROBÔ BÍPEDE**

Uberlândia – MG

2022

RONALDO CÉSAR DE ALMEIDA

**APLICAÇÃO DE TÉCNICAS DE CONTROLE PARA ESTABILIZAÇÃO DA
CAMINHADA DE UM ROBÔ BÍPEDE**

Trabalho de Conclusão de Curso apresentada ao Programa de Graduação em Engenharia Mecatrônica da Universidade Federal de Uberlândia, como parte dos requisitos para obtenção do título de Engenheiro Mecatrônico.

Orientador: Prof. Dr. Rogério Sales Gonçalves

Uberlândia – MG

2022

RONALDO CÉSAR DE ALMEIDA

**APLICAÇÃO DE TÉCNICAS DE CONTROLE PARA ESTABILIZAÇÃO DA
CAMINHADA DE UM ROBÔ BÍPEDE**

Trabalho de Conclusão de Curso apresentada ao Programa de Graduação em Engenharia Mecatrônica da Universidade Federal de Uberlândia, como parte dos requisitos para obtenção do título de Engenheiro Mecatrônico

Orientador: Prof. Dr Rogério Sales Gonçalves

BANCA EXAMINADORA

Prof. Dr. Rogério Sales Gonçalves - UFU

Orientador

Prof. Dr.

Examinador

Prof. Dr.

Examinador

Uberlândia – MG

2022

AGRADECIMENTOS

A todo o corpo docente da Faculdade de Engenharia Mecânica, que mantém um alto nível de graduação, em especial à Engenharia Mecatrônica, referência no Brasil.

A todos os funcionários tanto da Faculdade de Engenharia Mecânica quanto da Universidade Federal de Uberlândia, que proveem a infraestrutura que torna possível a existência e manutenção dos cursos da instituição.

Ao meu orientador Prof. Rogério, pela disposição que sempre teve ao longo do trabalho, apesar dos diversos contratemplos encontrados ao longo dele.

À minha família que me deu suporte, e sendo importante para que eu mantivesse o foco necessário durante a realização da graduação, principalmente quando algo não saía como esperado.

Aos colegas de curso, em especial os que tive contato mais próximo.

A todos os membros da equipe EDROM, pelos momentos de aprendizado e também pelos momentos de glória nas diversas competições que participei.

A todos aqueles que de alguma forma contribuíram, direta ou indiretamente, para que a realização desse curso fosse possível.

ALMEIDA, R. C., APLICAÇÃO DE TÉCNICAS DE CONTROLE PARA ESTABILIZAÇÃO DA CAMINHADA DE UM ROBÔ BÍPEDE. 2022. 50p. Monografia de Conclusão de Curso, Faculdade de Engenharia Mecânica, Universidade Federal de Uberlândia, Uberlândia, MG, Brasil.

Resumo

O desenvolvimento da robótica móvel tem sido estimulado pelas inúmeras possibilidades na utilização de robôs humanoides para auxiliar o ser humano em diversas tarefas como: atividades domésticas, em ambientes hostis, e em competições, como a RoboCup.

Dessa forma, para lidar com os mais variados tipos de atividades, é de grande valia que os robôs consigam desenvolver estratégias de locomoção que sejam apropriadas de acordo o meio em que está introduzido. Neste contexto que este trabalho de conclusão de curso apresenta técnicas para a melhoria da locomoção de um robô humanoide, a fim de atender as demandas da Equipe de Desenvolvimento em Robótica Móvel (EDROM) da UFU nas mais diversas competições, aplicando estratégias de controle baseadas em um filtro de Kalman estendido (EKF), com o intuito de aumentar a estabilização da locomoção do robô.

Para isso, a modelagem da trajetória de caminhada do robô utilizou do modelo cinemático inverso, em seguida, foi realizada a compensação da inclinação do tronco do robô a partir de uma unidade de medida inercial (IMU), responsável por medir a aceleração e velocidade angular do tronco, e medir os movimentos de *roll* e *pitch* instantâneos. A partir de tais medições foi possível implementar o filtro de Kalman, a fim de diminuir os ruídos, e por sua vez aplicar o controle para a caminhada do robô. Este trabalho foi desenvolvido a nível de simulação computacional.

Palavras chave: Filtro de Kalman Estendido, Robótica Móvel, Unidade de Medição Inercial, Controle.

ALMEIDA, R. C., APPLICATION OF CONTROL TECHNIQUE FOR STABILIZATION OF ROBOT BIPEDS WALK. 2022. 50p. Course Conclusion Monograph, Faculdade de Engenharia Mecânica, Universidade Federal de Uberlândia, Uberlândia, MG, Brazil.

ABSTRACT

The development of mobile robotics is stimulated by the countless possibilities in the use of humanoid robots to assist humans in various tasks, such as: in housework, in hostile environments and in competitions like the RoboCup.

Therefore, to deal with the most varied types of activities, it is important for robots to be able to develop locomotion strategies that are appropriate according to the environment in which they find themselves in. In this context, this work presents techniques for improving the locomotion of a humanoid robot, in order to meet the demands of the Mobile Robotics Development Team (EDROM) at UFU in competitions, applying control strategies based on the Extended Kalman filter (EKF), in order to increase the robot's locomotion stabilization.

To achieve this, the robot's walking trajectory was modelled using the inverse kinematic model. The robot's trunk inclination was compensated by an inertial measurement unit (IMU), responsible for measuring the acceleration and angular velocity of the trunk and the instantaneous roll and pitch movements. With such measurements, it is possible to implement the Extended Kalman filter to reduce the noise and apply the control to the robot's walk.

Lista de Figuras

Figura 1.1 - Robô humanoide desenvolvido pela EDROM.

Figura 2.1 - Modelo do robô humanoide e seus atuadores.

Figura 2.2 - Ângulos da unidade de medição inercial.

Figura 2.3 - Rotação da perna em relação ao quadril.

Figura 2.4 - Vista lateral da perna esquerda.

Figura 2.5 - Vista frontal da perna esquerda.

Figura 3.1 - Ciclo completo da caminhada.

Figure 3.2 - Os quatro períodos dos padrões de splines.

Figure 3.3 - Modelo Cinemático e as referências da estrutura.

Figure 3.4 - Resultados nas posições dos motores da perna durante o ciclo da caminhada.

Figura 1.5 - Processo de Filtragem com o ruído da IMU de 0.0063 radianos.

Figura 4.1 - Posição inicial para mensuração do IMU.

Figura 4.2 - Coleta dos ângulos ϕ e θ (IMU)

Figura 4.3 - Ângulo do torso ϕ e θ após a ação do EKF.

Figura 4.4 - Ângulo ϕ antes e após passarem pela filtragem.

Figura 4.5 - Ângulo θ antes e após passarem pela filtragem.

Figura 5.1 - Representação do deslocamento do torso em ϕ .

Figura 5.2 - Representação do deslocamento do torso em θ .

Lista de Tabelas

Tabela 1 - Tabela de Dimensões usadas na Cinemática Inversa.

Tabela 2 - Parâmetros para caminhada.

Tabela 3 - Valores dos ganhos K_p e K_v para os roll e pitch.

LISTA DE SIGLAS E ABREVIATURAS

DKF	<i>Discrete Kalman Filter</i>
EDROM	Equipe de Desenvolvimento em Robótica Móvel
EKF	<i>Extended Kalman Filter</i>
EPFL	Instituto Federal de tecnologia da Suíça
FEMEC	Faculdade de Engenharia Mecânica
FPGA	<i>Field-Programmable Gate Array</i>
IMU	<i>Inertial Measurement Unit</i>
HDMI	<i>High-Definition Multimedia Interface</i>
LAMI	Laboratório de <i>Micro Informatic</i>
NASA	<i>National Aeronautics and Space Administration</i>
OP3	<i>Open Platform 3</i>
PC	<i>Personal Computer</i>
ROS	<i>Robot Operating System</i>
UFU	Universidade Federal de Uberlândia
USB	<i>Universal Serial Bus</i>
ZMP	<i>Zero Moment Point</i>

Sumário

INTRODUÇÃO	11
1.1 OBJETIVOS	12
1.2 OBJETIVOS ESPECÍFICOS.....	12
1.3 JUSTIFICATIVA.....	13
FUNDAMENTAÇÃO TEÓRICA.....	13
2.1. ROBOTIS OP3	13
2.2. UNIDADE DE MEDIÇÃO INERCIAL.....	14
2.3. IMPLEMENTAÇÃO DA CAMINHADA.....	15
2.3.1 MODELO CINEMÁTICO INVERSO	15
2.4. FILTRO DE KALMAN DISCRETO	19
2.5. FILTRO DE KALMAN ESTENDIDO.....	21
2.6. COMPENSAÇÃO DA INCLINAÇÃO DO TRONCO	22
2.7. CONTROLADOR PROPORCIONAL	24
2.8. CONTROLADOR DE INCLINAÇÃO DO TRONCO.....	25
METODOLÓGIA	27
3.1. IMPLEMENTAÇÃO DO FILTRO DE KALMAN ESTENDIDO	27
3.2. WEBOTS.....	27
3.3. SOFTWARE DESENVOLVIDO.....	28
3.4. DESENVOLVIMENTO DA CAMINHADA	29
RESULTADO	40
4.1. Validação do Filtro de Kalman Estendido.....	40
4.2. Validação do Controlador de Caminhada	44
CONCLUSÃO	47
TRABALHOS FUTUROS	49

CAPÍTULO I

INTRODUÇÃO

Segundo SANTOS (2010), a origem da palavra robô, de origem tcheca, significa “trabalho”, e é atribuído ao escritor Karel Capeck, em sua peça de teatro Rossum’s Universal Robots, já o termo robótica foi introduzido pela primeira vez por Isaac Asimov, na sua obra Eu, Robô, do ano de 1941, e se refere à uma maneira de representar a ciência dedicada ao estudo dos robôs.

Para MCKERROW (1986), a robótica é uma ciência que envolve conhecimentos de várias áreas do saber com o objetivo de desenvolver e integrar técnicas e algoritmos para a criação, controle e programação de robôs, com vista à sua aplicabilidade nas mais diversas situações, substituindo a ação humana

Neste trabalho, será abordado os robôs móveis humanoides. O desenvolvimento de locomoção mais eficientes para os robôs móveis têm sido o foco de muitos estudos ao redor do mundo, dessa forma, devido à sua versatilidade, a pesquisa na área de robôs humanoides vem sendo explorada nas últimas décadas, sendo que as aplicações variam de desenvolvimento de exoesqueletos robóticos à robôs que podem auxiliar no resgate de vítimas de tragédias, WESTERVELT et al. (2007).

De acordo com MOUSSA (2013), a aparência do robô humanoide é semelhante ao corpo humano e, geralmente, apresentam tronco, cabeça, dois braços e duas pernas, e são providos de sensores que auxiliam na percepção do ambiente externo, tais como: câmera, sensores inerciais, sensor de toque dentre outros. Tendo isso em vista, a fim de desenvolver a robótica no mundo, em 1997, foi criada a competição RoboCup, que contempla diversas categorias, dentre elas o futebol de robô humanoides (“A Brief History of RoboCup”, [S.d.]). Nessa categoria, os robôs devem apresentar características de locomoção bípede, para lidarem com os diferentes desafios pré-determinados, como: desenvolver uma caminhada estável e omnidirecional, chutar uma bola para marcar gols, localizar a bola e o gol no campo de futebol e desenvolver estratégias de jogo, VENÂNCIO (2018).

Com o passar dos anos, a RoboCup tem aumentado o nível dos problemas a serem desenvolvidos pelas as equipes, desde o aumento do peso do robô, mudança na bola utilizada, bem como, a mudança do material do campo, que passou de um carpete liso para grama sintética, aumentando a dificuldade de equilíbrio durante a caminhada. Dessa forma, este trabalho é de interesse da Equipe de Desenvolvimento em Robótica Móvel (EDROM) da UFU, com o intuito de desenvolver um método computacional para o controle estável da caminhada do robô durante as competições, utilizando um filtro de Kalman estendido (EKF).



Figura 2.1 – Robô humanoide.

Fonte – Acervo do Autor.

