

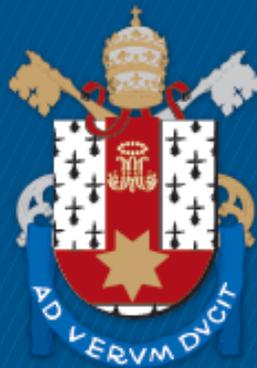
ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA
MESTRADO EM ENGENHARIA ELÉTRICA

RODRIGO ALVES MEDEIROS

**FILTRO DE KALMAN ESTENDIDO BASEADO EM QUATERNIONS PARA
ESTIMAÇÃO DE UM SISTEMA DE 6-DOF EMBARCADO**

Porto Alegre
2020

PÓS-GRADUAÇÃO - *STRICTO SENSU*



Pontifícia Universidade Católica
do Rio Grande do Sul

Rodrigo Alves Medeiros

**Filtro de Kalman Estendido Baseado em
Quaternions para Estimação de um Sistema de
6-DOF Embarcado**

Porto Alegre - RS, Brasil

2020

Rodrigo Alves Medeiros

Filtro de Kalman Estendido Baseado em Quaternions para Estimação de um Sistema de 6-DOF Embarcado

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Pontifícia Universidade Católica do Rio Grande do Sul, como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica.

Área de concentração: Sinais, Sistemas e Tecnologia da Informação

Linha de Pesquisa: Automação e Sistemas.

Pontifícia Universidade Católica do Rio Grande do Sul – PUCRS

Escola Politécnica

Programa de Pós-Graduação em Engenharia Elétrica

Orientador: Rafael Fraga Garibotti

Coorientador: Guilherme Araujo Pimentel

Porto Alegre - RS, Brasil

2020

Agradecimentos

Primeiramente, eu gostaria de agradecer aos meus pais, Evandro Roxo Medeiros e Ezezi Alves Pimenta, ao meu irmão Alexandre Alves Medeiros, e à toda minha família, que sempre esteve ao meu lado me incentivando e dando o apoio necessário para seguir em frente.

Agradeço também aos meus orientadores, Prof. Rafael Garibotti e Prof. Guilherme Pimentel, pelas incontáveis sugestões e auxílios durante a elaboração deste trabalho.

Meus sinceros agradecimento a meus colegas de mestrado e de laboratório, em especial ao amigos Eduardo Nery, Lucas Campos, Ricardo Rodrigo e Tomlyta Velasquez, cujas amizades durante o curso me ajudaram nas mais diversas situações.

Finalmente, agradeço à PUCRS e à CAPES pela infraestrutura fornecida e pelo apoio financeiro.

*“Por vezes sentimos que aquilo que fazemos não é senão uma gota de água no mar.
Mas o mar seria menor se lhe faltasse uma gota.”*
(Madre Teresa de Calcuta)

Resumo

Este trabalho propõe a implementação de um Filtro de Kalman Estendido para estimar a orientação e posição de um corpo rígido de seis graus de liberdade (6-DOF) embarcado em um FPGA utilizando um processador comercial. A abordagem utilizada para parametrizar a dinâmica do corpo rígido foi por Quaternions, que por possuírem função particular para rotações tridimensionais, fornecem uma alternativa aos tradicionais Ângulos de Euler, reduzindo o esforço computacional envolvido, bem como o problema de *Gimbal Lock*, que ocorre quando há singularidades nas matrizes de rotações. Para validação do estimador foi utilizado uma Plataforma Stewart, equipada com um acelerômetro, um giroscópio e uma câmera. Embora este trabalho considere o uso de sensores comuns de baixo custo e o uso de marcadores com simples geometria, os resultados mostram excelente desempenho do filtro desenvolvido, podendo estimar a posição e orientação com um erro abaixo de 8,14 mm e 0,63°, respectivamente. Além disso, a eficácia da abordagem também foi avaliada, mostrando que o filtro é capaz de convergir rapidamente quando os marcadores são recuperados após uma perda de dados da câmera por um curto período de tempo.

Palavras-chave: Filtro de Kalman Estendido, Sistemas Embarcados, FPGA, Plataforma Stewart, Unidade de Medidas Inerciais, Câmera.