1.1 OBJETIVOS

Como objetivo geral, este TCC apresenta o desenvolvimento e avaliação de uma arquitetura em software do algoritmo de Filtro de Kalman Estendido (EKF) para o problema de caminhada de robôs humanoides, considerando um sistema multissensorial.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos deste trabalho são:

- Obtenção do modelo do sistema correspondente à plataforma robótica móvel;
- Obtenção do modelo de medição tendo em conta a disposição dos sensores no robô;

- Obtenção de todas as equações do algoritmo EKF, baseadas no modelo de sistema não linear ruidoso e medições propostas para um ambiente específico;
- Implementação das equações que formam parte do algoritmo EKF em software para efeito de validação dos modelos estáveis de caminhada.

1.3 JUSTIFICATIVA

A pesquisa em robôs humanoides é de interesse para a comunidade científica, pois esses robôs podem ser utilizados nos ambientes humanos. Muitos laboratórios do mundo que desenvolvem pesquisas em robótica não têm acesso a robôs humanoides reais. Porém é muito comum encontrar robôs humanoides menos robustos, desenvolvidos por equipes universitárias, para competições como a RoboCup.

Tal monografia se justifica a partir das demandas da equipe EDROM. Tendo em vista a necessidade de uma melhor performance dos robôs humanoides nas competições, a partir da implementação de um controle baseado em um filtro de Kalman estendido, a fim de obter uma maior precisão para a caminhada do robô no campo de futebol.

CAPÍTULO II

FUNDAMENTAÇÃO TEÓRICA

2.1. ROBOTIS OP3

O robô OP3 é um robô humanoide, desenvolvido em colaboração com a Universidade da Pensilvânia, utilizado em pesquisas e competições. Além disso, OP significa “*Open Platform*”, o que significa que os designs do robô estão disponíveis para download no site do projeto.

O robô tem 20 graus de liberdade, 510 mm de altura e pesa 3,5 kg. Estas 20 articulações estão distribuídas como: 2 no pescoço, 3 por braço e 6 por perna, conforme mostrado na Fig. 2.1.

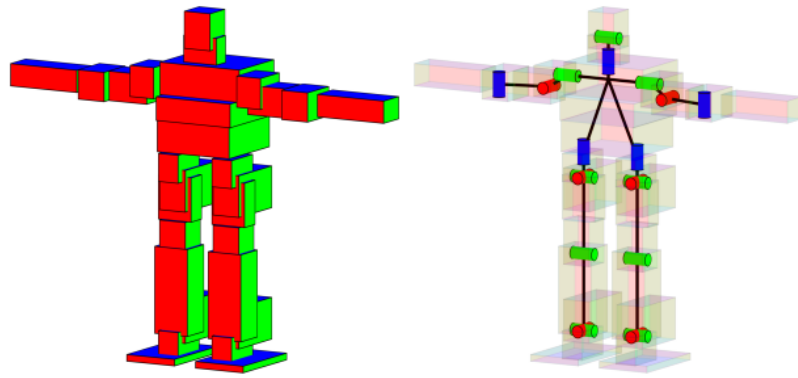


Figura 3.1 – Modelo do robô humanoide e seus atuadores.

Fonte – Máximo (2017).

Apesar de pequeno, possui um PC integrado com um processador Intel Core i3 dual core de 2133 MHz para processamento de alto nível. Além disso, um subcontrolador OpenCR é usado para fazer interface com sensores e atuadores. A articulação é acionada por um servomotor DYNAMIXEL 2.0. Em relação ao sensoriamento, o robô tem uma câmera USB e microfones na cabeça, uma unidade de medição inercial e um giroscópio no seu torso e encoders em cada servomotor. O PC embutido também fornece conectividade: USB, Ethernet, WiFi e HDMI.

2.2. UNIDADE DE MEDIÇÃO INERCIAL

Segundo MOURA (2013), uma unidade de medição inercial é um dispositivo eletrônico que mede e relata a força específica de um corpo, taxa angular e a orientação do corpo, usando uma combinação de acelerômetros, giroscópios e magnetômetros. IMUs são normalmente usados para manobrar aeronaves, incluindo veículos aéreos não tripulados, entre muitos outros, e espaçonaves, incluindo satélites.

Uma unidade de medição inercial funciona detectando a aceleração linear usando um ou mais acelerômetros e a taxa de rotação usando um ou mais giroscópios. Alguns também incluem um magnetômetro que é comumente usada como uma referência de direção. As configurações típicas contêm um acelerômetro, giroscópio e magnetômetro por eixo para cada um dos três eixos principais: inclinação, rotação e guinada Fig. 2.2.

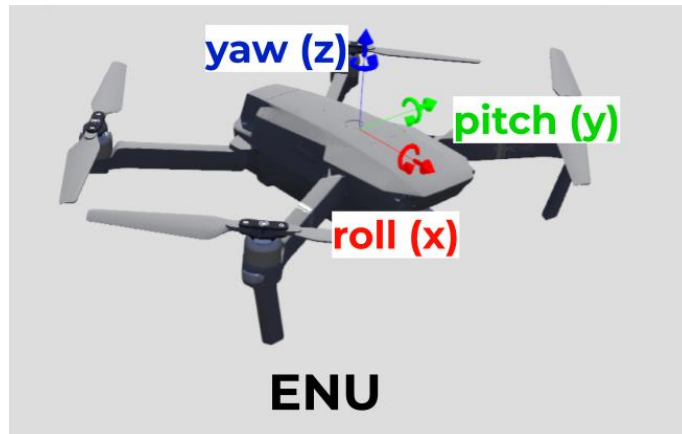


Figura 2.2 – Ângulos da unidade de medição inercial.

Fonte – Cyberbotics (2022).

2.3. IMPLEMENTAÇÃO DA CAMINHADA

Uma abordagem comum no projeto de robôs humanoides é implementar cada junta usando um servomotor com controle de posição. O ROBOTIS OP3, que é a plataforma robótica utilizada neste trabalho, segue essa abordagem.

Neste caso, as posições das articulações para atingir a posição do centro de massa desejada são calculadas e alimentam os servomotores. Apesar do cálculo exato das posições das juntas, o erro no centro de massa acontece devido a distúrbios, dinâmica dos servomotores e erros de modelagem. Portanto, compensações de *feedforward* e *feedback* são usadas para melhorar o comportamento de caminhada.

2.3.1 MODELO CINEMÁTICO INVERSO

Descrever trajetórias usando as posições dos pés é muito mais intuitivo do que com ângulos das articulações. No entanto, em robôs humanoides controlados por posição, como ROBOTIS OP3, o controlador só pode comandar os ângulos das juntas. Portanto, nesta seção, derivamos equações que mapeiam uma posição desejada do pé para os ângulos das articulações necessários.

Para o modelo do robô bípede baseado em ROBOTIS OP3, é possível obter uma solução analítica utilizando a abordagem inspirada em GRAF (2009), levando em consideração as particularidades próprias relacionadas ao robô em estudo.

As entradas para o modelo cinemático inverso são as poses desejadas dos pés esquerdo e direito, dadas pelas equações 2.1 e 2.2.

$$w_l = [x_l, y_l, z_l, \psi_l]^T \quad (2.1)$$

$$w_r = [x_r, y_r, z_r, \psi_r]^T \quad (2.2)$$

Onde, x_l, y_l, z_l são as coordenadas lineares da pose desejada do pé esquerdo e ψ_l a coordenada angular, descritos no sistema de coordenadas do tronco.

De agora em diante, será trabalhado apenas as equações para a perna esquerda, sendo as equações da perna direita encontradas da mesma forma. Primeiramente, de acordo com a equação 2.3, subtrai-se a translação entre o tronco e o quadril esquerdo (ponto de interseção das articulações do quadril esquerdo) da pose do pé desejada:

$$w_{l,h2f} = w_l - \begin{bmatrix} x_{l,t2h} \\ y_{l,t2h} \\ z_{l,t2h} \\ 0 \end{bmatrix} \quad (2.3)$$

Onde, $[x_{l,t2h} \ y_{l,t2h} \ z_{l,t2h}]^T$ é o vetor de translação entre o tronco e o quadril esquerdo. Já a elevação do pé é simplesmente alcançada pela rotação da articulação do quadril. Dessa forma,

$$\theta_{hy} = \psi_l \quad (2.4)$$

Tal rotação do atuador do quadril movimenta toda a perna, como pode ser vista na Fig. 2.3.

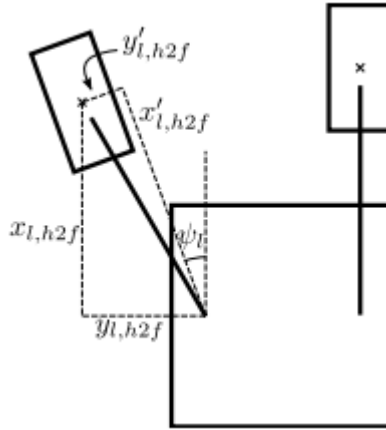


Figura 2.3. – Rotação da perna inércia em relação ao quadril.

Fonte – Máximo (2017).

Dessa forma, as translações $x'_{l,h2f}$ e $y'_{l,h2f}$ do pé, devido a rotação do atuador do quadril. Logo,

$$x'_{l,h2f} = x_{l,h2f} \cos \psi_l + y_{l,h2f} \sin \psi_l \quad (2.5)$$

$$y'_{l,h2f} = -x_{l,h2f} \sin \psi_l + y_{l,h2f} \cos \psi_l \quad (2.6)$$

Então, obtém-se as coordenadas do tornozelo em relação ao quadril esquerdo, 2.7, 2.8 e 2.9, subtraindo a translação entre o tornozelo esquerdo (intersecção entre as articulações do tornozelo esquerdo) e o pé.

$$x'' = x'_{l,h2f} - x_{l,a2f} \quad (2.7)$$

$$y'' = y'_{l,h2f} - y_{l,a2f} \quad (2.8)$$

$$z'' = z - z_{l,a2f} \quad (2.9)$$

Onde, $[x_{l,a2f} \ y_{l,a2f} \ z_{l,a2f}]^T$ é o vetor de translação entre o tornozelo esquerdo e o pé esquerdo. Na Fig. 2.4. uma vista lateral da perna esquerda é mostrada,

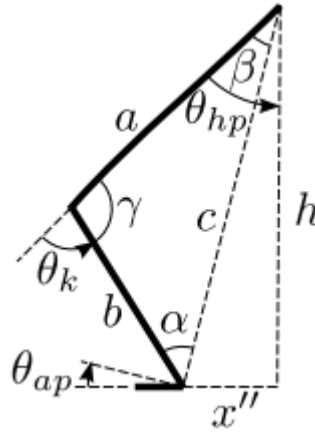


Figura 2.4. – Vista lateral da perna esquerda.

Fonte – Máximo (2017).

Dessa forma, aplicando a lei dos cossenos, chega-se as equações 2.10, 2.11 e 2.12.

$$\alpha = \text{acos} \left(\frac{b^2 + c^2 - a^2}{2bc} \right) \quad (2.10)$$

$$\beta = \text{acos} \left(\frac{a^2 + c^2 - b^2}{2ac} \right) \quad (2.11)$$

$$\gamma = \text{acos} \left(\frac{a^2 + b^2 - c^2}{2ab} \right) \quad (2.12)$$

Onde, a e b são o comprimento das partes superior e inferior da perna, respectivamente. E o comprimento c é a norma do vetor de translação do quadril para o tornozelo. Dessa forma.

$$c = \sqrt{(x'')^2 + (y'')^2 + (z'')^2} \quad (2.13)$$

E,

$$h = \sqrt{(y'')^2 + (z'')^2} \quad (2.14)$$

Logo,

$$c = \sqrt{(x'')^2 + h^2} \quad (2.15)$$

A partir da Fig. 2.4, podemos derivar os ângulos das juntas de passo:

$$\theta_{hp} = -(\text{atan2}(x'', h) + \beta) \quad (2.16)$$

$$\theta_k = \pi - \gamma \quad (2.17)$$

$$\theta_{ap} = \text{atan2}(x'', h) + \alpha \quad (2.18)$$

Onde, θ_{hp} , θ_k e θ_{ap} são os ângulos das articulações do quadril, joelho e tornozelo. A função $\text{atan2}(x, y)$ é o arco tangente de x/y . Agora, da Fig. 2.5, podemos deduzir as seguintes fórmulas para os ângulos das juntas:

$$\theta_{hp} = \text{atan2}(y'', z'') \quad (2.19)$$

$$\theta_{fr} = -\theta_{hr} \quad (2.20)$$

Onde θ_{hr} e θ_{fr} são os ângulos das articulações do quadril e do pé.

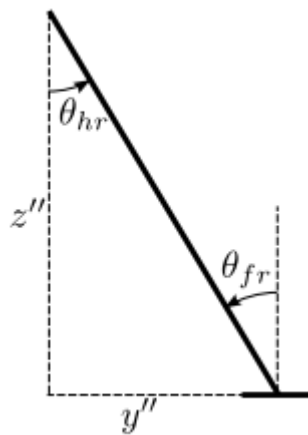


Figura 2.5. – Vista frontal da perna esquerda.

Fonte – Máximo (2017).

2.4. FILTRO DE KALMAN DISCRETO

Aiube et al. (2006) definem filtro de Kalman como um conjunto de equações matemáticas que constitui um processo recursivo eficiente de estimação, uma vez que o erro quadrático de estimação é minimizado. Através da observação da variável denominada variável de observação outra variável, não observável, denominada variável de estado pode ser estimada.

A modelagem do Filtro de Kalman Discreto pressupõe que o sistema seja linear e descrito pelo modelo de equações 2.21:

$$\begin{cases} s_t = A_t s_{t-1} + B_t u_{t-1} + \gamma_t \\ z_t = C_t s_t + \delta_t \end{cases} \quad (2.21)$$

Onde,

$s \in R^n$ é o vetor de estados;

$u \in R^l$ é o vetor das entradas de controle;

$z \in R^m$ é o vetor de medições;

A a matriz de transição de estados;

B a matriz de coeficientes de entrada;

C a matriz de observação;

γ vetor de ruídos do processo;

δ vetor de erros de medição.

Os índices t e $t - 1$ representam os instantes de tempo atual e anterior respectivamente.

O filtro opera em modo de predição-atualização levando em consideração as propriedades estatísticas do ruído. Um modelo interno do sistema é usado para atualização e um esquema de realimentação realiza as medições. As etapas de predição e atualização para o DKF podem ser descritas pelos sistemas lineares 2.22 e 2.23.

$$\begin{cases} \bar{\mu}_t = A_t \mu_{t-1} + B_t u_{t-1} \\ \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t \end{cases} \quad (2.22)$$

E,

$$\begin{cases} K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_{t-1} C_t^T + Q_t)^{-1} \\ \mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t) \\ \Sigma_t = (I - K_t C_t) \bar{\Sigma}_t \end{cases} \quad (2.23)$$

O filtro de Kalman representa o vetor de estados s_t no tempo t por sua média μ_t e covariância Σ_t . As matrizes R , $n \times n$, e Q , $l \times l$, são as matrizes de covariância dos ruídos de processo γ e medição δ respectivamente e a matriz K , $n \times m$, representa o ganho do sistema.

A partir da especificação matemática apresentada, um algoritmo para o DKF é apresentado a seguir no quadro 1. Maior detalhamento sobre o algoritmo bem como seu desenvolvimento matemático são descritos por Thrun (2005).

Algoritmo 1 *Filtro de Kalman Discreto.*

- 01:** $\text{FKD}(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$
02: $\bar{\mu}_t = \mathbf{A}_t \mu_{t-1} + \mathbf{B}_t \mathbf{u}_{t-1}$
03: $\bar{\Sigma}_t = \mathbf{A}_t \Sigma_{t-1} \mathbf{A}_t^T + \mathbf{R}_t$
04: $\mathbf{K}_t = \bar{\Sigma}_t \mathbf{C}_t^T (\mathbf{C}_t \bar{\Sigma}_t \mathbf{C}_t^T + \mathbf{Q}_t)^{-1}$
05: $\mu_t = \bar{\mu}_t + \mathbf{K}_t (z_t - \mathbf{C}_t \bar{\mu}_t)$
06: $\Sigma_t = (\mathbf{I} - \mathbf{K}_t \mathbf{C}_t) \bar{\Sigma}_t$
07: *return*(μ_t, Σ_t)
-

Uma das derivações do filtro de Kalman aplicado a sistemas não lineares é o Filtro de Kalman Estendido.

2.5. FILTRO DE KALMAN ESTENDIDO

A ideia do EKF é linearizar as funções em torno da estimação corrente usando as derivadas parciais do processo e das funções de medição para calcular as estimações, mesmo em face a relações não-lineares.

O modelo do sistema para o FKE é dado pelo sistema 2.24:

$$\begin{cases} s_t = g(u_{t-1}, s_{t-1}) + \gamma_t \\ z_t = h(s_t) + \delta_t \end{cases} \quad (2.24)$$

Onde, $g(u_{t-1}, s_{t-1})$ é uma função não-linear que representa o modelo do sistema, e $h(s_t)$ é uma função não-linear que representa o modelo das medições. Suas etapas de predição e atualização podem ser obtidas pelos sistemas de equações 2.25 e 2.26.

$$\begin{cases} \bar{\mu}_t = g(u_{t-1}, \mu_{t-1}) \\ \bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t \end{cases} \quad (2.25)$$

E,

$$\begin{cases} \mathbf{K}_t = \bar{\Sigma}_t \mathbf{H}_t^T (\mathbf{H}_t \bar{\Sigma}_t \mathbf{H}_t^T + \mathbf{Q}_t)^{-1} \\ \mu_t = \bar{\mu}_t + \mathbf{K}_t (z_t - h(\bar{\mu}_t)) \\ \Sigma_t = (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \bar{\Sigma}_t \end{cases} \quad (2.26)$$

A matriz G , $n \times n$, é o jacobiano que lineariza o modelo, e H , $l \times n$, o jacobiano que lineariza o vetor de medições. Tais matrizes são definidas pelas derivadas, 2.27 e 2.28.

$$G_t = \frac{\partial g(u_{t-1}, s_{t-1})}{\partial s_{t-1}} \quad (2.27)$$

E,

$$H_t = \frac{\partial h(s_t)}{\partial s_t} \quad (2.28)$$

Um algoritmo para o EKF é apresentado no quadro 2.

Algoritmo 2 *Filtro de Kalman Extendido.*

01: FKE($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$)
02: $\bar{\mu}_t = g(\mathbf{u}_{t-1}, \mu_{t-1})$
03: $\mathbf{G}_t = \frac{\partial g(\mathbf{u}_{t-1}, s_{t-1})}{\partial s_{t-1}}$
04: $\bar{\Sigma}_t = \mathbf{G}_t \Sigma_{t-1} \mathbf{G}_t^T + \mathbf{R}_t$
05: $\mathbf{H}_t = \frac{\partial h(s_t)}{\partial s_t}$
06: $\mathbf{K}_t = \bar{\Sigma}_t \mathbf{H}_t^T (\mathbf{H}_t \bar{\Sigma}_t \mathbf{H}_t^T + \mathbf{Q}_t)^{-1}$
07: $\mu_t = \bar{\mu}_t + \mathbf{K}_t (z_t - h(\bar{\mu}_t))$
08: $\Sigma_t = (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \bar{\Sigma}_t$
09: *return*(μ_t, Σ_t)

Comparando os algoritmos apresentados nos quadros 1 e 2 percebe-se duas diferenças:

- linha 3 do quadro 1 com linha 4 do quadro 2;
- linha 4 do quadro 1 com linha 6 do quadro 2;

Isso se deve ao fato de o FKE utilizar generalizações não-lineares das matrizes utilizadas no FKD. No filtro estendido o jacobiano G_t corresponde as matrizes A_t e B_t e o jacobiano H_t corresponde à matriz C_t .

2.6. COMPENSAÇÃO DA INCLINAÇÃO DO TRONCO

Primeiro, precisamos estimar a orientação do tronco. Assumimos que o tronco do robô possui uma unidade de medida inercial (IMU) embutida, que é capaz de medir

as acelerações a_x, a_y e a_z e as velocidades angulares do tronco p, q e r , nos quais são medido para o sistema de orientação do torso x, y e z respectivamente. Integrando a velocidade angular, os ângulos de orientação do torso ϕ, θ e ψ pode ser estimada. Por outro lado, o *roll* e *pitch* instantâneos sem desvios angulares podem ser obtidos usando o acelerômetro, mas esta medida tende a ser muito ruidosas, especialmente se o tronco estiver acelerando, o que acontece com frequência durante a caminhada. Portanto, um observador pode ser usado para fundir ambas as informações e produzir uma estimativa sem desvio da orientação do tronco, MAXIMO (2017).

Usando ângulos de Euler, a equação de cinemática do corpo rígido para rotação é dada por 2.29.

$$f(x, u) = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} p + (q \sin \phi + r \cos \phi) \tan \theta \\ q \cos \phi - r \sin \phi \\ (q \sin \phi + r \cos \phi) \sec \theta \end{bmatrix} \quad (2.29)$$

Podemos discretizar a equação 2.29 usando a aproximação de Euler, chegamos a equação 2.30.

$$x[k+1] = f_k(x_k, u_k) = \begin{bmatrix} \phi[k] + T[p + (q \sin \phi [k] + r \cos \phi [k]) \tan \theta [k]] \\ \theta[k] + T(q \cos \phi [k] - r \sin \phi [k]) \\ \psi[k] + T[(q \sin \phi [k] + r \cos \phi [k]) \sec \theta [k]] \end{bmatrix} \quad (2.30)$$

Dessa forma podemos calcular os Jacobianos, 2.31 e 2.32.

$$F_k = \frac{\partial f_k}{\partial x} = \begin{bmatrix} 1 + T(q \cos \phi + r \sin \phi) \tan \theta & T(q \sin \phi + r \cos \phi) \sec^2 \theta & 0 \\ -T(q \sin \phi - r \cos \phi) & 1 & 0 \\ T(q \cos \phi - r \sin \phi) \sec \theta & T(q \sin \phi + r \cos \phi) \tan \theta \sec \theta & 1 \end{bmatrix} \quad (2.31)$$

E,

$$L_k = \frac{\partial f_k}{\partial u} = \begin{bmatrix} T & T \sin \phi \tan \theta & T \cos \phi \tan \theta \\ 0 & T \cos \phi & -T \sin \phi \\ 0 & T \sin \phi \sec \theta & T \cos \phi \sec \theta \end{bmatrix} \quad (2.32)$$

Se nenhuma aceleração está sendo imposta ao torso do robô, o acelerômetro mede a aceleração devido à gravidade da Terra. Portanto, se o tronco estiver inclinado, a aceleração da gravidade será projetada em seus eixos como:

$$h_k(x) = \begin{bmatrix} -g \sin \theta \\ g \sin \phi \cos \theta \\ g \cos \phi \cos \theta \end{bmatrix} \quad (2.33)$$

Para verificar a equação anterior, a matriz de rotação dados os ângulos de Euler é:

$$R = R_z(\psi)R_y(\theta)R_x(\phi) = \begin{bmatrix} \cos \theta \cos \psi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\ \cos \theta \sin \psi & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi \\ -\sin \theta & \sin \psi \cos \theta & \cos \psi \cos \theta \end{bmatrix} \quad (2.34)$$

Dados os ângulos de Euler do torso, a equação contém uma transformação do sistema de coordenadas para o sistema de coordenadas global, ou seja, $T_{torso}^{global} = R$. Podemos calcular a aceleração medida pelo acelerômetro, transformando a aceleração adequada no sistema de coordenadas global para o sistema de coordenadas do torso, MAXIMO (2017).

$$a_{IMU}^{torso} = T_{IMU}^{torso} a_{IMU}^{global} = R^T \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (2.35)$$

$$a_{IMU} = \begin{bmatrix} -g \sin \theta \\ g \sin \phi \cos \theta \\ g \cos \phi \cos \theta \end{bmatrix} \quad (2.36)$$

Com isso, temos o jacobiano, dada pela equação 2.37:

$$H_k = \frac{\partial h_k}{\partial x} = \begin{bmatrix} 0 & -g \cos \theta & 0 \\ g \cos \phi \cos \theta & -g \sin \phi \sin \theta & 0 \\ -g \sin \phi \cos \theta & -g \cos \phi \sin \theta & 0 \end{bmatrix} \quad (2.37)$$

É claro que o torso inevitavelmente acelera durante a caminhada, para evitar alimentar o observador com medidas erradas, só executamos a etapa de filtragem se a magnitude do vetor estiver dentro do intervalo, $g_{\min} \leq g \leq g_{\max}$.