Abstract

This work proposes the implementation of an Extended Kalman Filter to estimate the orientation and position of a six degrees of freedom (6-DOF) rigid body embedded in an FPGA with a commercial processor. The approach used to parameterize the dynamics of the rigid body was by Quaternions, which, because they have a particular function for three-dimensional rotations, provide an alternative to traditional Euler angles, reducing the computational effort involved, as well as the *Gimbal Lock* problem, that occurs when there are singularities in the rotation matrices. To validate the estimator, a Stewart Platform was used, equipped with an accelerometer, a gyroscope and a camera. Although this work considers the use of common low-cost sensors and the use of markers with simple geometry, the results show excellent performance of the developed filter, being able to estimate the position and orientation with an error below 8.14 mm and 0.63° , respectively. Furthermore, the effectiveness of the approach has also been evaluated, showing that the filter is able to converge quickly when the markers are retrieved after a loss of camera data for a short period of time.

Keywords: Extended Kalman Filter, Embedded Systems, FPGA, Stewart Platform, Inertial Measurement Unit, Camera.

Lista de ilustrações

Figura 1 – Representação gráfica de uma rotação utilizando quaternion no espaço 3D.	27
Figura 2 – Plataforma Stewart e os sistemas de referências.	28
Figura 3 – Relações geométricas.	28
Figura 4 – Movimento de um corpo rígido entre o sistema de coordenadas da câmera C e o sistema de coordenadas global W.	32
Figura 5 – Relação entre o plano da retina da câmera e a matriz de pixels do plano de imagem.	33
Figura 6 – Processo de integração.	39
Figura 7 – Representação gráfica da IMU.	40
Figura 8 – Disposição dos marcadores.	41
Figura 9 – Processo de validação dos estados estimados.	43
Figura 10 – Placa de circuito impresso do conversor RS-232/RS-485: a) bottom layer, b) renderização do top layer, c) renderização do bottom layer.	44
Figura 11 – Experimental setup.	45
Figura 12 – Fluxograma do desenvolvimento do <i>Software</i> embarcado em um FPGA.	52
Figura 13 – Simulação do sensor giroscópio em relação aos dados medidos.	56
Figura 14 – Simulação do sensor acelerômetro em relação aos dados medidos.	57
Figura 15 – Simulação da câmera em relação aos dados medidos.	57
Figura 16 – Orientação e posição inferida pela cinemática direta.	58
Figura 17 – Experimento 1: estimativa da posição inferida pelo Filtro de Kalman Estendido e o erro em relação ao valor real.	59
Figura 18 – Experimento 1: estimativa da orientação inferida pelo Filtro de Kalman Estendido e o erro em relação ao valor real.	59
Figura 19 – Experimento 2: estimativa da posição inferida pelo Filtro de Kalman Estendido e o erro em relação ao valor real.	60
Figura 20 – Experimento 2: estimativa da orientação inferida pelo Filtro de Kalman Estendido e o erro em relação ao valor real.	61

Lista de tabelas

Tabela 1 – Algoritmo do Filtro de Kalman.	36
Tabela 2 – Algoritmo do Filtro de Kalman Estendido.	37
Tabela 3 – Parâmetros Geométricos da Plataforma Stewart.	45
Tabela 4 – Componentes Utilizados	45
Tabela 5 – Algoritmo do EKF para Fusão de Sensores.	51
Tabela 6 – Bibliotecas Desenvolvidas para Implementação do Algoritmo de Estim- mação de Pose.	52
Tabela 7 – Protocolo de Comunicação.	53
Tabela 8 – Covariância Associada aos Sensores.	58
Tabela 9 – Análise Estatística do Experimento Realizado.	60

Lista de abreviaturas e siglas

EKF	Filtro de Kalman Estendido (do Inglês, Extended Kalman Filter)
FPGA	Matriz de Portas Programáveis (do Inglês, Field Programmable Gate Array)
IMU	Unidade de Medida Inercial (do Inglês, Inertial Measurement Unit)
6-DOF	Seis Graus de Liberdade (do Inglês, Six degree of freedom)
UKF	Filtro de Kalman Unscented (do Inglês, Unscented Kalman Filter)
GNRMD	Newton-Raphson Global com Descendência Monotônica (do Inglês, Global Newton-Raphson with Monotonic Descent)
MTT	Rastreamento de Múltiplos Alvos (do Inglês, Multi Tracking Target)
CRC	Verificação de Redundância Cíclica (do Inglês, Cyclic Redundancy Check)