2.7. CONTROLADOR PROPORCIONAL

O controle proporcional é uma tecnologia de sistema de controle baseada em uma resposta proporcional à diferença entre o que é definido como uma variável de processo desejada (ou *set point*) e o valor atual da variável.

No algoritmo de controle proporcional, a saída do controlador é proporcional ao sinal de erro, que é a diferença entre o *setpoint* e a variável de processo. Em outras

palavras, a saída de um controlador proporcional é o produto da multiplicação do sinal de erro pelo ganho proporcional. Isso pode ser expresso matematicamente pela equação 2.38.

$$P_{out} = K_p e(t) + p_o \quad (2.38)$$

Onde,

P_{out} é a saída do controlador proporcional;

K_p é o ganho proporcional;

$e(t)$ é o erro do processo;

p_o saída do controlador com erro zero.

2.8. CONTROLADOR DE INCLINAÇÃO DO TRONCO

Usamos controladores proporcionais desacoplados para os canais *roll* e *pitch*, as medições de velocidade angular são usadas para evitar a amplificação do ruído. Além disso, as seguintes juntas são consideradas neste controlador: *hip-pitch*, *knee-pitch*, *ankle-pitch*, *hip-roll* e *ankle-roll*. Como o robô é controlado pela posição, esses controladores de orientação do tronco calculam os comandos conjuntos dos deslocamentos:

$$\Delta\theta_{hr,o} = -K_{p,hr}(\phi_d - \phi) - K_{v,hr}p \quad (2.39)$$

$$\Delta\theta_{ar,o} = -K_{p,ar}(\phi_d - \phi) - K_{v,ar}p \quad (2.40)$$

$$\Delta\theta_{hp,o} = -K_{p,hp}(\theta_d - \theta) - K_{v,hp}q \quad (2.41)$$

$$\Delta\theta_{kp,o} = -K_{p,kp}(\theta_d - \theta) - K_{v,kp}q \quad (2.42)$$

$$\Delta\theta_{ap,o} = -K_{p,ap}(\theta_d - \theta) - K_{v,ap}q \quad (2.43)$$

Para isso, modelamos o robô em suporte único como dois manipuladores desacoplados. Um para o plano sagital (*pitch*) e outro para o plano coronal (*roll*). As equações dinâmicas dos manipuladores de três e dois links são determinadas usando da Mecânica Lagrangiana, uma abordagem comum na literatura de manipuladores robóticos, como descrito por CRAIG (2017). No plano sagital,

consideramos um modelo de três elos com as articulações *hip-pitch*, *knee-pitch* e *ankle-pitch*. Para o plano coronal, usamos um modelo de dois links com articulações *hip-roll* e *anke-roll*. Nestes modelos, a parte superior do corpo e o pé oscilante são agrupados como um único *link*. Em seguida, esses modelos são linearizados em torno da caminhada nominal (a postura do robô quando em fase de apoio duplo com velocidade de caminhada zero).

CAPÍTULO III

METODOLÓGIA

3.1. IMPLEMENTAÇÃO DO FILTRO DE KALMAN ESTENDIDO

Este capítulo detalha os procedimentos realizados para implementar o filtro de kalman estendido em um robô humanoide. Pode-se dividir o projeto em 3 etapas fundamentais que serão explicadas nos tópicos seguintes:

- Ambiente de simulação Webots;
- Desenvolvimento da Caminhada;
- Aplicação do Filtro;

3.2. WEBOTS

O Webots é um simulador 3D *open source* de múltiplas plataformas de aplicação de alta fidelidade que é capaz de simular vários tipos de ambientes. O Webots foi originalmente desenvolvido como uma ferramenta de pesquisa para investigar algoritmos de controle variados em robô móvel no Laboratório de *Micro Informatic* (LAMI) do Instituto Federal de tecnologia da Suíça (EPFL). A escolha desse simulador foi devido a sua larga biblioteca de ativos que inclui robôs, sensores, atuadores, objetos e materiais e um guia bem documentado o que oferece vários exemplos de aplicação para as suas ferramentas, além da sua comunidade ativa onde pode se recorrer para encontrar suporte para retirar possíveis dúvidas em diferentes plataformas. Nesse trabalho foi usado a versão R2022a;

Devido a sua integração com o *open source Robot Operating System* (ROS), o Webots se torna um simulador bastante flexível para o desenvolvimento de códigos. ROS é um kit de desenvolvimento de sistema (SDK), que providencia blocos de comunicação capaz de comunicar com qualquer software que interagem com o mesmo *hardware* através de um sistema de compartilhamento de mensagens e tópicos entre os softwares chamado *middleware*, concedendo ao projeto uma solução simples de manter, contribuir e reusar. O ROS oferece também um conjunto de ferramentas que facilita o desenvolvimento do software, de descobrimento de erros e visualização de gráficos.

A decisão da utilização da simulação foi dada pela necessidade de produzir um ambiente fácil para testar sem a necessidade de ter uma dependência de um modelo físico.

Tal como o objetivo do manuscrito é de desenvolver técnicas de controle de caminhada para ajudar a EDROM com o desenvolvimento da equipe nas competições de robótica, foi utilizado para o desenvolvimento desse trabalho o robô Robotis OP3 (2.1. ROBOTIS OP3), onde é um modelo já incluso no Webots, e foi escolhido por ter estrutura, tamanho e funcionalidades similar aos robôs de competição da categoria *small size* da liga de futebol de robô da competição mundial de robótica (RoboCup).

3.3. SOFTWARE DESENVOLVIDO

Os softwares e métodos apresentado anteriormente foram aplicados no desenvolvimento desse trabalho utilizando as linguagens de programação C++ e Python.

Pode se dividir o presente trabalho em 3 partes fundamentais, que foram desenvolvidos softwares para a implementação das mesmas, sendo essas partes o desenvolvimento da caminhada, a comunicação entre a simulação e a caminhada e o desenvolvimento do filtro de kalman estendido. Para a realização da caminhada foi utilizado C++ junto com a biblioteca IKWalk, que será mais detalhada na seção abaixo (3.4. Desenvolvimento da caminhada). Para o a realização do filtro de kalman estendido foi utilizado o Python, com a implementação da biblioteca Numpy que facilita os cálculos de matrizes. Com tudo, a comunicação entre os métodos e firmware, onde se passam todos os parâmetros para o Webots e entre eles foi realizada pelo ROS.

3.4. DESENVOLVIMENTO DA CAMINHADA

Nessa sessão apresenta-se os conceitos utilizados para a realização da caminhada de um robô humanoide. A caminhada para um robô humanoide bípede é um ciclo que passa por 4 fases que são alternadas entre suporte duplo e suporte único, onde deve haver ao menos um pé no chão em todos os momentos, Fig. 3.1. A fase de suporte único é onde quando existe apenas um pé no chão, enquanto a fase de suporte duplo os dois pés permanecem no chão. O movimento de caminhada começa com a fase do suporte duplo e alternado para a fase de suporte único, onde apenas um pé suporta o corpo enquanto o outro está em fase de transição.

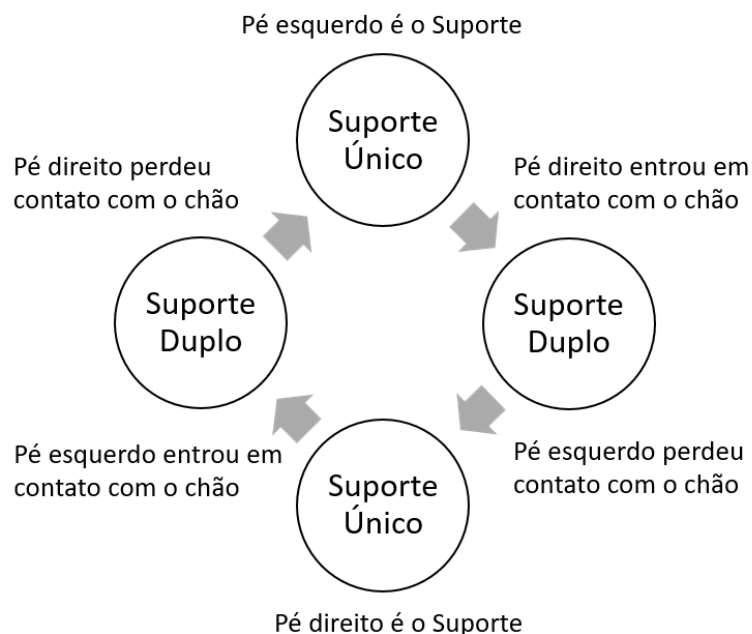


Figura 3.1 – Ciclo completa da caminhada.

Fonte – Acervo do Autor.

A característica da dinâmica do robô se difere consideravelmente durante o suporte único e o suporte duplo. Durante a fase de suporte duplo, ambos os pés estão no chão, portanto eles são impedidos de mover sem deslizar, podemos dizer então que nessa fase o robô está em cadeia cinemática fechada. Durante a fase de suporte simples, o pé que está em transição está fora do chão e livre para mover, assim o robô se encontra em cadeia cinemática aberta (MAXIMUS, 2022). As fases de geração da caminhada têm grande interferência na estabilização do robô humanoide, onde na fase de suporte duplo o robô está mais estável, uma vez que durante o movimento de

transição para a fase de suporte único, o robô desloca o seu centro de massa não sendo hábil de manter o equilíbrio sem o uso de um método de controle.

Caminhada de um robô humanoide bípede é um assunto bastante estudado ainda nos tempos atuais e onde ainda é um desafio para muitas equipes de robóticas que desenvolvem projetos baseados em caminhada de um robô humanoide. Nos dias atuais, tem se vários métodos e abordagem para resolver o problema de caminha bípede, sendo uma das mais utilizadas o *Zero Moment Point (ZMP)*.

O ZMP é uma técnica de caminhada que leva em consideração a posição dos pés e do centro de massa, onde o centro de massa do robô deve estar dentro do polígono de suporte, que é uma área calculada definida pelos pontos de contatos do pé no solo, para garantir equilíbrio durante a caminhada. ZMP é simplesmente um ponto de reação no chão dentro desse polígono de suporte que gera estabilidade durante as fases de caminhada, levando em consideração os efeitos de dinâmica, como por exemplo a aceleração angular do torso, QUENTI (2015).

No entanto o ZMP é um método complexo de se implementar, onde envolve cálculos de matrizes e estruturas cinemáticas complexas. Por isso, nesse trabalho foi usado um modelo para calcular os passos mais simples que oferece uma caminhada multidirecional e estável, o Modelo Cinemático Inverso, no qual os conceitos foram trabalhados na sessão 2.3.1.

A implementação do modelo foi realizada usando a biblioteca IKWalk e C++, essa biblioteca implementa o modelo cinemático inverso para gerar as posições alvo do servomotor em cadeia cinemática aberta. O cálculo das posições alvo dos servomotores é realizada através de splines periódicos que geram o principal padrão oscilatórios que definem a trajetória cartesiana de tronco e pés, QUENTI (2015).

A biblioteca IKWalk foi utilizada na competição da Robocup no ano de 2015 na china pela equipe da Rhoban na categoria Small Size, permitindo a locomoção bípede de um robô humanoide em grama artificial, QUENTI (2015).