Lista de símbolos

$diag\{A, B\}$	bloco de matriz diagonal formada por A e B
A^T	transposta da matriz A
A^{-1}	inversa da matriz A
I_n	matriz identidade de tamanho n por n
$0_{m \times n}$	matriz de zeros de tamanho de m linhas por n colunas
\dot{x}	derivada em relação ao tempo dx/dt
x_i	i -th elemento do vetor \mathbf{x}
$\ \mathbf{x}\ $	norma euclidiana $\sqrt{\mathbf{x}^T \mathbf{x}}$
\mathbb{R}	conjunto dos números reais
\mathbb{R}_+	conjunto dos números reais positivos
\mathbb{R}^n	vetor com n elementos pertencente ao conjunto de números reais
$\mathbb{R}^{n \times m}$	matriz com $n \times m$ elementos pertencente ao conjunto de números reais
T	período de amostragem
\mathbf{g}	vetor da gravidade

Sumário

1	INTRODUÇÃO	21
1.1	Estado da Arte	22
1.2	Organização do Manuscrito	23
2	PRELIMINARES	25
2.1	Quaternions	25
2.2	Plataforma Stewart	27
2.2.1	Geometria da Plataforma de Stewart	27
2.2.2	Dinâmica da Plataforma Stewart	29
2.3	Modelo do Giroscópio e Acelerômetro	30
2.4	Modelo da Câmera	31
2.4.1	Parâmetros Extrínsecos	31
2.4.2	Parâmetros Intrínsecos	33
2.4.3	Modelo da Câmera <i>Pinhole</i>	34
2.5	Filtro de Kalman Estendido	36
3	SENSOREAMENTO E <i>HARDWARE</i>	39
3.1	Aquisição de Dados	39
3.1.1	IMU – Unidade de Medida Inercial	40
3.1.2	Visão Computacional	40
3.2	Validação dos Estados Estimados	42
3.3	Descrição do <i>Hardware</i>	44
4	MÉTODO DE ESTIMAÇÃO PROPOSTO	47
4.1	Formulação do Filtro de Kalman Estendido	47
4.2	Proposta do Software Embarcado	52
5	RESULTADOS EXPERIMENTAIS	55
5.1	Simulação dos Sensores	55
5.2	Aplicação do Filtro de Kalman Estendido	56
6	CONCLUSÕES	63
6.1	Perspectivas de Trabalhos Futuros	63
	REFERÊNCIAS	65

1 Introdução

Sistemas de seis graus de liberdade (6-DOF), devido sua capacidade de execução de tarefas complexas, estão sendo cada vez mais utilizados em diversos campos industriais, como o exemplo do setor automotivo (MATTILA et al., 2017; Zhang et al., 2018; MARKUS; AGEE; JIMOH, 2017), de entretenimento (SHARMA; RAWAL; TAYAL, 2018; ABOUDONIA; EL-BADAWY; RASHAD, 2017) e militar (XIANG; YU; ZHANG, 2017). Em consequência, há uma ampliação dos campos de estudos voltados aos sistemas 6-DOF, sendo um deles a estimação da posição e orientação. Posto isto, na busca por estimadores mais precisos, encontramos na literatura diferentes metodologias e topologias de sensores que normalmente utilizam câmeras, *encoders*, acelerômetro e giroscópio (GARCÍA et al., 2016; ZHENG; GUO, 2017; CHUNYANG et al., 2016).

Para inferir a posição e orientação de um sistema 6-DOF são utilizados um conjunto de sensores capazes de medir a velocidade angular e a aceleração de objetos e, em alguns casos, também a direção do campo magnético - o conjunto de tais sensores para navegação é chamado de Unidade de Medidas Inerciais (IMU). Tais dispositivos, embora capazes de fornecer medidas a uma frequência superior a 1 kHz, são corrompidos por ruídos e deslocamentos aditivos, também denominados como *Bias* de medição. Diante do acúmulo de erros introduzidos nas medições, o sinal acaba sofrendo um fenômeno conhecido como *Drift*. A fim de mitigar esses problemas, usualmente, são utilizados outros sensores em conjunto com a IMU através de correção e fusão de sinais.

Existem diversas metodologias capazes de realizar a fusão de sensores com o objetivo de obter a estimativa de orientação e posição de um sistema de interesse. Sobre este ponto, existe soluções consideradas mais simples, como a utilização dos filtros complementares (EUSTON et al., 2008), e as técnicas consideradas mais complexas e robustas, como as que utilizam o Filtro de Partículas (OSHMAN; CARMI, 2006). Contudo, devido às propriedades estatísticas, o Filtro de Kalman Estendido (EKF) é a metodologia de estimação e fusão de sensores mais empregada em pesquisas e aplicações práticas, devido à sua maior versatilidade para utilização em diferentes modelos dinâmicos. A funcionalidade deste filtro ocorre em dois processos distintos denominados de predição e correção, através dos quais, respectivamente, se utiliza do modelo do sistema para inferir a saída desejada e, posteriormente, realiza a correção através dos sensores.

A aplicação do EKF requer a representação do sistema de interesse através do seu modelo matemático e, uma vez que a orientação de um objeto é intrinsecamente não linear, as relações matemáticas que descrevem esta dinâmica se tornam complexas. Não obstante a tradicional utilização dos ângulos de Euler para representação da dinâmica

de orientação, por utilizar relações trigonométricas, o custo computacional torna-se alto. Como alternativa, este trabalho propõe a utilização da parametrização por quaternion, visto que se torna vantajoso em comparação aos Ângulos de Euler, pois altera as não linearidades trigonométricas por não linearidades polinomiais, reduzindo o esforço computacional envolvido e evitando o problema de *Gimbal Lock* (DEBRUIN, 2008).

1.1 Estado da Arte

No que tange ao problema de estimativa de pose de sistemas 6-DOF, é possível observar que os trabalhos encontrados na literatura partem de abordagens diversas de tratamento, apresentando distinções nas tecnologias de detecção e nas técnicas de processamento. Considerando as tecnologias de detecção, três diferentes abordagens são geralmente consideradas: baseadas em sensor, visão e sensor-visão. Por sua vez, no que se refere a técnica de processamento, várias são as pesquisas relacionadas à aceleração de algoritmos através do uso de arquiteturas paralelas.

Kumar, Behera e Bandyopadhyay (2017) propuseram o algoritmo *Super-Twisting Observer*, técnica que parametriza a dinâmica do sistema empregando o método de Lagrange, utilizando apenas os comprimentos dos atuadores para garantir a convergência finita no tempo do erro de estimativa dos estados. Embora inovadora, por depender unicamente de um sensor, esta técnica se torna frágil, ao passo que amplia as chances de erros de estimativa.

Sob outro viés, Miletović et al. (2017) apresentam um método para reconstrução dos estados de um sistema de interesse baseado no *Unscented Kalman Filter* (UKF). Esta abordagem é baseada nas medidas das posições dos atuadores e de uma IMU. Ainda que considerada mais eficiente, esta metodologia demanda um maior esforço computacional quando comparada ao algoritmo EKF.

Por sua vez, Yang et al. (2009) apresentam o algoritmo *Global Newton-Raphson with Monotonic Descent* (GNRMD), um método numérico baseado em *Newton-Raphson* para a estimação em tempo real da orientação e posição. Neste trabalho é utilizada a parametrização por Ângulos de Euler, com uma abordagem baseada na configuração geométrica do manipulador para inferir a pose. Porém, este algoritmo requer condições iniciais conhecidas a fim de evitar a instabilidade do estimador.

Similar à metodologia proposta no presente trabalho, Mirzaei e Roumeliotis (2008) apresentam a formulação de um EKF para estimar a pose de um sistema 6-DOF, empregando uma câmera e uma IMU. Entretanto, este algoritmo suporta apenas aplicações *offline*.