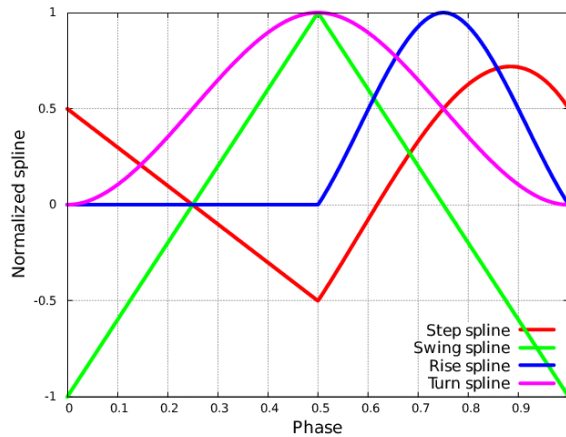


Figura 3.2. – Os quatro períodos dos padrões de splines.

Fonte – QUENTIN (2022).

A biblioteca IKWalk leva em consideração os 12 graus de liberdades das pernas do robô humanoide. As coordenadas seguem o sistema localmente tangente (ENU) que considera a Terra localmente plana, onde a sequência de eixo extrínseca x-y-z correspondem à composição de rotação elementares denotadas por ZYX, a Fig. 3.3 mostra o padrão seguindo para orientação de deslocamento dos motores. Para simplificar os cálculos operacionais da cinemática inversa 6 graus de liberdades de ambas as pernas são alinhados no eixo vertical Z.

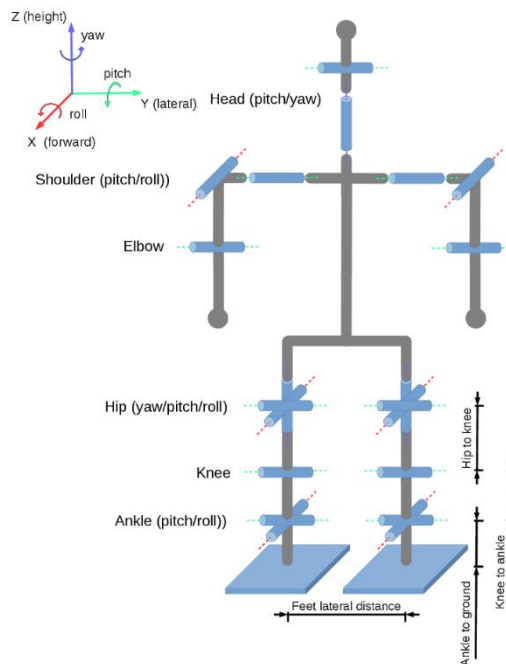


Figura 3.3 - Modelo Cinemático e as referências da estrutura.

Fonte – QUENTIN (2022).

Os cálculos da cinemática inversa levam em consideração quatro dimensões do modelo estrutural do robô humanoide, que são as dimensões destacadas na Fig. 3.3. As dimensões do modelo que foram utilizadas neste trabalho são destacadas na tabela 1.

Tabela 1 - Tabela de Dimensões usadas na Cinemática Inversa.

Dimensão	Distância (m)
Centro do eixo da cintura até o eixo do joelho	0,110
Eixo do joelho até o centro dos eixos do tornozelo	0,110
O centro dos eixos do tornozelo até as solas dos pés	0,034
Distância lateral dos pés	0,095

Na Fig. 3.2 é possível ver os quatro padrões de splines que são usados para gerar todos os movimentos no espaço cartesiano. Na fase igual a 0 o pé esquerdo deixa o chão e começa o movimento de caminhada, no momento em que a fase é igual a 0,5, o pé direito começa a deixar o chão enquanto o pé esquerdo está entrando em contato com o chão, no entanto quando o pé direito termina o seu ciclo de caminhada o valor da fase volta a ser igual a zero. Os quatro padrões de splines podem ser definidos como, QUENTI (2015):

- Step spline: Deslocamento dos passos para frente e na lateral (eixo X e Y) em relação ao tronco.
- Rise spline: Movimento do pé no eixo Z.
- Turn spline: Rotação do pé em relação a yaw usado para a rotação do robô.
- Swing spline: Oscilação do tronco na lateral (Y) com relação aos pés usados para mover o centro de massa em direção ao pé de apoio.

Além da implementação das dimensões do modelo, a biblioteca IKWalk precisa de alguns parâmetros a mais para configuração da caminhada. Os parâmetros usados nesse projeto, os nomes das variáveis e seus valores foram retirados do diretório online da equipe de robótica da Rhoban, nos quais os mesmos foram usados durante a competição mundial de robótica, Robocup, e esses valores podem ser visto na Tabela 2. Os únicos parâmetros que foram alterados para atender os requisitos desse trabalho foi o *freq* e *stepGain*. O parâmetro *freq* foi diminuído, uma vez que o tempo para completar os passos durante o ciclo é maior, tem-se a garantia da aplicação do controle mais vezes durante o ciclo completo de caminhada. E foi preciso aumentar o *stepGain*, para ter uma melhor percepção da correção de postura durante os passos.

Tabela 2 - Parâmetros para caminhada.

Parâmetro	Definição	Valores
enableGain	Permite a suavização da caminhada com pausas	1.0
stepGain	Distância do passo para frente (X)	0.04 m
lateralGain	Distância do passo para lateral (Y)	0.0 m
turnGain	Ângulo de rotação do passo.	0.0°
freq	Frequência para completar o ciclo de caminhada (dois passos).	1.3 Hz
supportPhaseRatio	A duração da fase de suporte duplo no ciclo da caminhada.	0.0 m
footYOffset	Distância lateral entre os 2 pés.	0.03 m

riseGain	Altura do pé (Z) durante o passo para frente.	0.04 m
swingGain	Amplitude da oscilação lateral do tronco em relação aos pés.	0.02
swingPhase	Mudança de fase usada para dessincronizar a oscilação lateral do tronco durante o movimento dos pés.	0.25
trunkXOffset	Translação frontal do tronco (X) em relação aos pés.	0.02 m
trunkYOffset	Translação lateral do tronco (Y) em relação aos pés.	0.0 m
trunkZOffset	Altura do tronco em relação ao chão.	0.02 m
trunkPitch	Orientação do tronco em pitch (Y).	0.15
trunkRoll	Orientação do tronco em roll (X).	0

Os resultados nas posições dos motores da perna durante o ciclo da caminhada podem ser vistos na Fig 3.4.

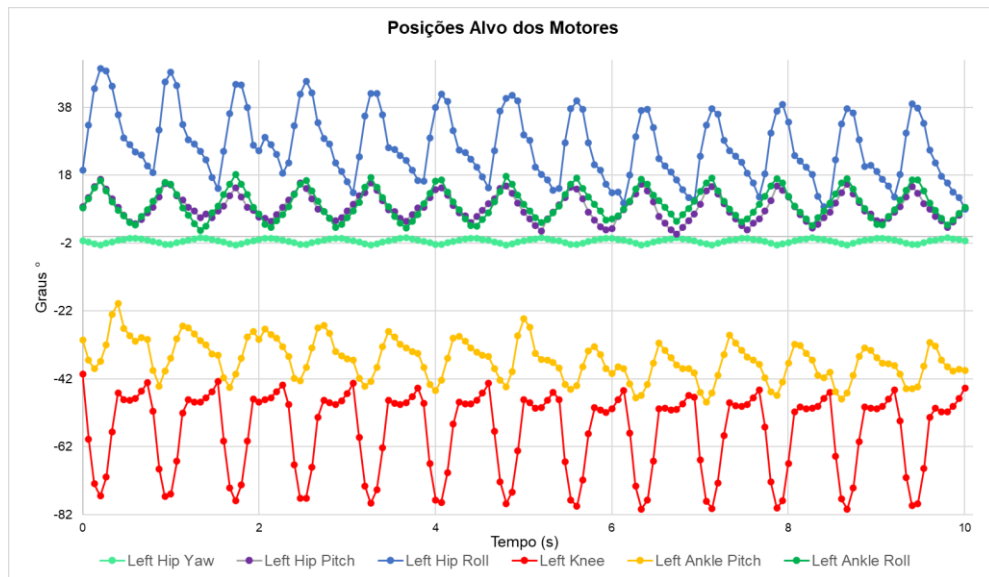


Figura 4.4 - Posição dos motores durante o tempo de caminhada.

Fonte – Acervo do Autor.

A biblioteca pode ser encontrada no repositório aberto do GitHub da equipe de robótica Rhoban, o link para o repositório se encontra no link, <https://github.com/Rhoban/IKWalk>.

3.5. IMPLEMENTAÇÃO DO FILTRO DE KALMAN ESTENDIDO

O método usado nesse trabalho para calcular o movimento de caminhada do robô bípede foi o cálculo da cinemática inversa em um circuito aberto, no qual o sistema apenas faz os cálculos de posição dos motores das pernas, não recebendo nenhuma informação de retorno de sensores enquanto o robô se desloca, portanto não é possível detectar nenhuma desestabilidade durante o percurso. Recebendo apenas os valores do sensor de unidade de medição inercial (IMU), é necessário para uma caminhada estável o torso do robô se manter em valores de ϕ_t e θ_t constantes, os valores concebidos nesse trabalho para ambas variáveis foram zero. Se durante a caminhada o robô sair da referência imposta, é necessário então fazer a correção do erro.

Para a realizar a medição da orientação do torso, foi necessário a implementação do IMU, com um giroscópio e um acelerômetro para monitoramento da angulação, velocidade angular e aceleração angular do torso respectivamente. Entretanto,

quando esses sensores são aplicados fora de simulação, pode-se perceber que os valores são contaminados por ruídos causados por diferentes interferências. Portanto, a aplicação de filtro se tornar necessária para coletar valores com confiabilidade. Nesse projeto será implementado o filtro de Kalman estendido.

Geralmente em sistemas não simulados não é possível ter acesso aos estados da planta, apenas acesso aos valores de entrada e saída, por isso que precisa do observador de estado, no caso do robô humanoide o observador de estados é composto pelos ângulos de Euler do torso $x = [\phi, \theta, \psi]^T$, onde ϕ , θ , e ψ são os ângulos roll, pitch e yaw e pode ser demonstrado na Fig 2.2. Os valores das velocidades angulares do torso na coordenada do torso $u = [p, q, r]^T$ medidos pelo giroscópio. Por último, o vetor de medição é composto pela aceleração do torso no sistema de coordenada do torso $z = [a_x, a_y, a_z]^T$, no qual a aceleração é medida pelo acelerômetro.

Para implementar o observador, foi utilizado o filtro de Kalman estendido (EKF), já que é uma extensão não linear do filtro de Kalman no qual usa a primeira ordem da expansão de Taylor. Nesse trabalho foi escolhido o uso do EKF, por se tratar de um sistema dinâmico não-linear que por sua vez é complicado de obter uma estimação ótima, e para obter um algoritmo viável computacionalmente é preciso usar as aproximações do método EKF.

Substituindo os valores de x , u , v , nas equações do controlador de inclinação de torso, é possível obter a equação de cinemática do corpo discretizada, equação 2.30 e suas linearização através dos cálculos Jacobianos equação 2.31 e 2.32. A atualização da medição linear ótima de uma matriz de medição não linear h_k , equação 2.33 é dada pelas x_k , K_k e P_k mostrado no Algoritmo 3. Para achar uma atualização de medição que possa ser convenientemente programada, é preciso aplicar o Jacobiano na função da aceleração da gravidade projetada nos eixos do torso, obtendo assim a função H_k , equação 2.37, MAXIMO (2017).

Após calcular o Jacobiano e definir as variáveis do observador, entrada e medição do sistema dinâmico, onde é possível definir a atualização de tempo e dos cálculos da predição do erro de covariância P_{k-1} e atualização da medição através do processo de estimativa P_k e \hat{x}_k , e do erro de covariância K_k , FRANK (2008). A

aplicação do EKF nesse trabalho foi feita em python e a lógica pode ser descrita pelo Algoritmo 3 mostrado abaixo:

Algorithm 3: Aplicação do Filtro de Kalman Estendido [EKF]

```

Function Inicializar( $P_0, \hat{x}_0$ ):
     $\hat{x}_{0|0} = \hat{x}_0$ 
     $P_{0|0} = P_0$ 
end
Function Atualização( $\hat{x}_{k-1|k-1}, P_{k-1|k-1}, u_{k-1}, z_k$ ):
    Priori
     $\hat{x}_{k|k-1} = f_k(\hat{x}_{k-1|k-1}, u_{k-1})$ 
     $F_k = \frac{\partial f_k}{\partial x} \Big|_{\hat{x}=\hat{x}_{k-1|k-1}, u=u_{k-1}}$ 
     $L_k = \frac{\partial f_k}{\partial w} \Big|_{\hat{x}=\hat{x}_{k-1|k-1}, u=u_{k-1}}$ 
     $P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + L_k Q_k L_k^T$ 
    Filtro
    if  $|z| \in [g_{min}, g_{max}]$  then
         $H_k = \frac{\partial h_k}{\partial x} \Big|_{\hat{x}=\hat{x}_{k|k-1}, u=u_{k-1}}$ 
         $S_k = H_k P_{k|k-1} H_k^T + R_k$ 
         $K_k = P_{k|k-1} H_k^T S_k^{-1}$ 
         $\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(z_k - h(\hat{x}_{k|k-1}))$ 
         $P_{k|k} = (I - (K_k H_k)) P_{k|k-1}$ 
    end Function

```

Considerando apenas os efeitos da gravidade da Terra para os cálculos de tempo de medição, define-se como a aceleração da Terra $g = 9,81 \text{ m/s}^2$. Os valores do intervalo para a fase de filtragem do EKF g_{min} e g_{max} são valores da ação da aceleração no eixo z, tal que foram encontrados por meio de repetição da caminhada do robô humanoide e foram analisados as acelerações que a robô atingia antes de cair, esses valores foram encontrados como $6,53 \text{ m/s}^2$ e $11,83 \text{ m/s}^2$, para g_{min} e g_{max} respectivamente. A implementação desses valores limites tem o objetivo de evitar alimentação do observador com valores que poderão deixar o sistema instável.

Com tudo, deve se assumir ruídos gaussianos $v \sim (0, Q)$ e $w \sim (0, R)$ para as medições do giroscópio e acelerômetro, respectivamente. Na simulação é possível adicional os valores de 0 a 1 para o sensor IMU, onde 0 é correspondente a nenhum ruído e 1 é correspondente ao ruído gaussiano com uma derivação de $\frac{\pi}{2}$. Nesse trabalho, foi considerado o ruído da unidade de medição inercial como aproximadamente 0,063 radianos, pois é um valor intermediário onde é possível ver claramente a atuação do filtro. Foram feitos teste com valores de ruídos maiores e

menores do que o aplicado para esse trabalho, os valores podem ser visto na Fig. 3.5. Para um ruído muito alto, ou seja de 0.63 radianos, é possível notar que o processo de filtragem tem um baixo desempenho,

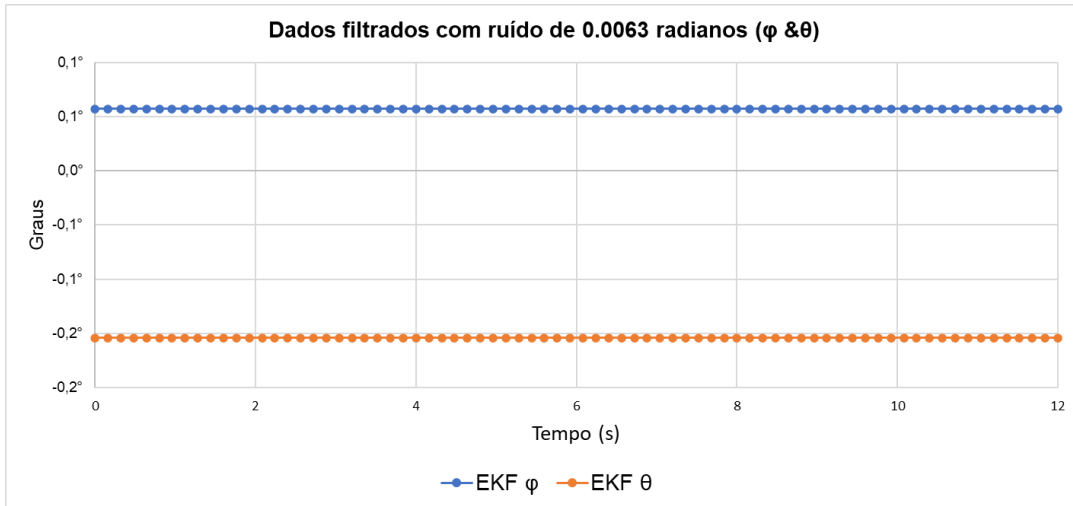


Figura 5.5 - Processo de Filtragem com o ruído da IMU de 0.0063 radianos.

Fonte – Acervo do Autor.

É preciso considerar que o ruído do giroscópio é integrado na equação 2.32 de uma maneira não linear, enquanto o ruído do acelerômetro é inserido na equação 2.32, por isso a importância de considerar o L_k no cálculo do EKF, MAXIMO (2017).

Finalmente, será possível calcular os parâmetros de erro em ϕ e θ do controlador de inclinação do tronco, que por sua vez são as equações de $\Delta\phi_{hr,o}$, $\Delta\phi_{ar,o}$, $\Delta\theta_{hp,o}$, $\Delta\theta_{kp,o}$ e $\Delta\theta_{ap,o}$. Após os cálculos do controle de orientação, os valores de $\Delta\phi$ e $\Delta\theta$ são somados à planta de controle para realizar a correção dos valores alvos dos motores.

Os valores dos ganhos K_p e K_v foram encontrados manualmente pela análise visual do comportamento da caminhada do robô humanoide durante o tempo de caminhada na simulação. Os valores para do ganho que apresentou o melhor resultado para as características do robô simulado nesse trabalho pode ser visto na Tabela 3.

Tabela 3 - Valores dos ganhos K_p e K_v para os roll e pitch.

Ganhos para as juntas <i>roll</i> e <i>pitch</i>		
Nomes	K_p	K_v
Hip Roll	0.08	0.0125
Ankle Roll	0.05	0.0085
Hip Pitch	0.3	0.04
Knee Pitch	0.1	0.003
Ankle Pitch	0.24	0.04

CAPÍTULO IV

RESULTADO

4.1. Validação do Filtro de Kalman Estendido

A implementação do filtro de Kalman estendido tem como objetivo o cálculo da estimativa ótima do estado de um sistema não linear através da estimação do seu erro de covariância e trabalhar como filtro para sensores, eliminando o ruído através da estimativa ótima. Portanto, nesse trabalho foi utilizado o EFK como filtro para o sensor que mede o grau de inclinação do torso IMU, uma vez que o ruído atrapalharia o cálculo do controle inserindo dados não desejável durante a aplicação, podendo deixar o sistema instável.

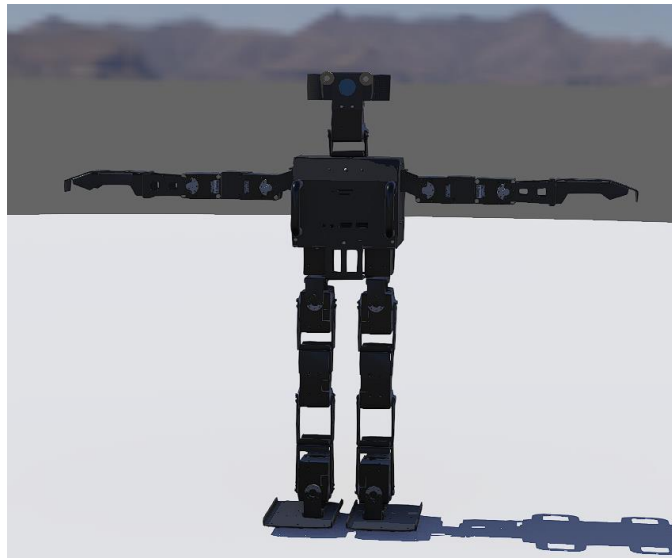


Figura 4.1 – Posição inicial para mensuração do IMU.

Fonte – Webots User Guide.

Foram realizadas coletas do IMU para os ângulos ϕ e θ , os quais são os ângulos principais que tem maior influência na caminhada, MAXIMO (2017). A coleta foi realizada sem a ação do EKF e com o robô no estado inicial de acordo com a Fig. 4.1, ou seja, parado e com os valores da IMU aproximadamente igual a zero. A imagem da coleta pode ser vista na Fig. 4.2.

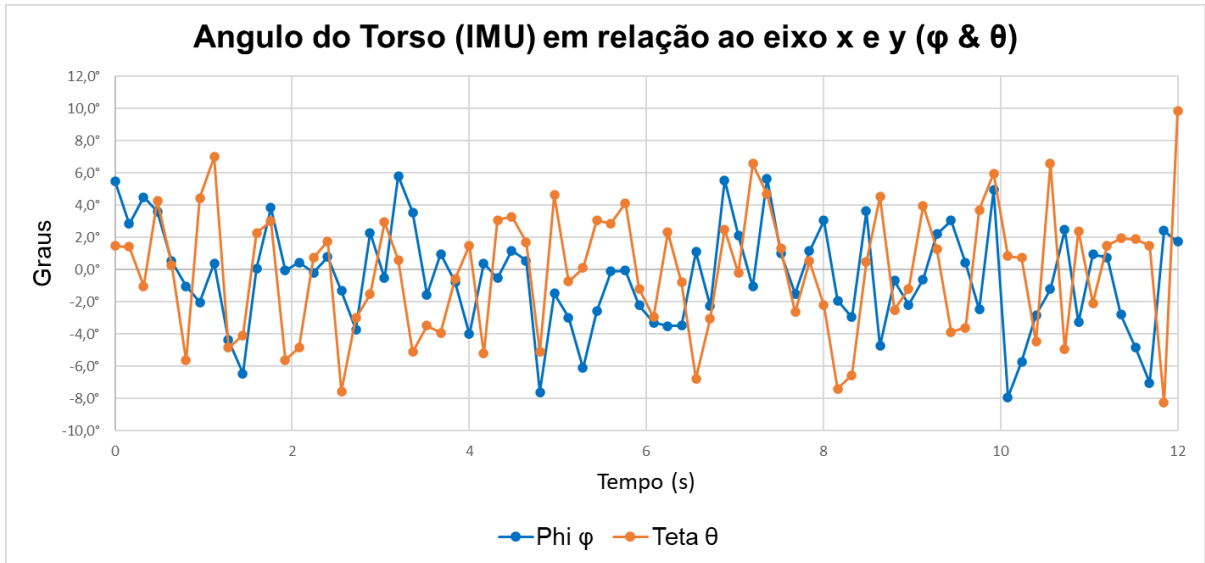


Figura 4.2 - Coleta dos ângulos ϕ e θ (IMU).

Fonte – Acervo do Autor.

Como dito na sessão 3.5, foi adicionado na simulação um ruído gaussiano de 0,063 radianos para o IMU. É possível ver o efeito do ruído para a coleta dos ângulos de ϕ e θ na Fig.4.2, onde o sensor coleta valores impuros durante o tempo de coleta, como o estado do nosso sistema são diretamente relacionados aos ângulos do torso propagando o ruído por todo o sistema, sendo assim o controlado que depende da valores do sensor se torna não confiável. Para mitigar o efeito do ruído, foi aplicado o EKF para eliminar o ruído.