Partindo do pressuposto de que algumas soluções requerem alto poder de proces-

samento para funcionar corretamente, Lee e Salcic (1997) foram os primeiros a propor um Filtro de Kalman implementado em um FPGA para sistemas de rastreamento de múltiplos alvos (MTT). Ao substituírem o software tradicional por um algoritmo baseado em hardware, aprimoraram o desempenho em duas a três ordens de magnitude, de modo que foi atingido um tempo de ciclo de 1.824 microssegundos.

Por fim, Guo et al. (2012) desenvolveram um algoritmo EKF na linguagem *C* em um processador Nios II com frequência de processamento de 100 MHz, embarcado em um dispositivo FPGA. Com o objetivo de tornar essa abordagem eficiente, adotaram instruções personalizadas de ponto flutuante com precisão única e otimização do algoritmo.

Diferentemente dos trabalhos supracitados, para aplicação em sistemas de seis graus de liberdade e visando aplicações *online*, o presente estudo propõe a formulação de um Filtro Kalman Estendido baseado em quaternion. Esta abordagem faz uso de um processador comercial facilmente encontrado em sistemas embarcados de baixo custo e alta disponibilidade no mercado (i.e., ARM Cortex-M3) (NXP, 2015; TI, 2011), o que facilita sua reutilização em outros sistemas reais. A topologia escolhida foi a baseada no sensor-visão, que contém: uma IMU com um giroscópio e um acelerômetro; e uma câmera que usa marcadores com cores diferentes e geometria simples para minimizar o custo computacional. Ademais, a presente proposta alcança uma eficiente execução ao empregar frequência de processamento de apenas 25MHz, reduzindo ainda mais o consumo de energia do sistema.

1.2 Organização do Manuscrito

Esta dissertação está organizada da seguinte maneira. O Capítulo 2 apresenta conceitos teóricos fundamentais para compreender a proposta deste trabalho. Este capítulo apresenta as principais relações matemáticas a cerca dos quaternions, da Plataforma Stewart, dos sensores utilizados, bem como o Filtro de Kalman Estendido. Na sequência, o Capítulo 3 inicia a descrição da metodologia proposta mostrando como é feito a aquisição de dados e a validação dos estados estimados, bem como a descrição do *hardware* utilizado. O Capítulo 4 é dedicado ao projeto numérico e na descrição de como foi realizado o processo de embarcar o filtro proposto. O Capítulo 5 apresenta os resultados da aplicação em tempo real, além de uma análise estatística dos dados estimados pelo filtro. Por fim, o Capítulo 6 traz as considerações finais do trabalho, sintetizando os conceitos mais importantes e resumindo as conclusões dos resultados de simulação encontrados, além de apresentar propostas para o desenvolvimento de trabalhos futuros.

2 Preliminares

Neste capítulo serão apresentados os principais conceitos e definições empregados ao longo do presente trabalho. Para isso, esta seção foi organizada da seguinte maneira: (i) breve apresentação das operações matemáticas com quaternions voltadas às operações de rotação no espaço tridimensional e também aos Ângulos de Euler equivalentes; (ii) análise sobre a definição da geometria de uma Plataforma Stewart e de sua dinâmica de movimento; (iii) apresentação dos modelos matemáticos que descrevem os sensores inerciais; (iv) descrição do modelo dinâmico da câmera; e por fim, (v) apresentação do Filtro de Kalman Estendido.

2.1 Quaternions

Quaternion pode ser interpretado como uma ferramenta matemática utilizada para realizar rotações no espaço tridimensional como alternativa aos Ângulos de Euler. Quando comparados com os tradicionais Ângulos de Euler, os quaternions apresentam vantagem computacional, pois trabalham com não linearidades polinomiais ao invés de não linearidades trigonométricas. Desta forma, acabam evitando o problema conhecido como *Gimbal Lock*, que acontece devido às singularidades nas matrizes de rotações, onde fisicamente representam a perda de algum grau de liberdade em virtude do alinhamento de um ou mais eixos do sistema. Sua forma vetorial $\mathbf{q} \in \mathbb{R}^4$ pode ser representada como:

$$\mathbf{q} = \begin{bmatrix} \eta \\ \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{bmatrix} = \begin{bmatrix} \eta \\ \boldsymbol{\varepsilon} \end{bmatrix}. \quad (2.1)$$

Pertencendo ao conjunto de números hipercomplexos temos $\eta \in \mathbb{R}$ como o componente real do quaternion e $\boldsymbol{\varepsilon} \in \mathbb{R}^3$ o componente imaginário (DIEBEL, 2006). Para parametrizar a orientação de um corpo rígido, é utilizado a propriedade em que a norma Euclidiana do quaternion deve ser unitária, $\|\mathbf{q}\| = \sqrt{\eta^2 + \boldsymbol{\varepsilon}^T \boldsymbol{\varepsilon}} = 1$.