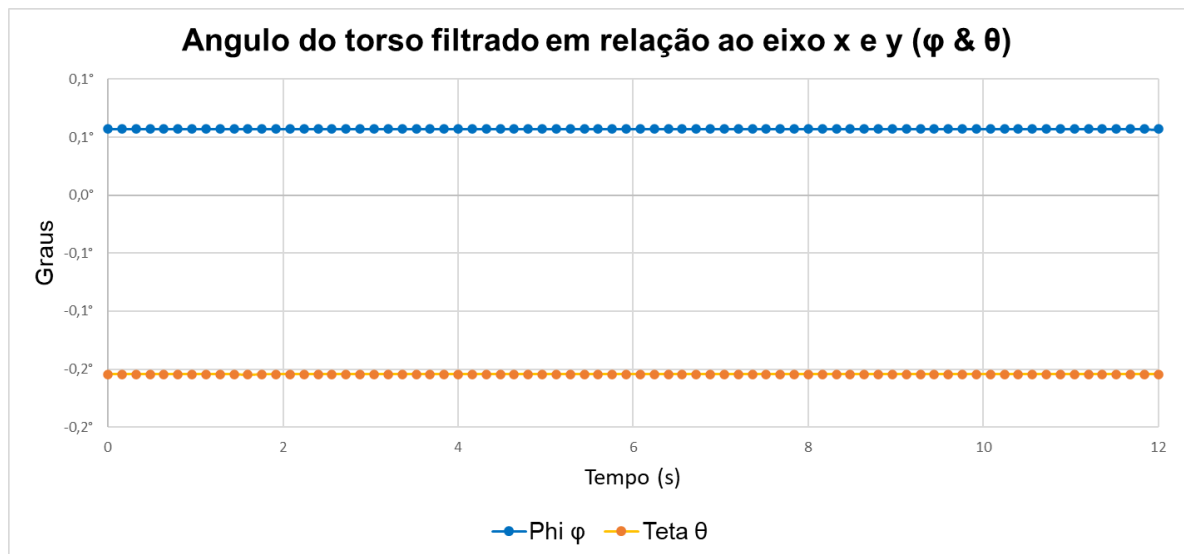


Figura 4.3 - Ângulo do torso ϕ e θ após a ação do EKF.

Fonte – Acervo do Autor.

Na Fig. 4.3 é possível ver o efeito do filtro da estimativa do filtro de kalman estendido na coleta dos ângulos ϕ e θ da IMU, uma vez que o dado depois do processo de filtragem está livre de ruídos e estável. Os valores de ϕ e θ estão próximos de zero e permanece constante durante o tempo de coleta, como o esperado na situação onde o robô permanece parado e com a IMU alinhada com o eixo XYZ.

Podemos ver melhor a discrepância entre os dados estimados pela EKF e os dados puros da IMU nas Fig 4.4 e Fig. 4.5.

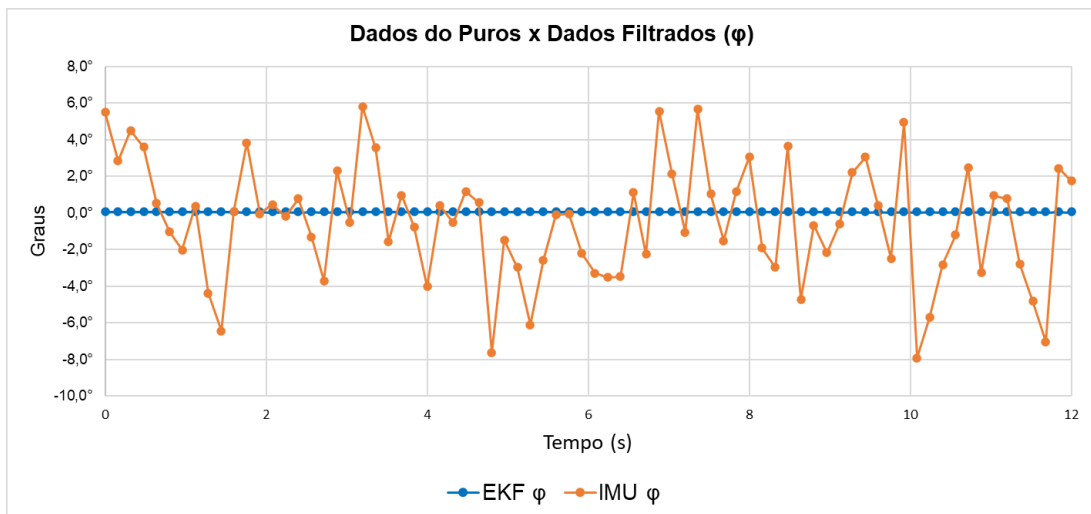


Figura 4.4 – Ângulo ϕ antes e após passarem pela filtragem.

Fonte – Acervo do Autor.

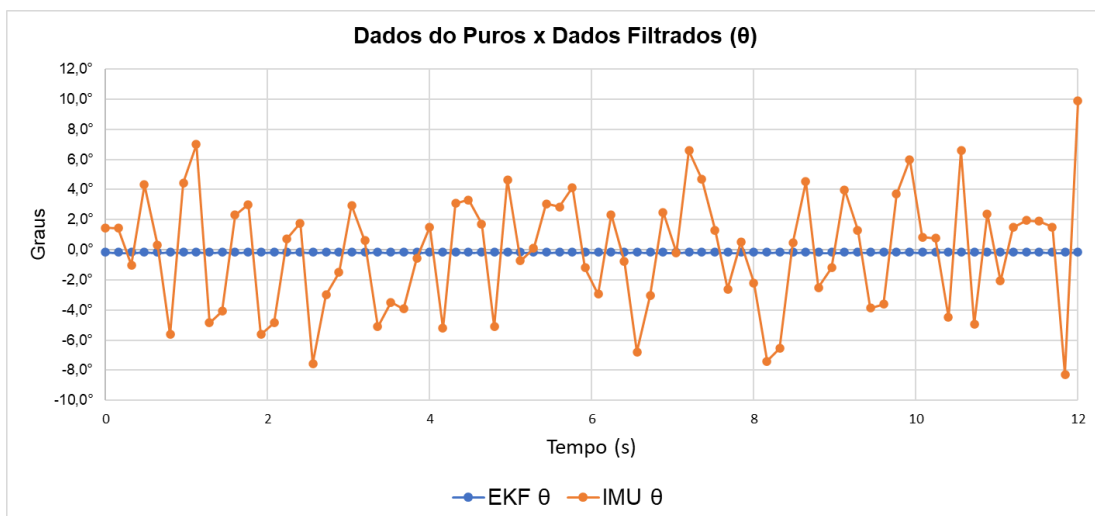


Figura 4.5 – Ângulo θ antes e após passarem pela filtragem.

Fonte – Acervo do Autor.

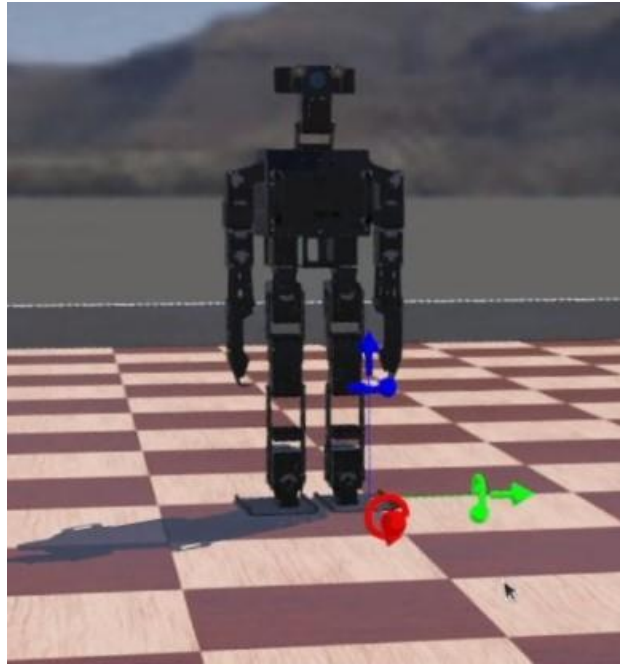


Figura 4.6 – Posição inicial do robô (Suporte Duplo).

Fonte – Acervo do Autor.

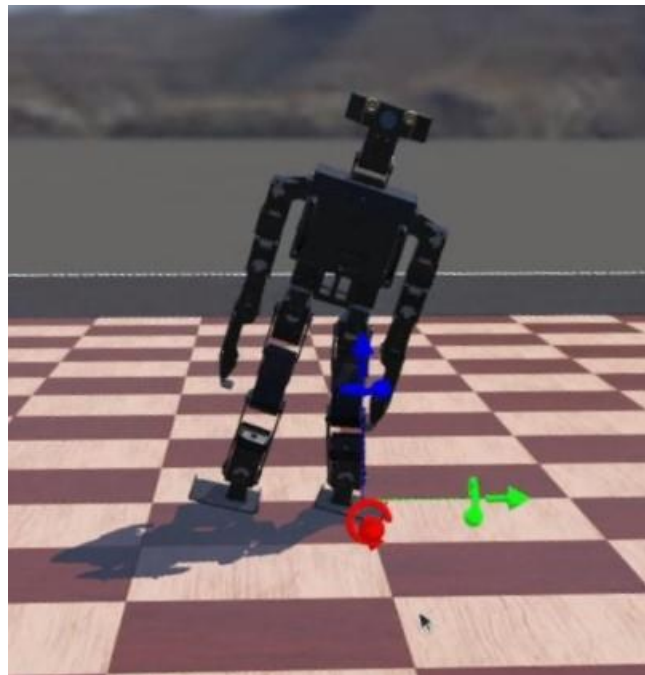


Figura 4.7 – Caminhada (Suporte Único).

Fonte – Acervo do Autor.

4.2. Validação do Controlador de Caminhada

Após o processo de filtragem do sensor IMU, é possível alimentar o controlador de caminhada com valores confiáveis, que demonstra o deslocamento da angulação do torso durante a caminhada. O controlador faz uma compensação para os valores alvos dos motores da perna quando enxerga que o robô sofre algum deslocamento do torso e sai da referência ϕ_t e θ_t , que são os ângulos do tronco em relação ao eixo cartesiano. Portanto quando os ângulos de inclinação do torso ϕ e θ apresentam valores diferentes de referência ϕ_t e θ_t , o controlador entende que pode ser uma possível queda e corrige para que o torso volte ao valor de referência.

Os ganhos do controle K_p e K_v foram encontrados através de tentativa e erro, analisando o efeito de cada valor de ganho nos motores durante o passo. Visualmente é possível ver o efeito do controle atuando nas juntas quando o torso sofre inclinação e saindo da sua referência. Uma vez que coloca o valor do ganho muito alto, a compensação para quando o torso sai da referência é agressiva levando a uma rotação brusca e desestabilizando o robô, e quando é colocado um valor de ganho muito baixo, não é possível ver o efeito da compensação na orientação dos motores.

As Fig 5.1 e Fig. 5.2 demonstram os ângulos de inclinação do torso do robô referente aos ângulos ϕ e θ durante a caminhada. Os pontos laranja representam o movimento de inclinação do tronco durante a caminhada sem nenhuma ação do controlador e os pontos azuis representam o deslocamento do tronco durante a caminhada com a aplicação do controle de caminhada que é representado pelas equações 2.39, 2.40, 2.41, 2.42 e 2.43.

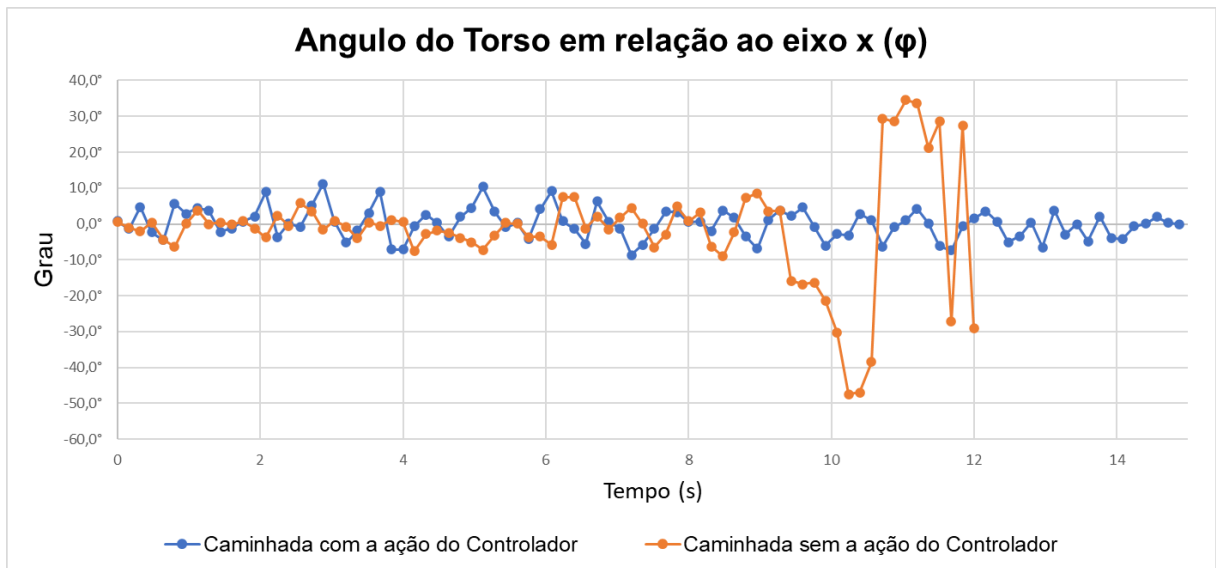


Figura 5.1 - Representação do deslocamento do torso em ϕ .

Fonte – Acervo do Autor.

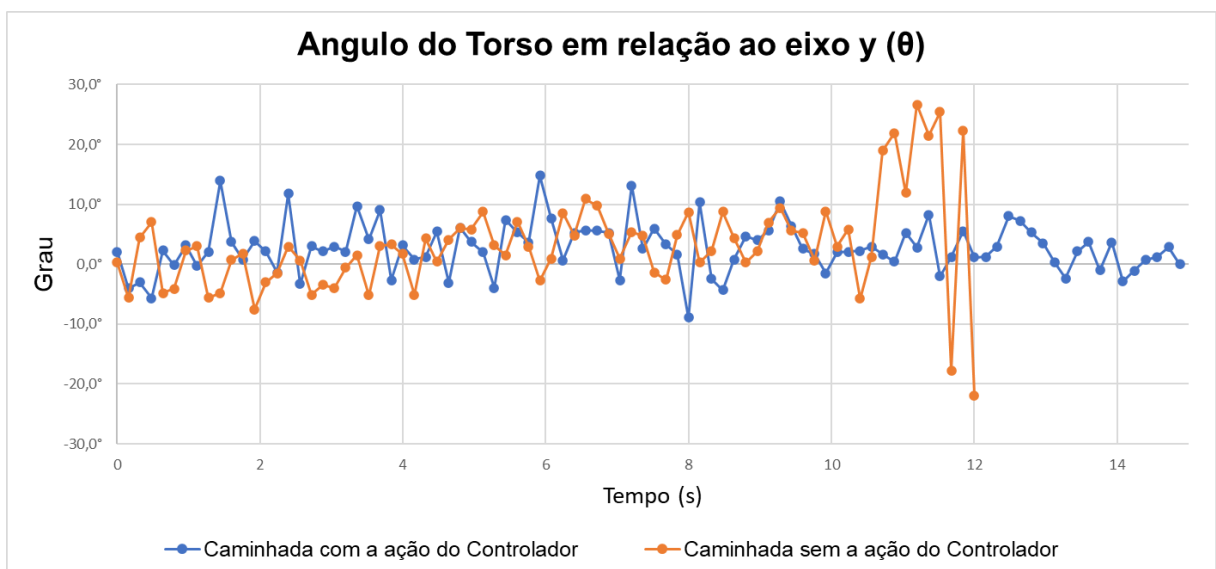


Figura 5.2 - Representação do deslocamento do torso em θ .

Fonte – Acervo do Autor.

As Fig 5.1 e Fig 5.2 nos mostram que entre 10 e 12 segundos o torso da caminhada sem a ação do controlador sofre uma perturbação durante o percurso e não consegue corrigir, causando assim a queda do robô, não podendo completar a caminhada no tempo proposto. No entanto, é possível perceber que durante a caminhada com a ação do controle o robô mantém uma oscilação quase constante de ϕ e θ ficando entre -0,2 e 0,2 radianos, sendo possível notar que no tempo entre

8 e 10 segundos, quando robô passa do intervalo citado, há uma correção na postura do robô, uma vez que a oscilação de inclinação do torso diminui.

Pode se concluir que o controlador de caminhada teve um papel importante para prevenção da queda do robô durante seu trajeto, controlando a inclinação do torso durante o trajeto pelo ajuste das posições alvo dos motores e impedindo que o torso sofra grandes perturbações, evitando assim a queda durante o percurso.

No link a seguir encontra-se o software desenvolvido pelo autor para o desenvolvimento desse manuscrito pode ser encontrado no *repositório aberto do GitHub*: <https://github.com/ronaldocda/ExtendedKalmanFilter/tree/master>.

CAPÍTULO V

CONCLUSÃO

O presente trabalho efetuou a investigação da aplicação do filtro de Kalman estendido em um sensor de unidade de medida inercial, onde a dinâmica desse sensor era representada pelas equações de cinemática de rotação de um corpo rígido de um robô bípede humanoide. Após a aplicação do filtro no sensor IMU, os dados filtrados foram utilizados para alimentação de um controle proporcional de caminhada com o intuito de estabilizar o torso do robô durante os passos da caminhada. A aplicação foi gerada em um ambiente simulado onde não se tem interferência de agentes externos.

A partir da implementação, analisou o efeito instantâneo de remoção de ruídos do filtro nos ângulos coletados pelo sensor. O resultado obtido evidencia que o EKF é consistente em filtrar sistemas não lineares através da previsão de estados e geração de ganhos pelo cálculo preditivo dos erros de covariância para cada interação, além de retornar um sinal totalmente sem ruídos.

Analisando o comportamento do controlador e a correção à cada passo durante a caminhada de um robô bípede, é possível observar pontos com potencial de aprimoramento. Uma vez dito que o EKF é um método de malha aberta, portanto não tem um retorno do posicionamento do centro de massa, implicando assim na dificuldade de correção dos valores alvos dos motores quando o torso sai de sua referência, ou melhor dizendo, quando o robô está em desequilíbrio. Além de que os ganhos do controlador foram ajustados sem nenhuma técnica de otimização, resultando em valores não ótimos para K_p e K_v , explicando assim a não estabilização do torso nos ângulos de referência ϕ_t e θ_t .

Com o objetivo de aumentar a estabilidade do torso durante o passo do robô humanoide propõe-se no CAPÍTULO VI algumas alternativas para aprimorar não só o filtro de Kalman estendido, mas também para a estabilização da caminhada, no qual se sabe que quando aplicada fora de um ambiente simulado, a dinâmica do movimento pode sofrer com fatores externos como: limites físicos dos motores, forças externas a caminhada, obstáculos, caminhada em solos diferentes e folgas entre elementos articulados.

O desenvolvimento do presente trabalho representou uma oportunidade valiosa para o autor de expandir e aprofundar seus conhecimentos em uma área que lhe despertou particular interesse durante a Graduação: Controle de Máquina aplicado à Robótica. De fato, o cumprimento da proposta exigiu do mesmo um grande esforço para assimilar várias técnicas e teorias fundamentais no corrente estado da arte de controladores ótimos concebidos para lidar com problemas relevantes da robótica. A experiência na condução deste trabalho abriu um leque de objetivos e desafios novos que representa um divisor de águas na formação acadêmica do autor.

CAPÍTULO VI

TRABALHOS FUTUROS

Neste capítulo serão tratadas as oportunidades de melhoria ou continuação desse projeto.

- Validar o modelo de simulação de corpo inteiro usando um robô humanoide real;
- Implementar e testar uma formulação de caminhada simplificada em um robô humanoide real;
- Implementação de um controle em malha fechada, como por exemplo, um PID (Proporcional, Integrativo e Derivativo) para a movimento do robô. Assim espera-se ser possível, um controle mais refinado de aceleração e velocidade dos atuadores das articulações;
- Desenvolver uma simulação completa de corpo inteiro para um robô humanoide, incluindo fenômenos relevantes, como colisão, atrito e tipo de superfície no qual o robô se movimenta;
- Determinar uma regra para exclusão de estados antigos no vetor de estados, a fim de melhorar a capacidade computacional do filtro de Kalman estendido;
- Realizar uma correlação estatística dos estados dentro do filtro de Kalman estendido, utilizando a distância de Mahalanobis, a fim de evitar que um estado já observado anteriormente seja adicionado novamente ao vetor de estados;
- Fazer um estudo detalhado sobre a influência da posição de um ponto de interesse na imagem sobre a matriz de covariância do ruído de medição (matriz Q no filtro de Kalman estendido)

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Holtkamp, M. J. and de Jong, S. (2006). Robot localisation using sift and active monocular vision.
- [2] Leonard, J.J., Durrant-Whyte, H. F. , Cox, I. J. Dynamic Map Building for an Autonomous Mobile Robot. IEEE IROS, 1990.
- [3] Choset, H. Lynch, K.M., Hutchinson, S. , Kantor, G., Burgard, W., Kavraki, L. E., Thrun, S. Principle of Robot Motion ; Theory, Algorithms, and Implementation..
- [4] Bailey, T., Nebot, E. Localisation in large-scale environment. Robotics and Autonomous Systems, v.37, 2001.
- [5] Siegwart, R. Nourbakhsh, I. R. Introduction to Autonomous Mobile Robots, MIT Press, 2004.
- [6] WOLF, D. F. et al. Robótica móvel inteligente: Da simulação às aplicações no mundo real. [S.l.]: Mini-Curso: Jornada de Atualização em Informática (JAI), 2009
- [7] <https://cyberbotics.com/doc/guide/robotis-op3>. Acessado em 22 de agosto de 2022.
- [8] <https://www.robotis.us/robotis-op3/>. Acessado em 23 de agosto de 2022.
- [9] TAKENAKA, T.; MATSUMOTO, T.; YOSHIIKE, T. Geração e Controle de Movimento em TempoReal para Robô Bípede -1º Relatório: Geração de Padrão de Marcha-. In: Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. St. Louis, EUA: IEEE, 2009.
- [10] THOMAZINI, D.; ALBUQUERQUE, B. U. P. Sensores Industriais - Fundamentos e Aplicações. 222p. 2011.
- [11] ROUXEL, Q; Multi-Contact Motion Retargeting using Whole-body Optimization of Full Kinematics and Sequential Force Equilibrium, EUA: IEEE, 2022
- [12] MAXIMO, M. R. O. A.; RIBEIRO, C. H. C.; AFONSO, R. J. M. Modeling of a position servo used in robotics applications. In: Proceedings of the 2017

Simpósio Brasileiro de Automação Inteligente (SBAI). Porto Alegre, SC, Brazil: SBA, 2017.

[13] ASTROM, K. J.; WITTENMARK, B. Adaptive Control. 2. ed. [S.I.]: Addison Wesley, 1995.

[14] MOUSSA, S. Fundamentos de Biomecânica: Aplicações da Mecatronica no Corpo Humano. [S.I.]: MOUSSA SALEN SIMHON, 2013.

[15] VENÂNCIO, M. M. Desenvolvimento de Técnicas de Estabilização de Caminhada para Robô Humanoide com Detecção de Diferentes Tipos de Terrenos. 2018. 132 f., Dissertação de Mestrado, Universidade Federal de Uberlândia, Brasil.

[16] McKerrow, Phillip John, Robotics, an Academic Discipline Robotics, 2, 3, pp. 267-274.

[17] SANTOS, G. L. Localização de Robôs Móveis Autônomos Utilizando Fusão Sensorial de Odometria e Visão Monocular. 2010. 66 f., Dissertação de Mestrado, Universidade Federal do Rio Grande do Norte, Brasil.

[18] <https://github.com/Rhoban/IKWalk>. Acessado em 04 de agosto de 2022.

[19] <https://www.ros.org/blog/why-ros/>. Acessado em 24 de agosto de 2022.

[20] QUENTIN, R.; GRÉGORE, P.; LUDOVIC, H.; STEVE N'GUYEN, OLIVER, L., Rhoban Hardware and Software Open Source Contributions for RoboCup Humanoids. 2015. KOREA: IEEE, 2015.

[21] WESTERVELT, E.; GRIZZLE, J.; CHEVALLEREAU, C.; CHOI, J.; MORRIS, B.; Feedback Control of DynamicBipedalRobot Locomotion. 2007. Inglaterra.