Em termos de ângulo-eixo, um quaternion também pode ser expressado como:

$$\mathbf{q} = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ \mathbf{r} \sin\left(\frac{\theta}{2}\right) \end{bmatrix}. \quad (2.2)$$

Fisicamente a equação (2.2) representa uma rotação θ na direção $\mathbf{r} \in \mathbb{R}^3$. Neste sentido, pode-se definir uma matriz de rotação baseada em quaternions, ou seja, temos

que $R(\mathbf{q}) \in \mathbb{R}^{3 \times 3}$ é definida por:

$$R(\mathbf{q}) = I_3 + 2\eta S(\boldsymbol{\varepsilon}) + 2S^2(\boldsymbol{\varepsilon}), \quad (2.3)$$

onde $I_3 \in \mathbb{R}^3$ é uma matriz identidade e o operador $S(\cdot) : \mathbb{R}^3 \rightarrow \mathbb{R}^{3 \times 3}$ definida como:

$$S(\mathbf{x}) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}, \quad (2.4)$$

em que, x_i é o i -ésimo termo do vetor \mathbf{x} .

Ainda segundo Diebel (2006), qualquer vetor $\mathbf{x} \in \mathbb{R}^3$ pode ser representado por meio de um quaternion na forma de:

$$\mathbf{x} = x_1i + x_2j + x_3k = \begin{bmatrix} 0 \\ \mathbf{x} \end{bmatrix}, \quad (2.5)$$

à vista disso, uma rotação de ângulo θ ao redor do vetor \mathbf{r} é dado pela seguinte expressão:

$$\mathbf{x}' = \mathbf{q}\mathbf{x}\mathbf{q}^{-1} \rightarrow \mathbf{x}' = R(\mathbf{q})\mathbf{x}. \quad (2.6)$$

Assim, de forma geral, a rotação de um vetor $\mathbf{x} \in \mathbb{R}^3$ no espaço tridimensional é representada como mostrada na Figura 1.

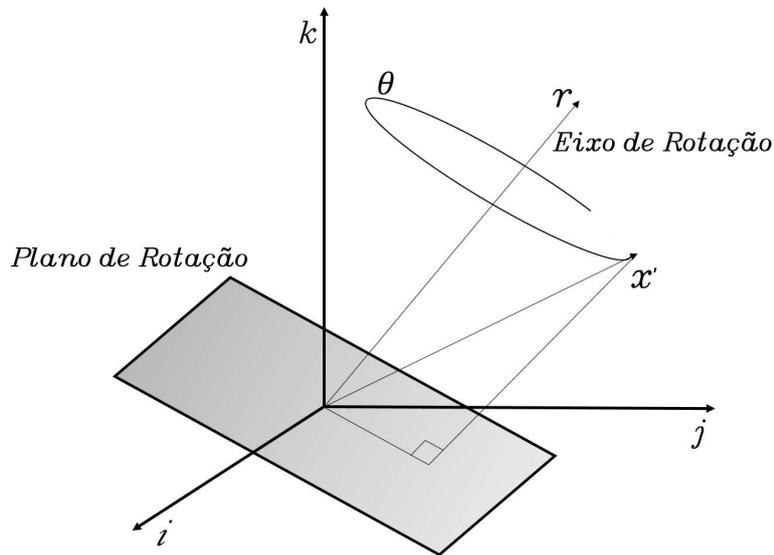
Por outro lado, os Ângulos de Euler, por serem bastante difundidos no meio acadêmico, são mais naturais e fáceis de serem entendidos e visualizados em comparação à representação da orientação de um corpo rígido utilizando quaternion. Portanto, com o objetivo de facilitar a visualização de um quaternion, têm-se os ângulos de Euler descritos como uma função de um quaternion de orientação \mathbf{q} que é dado por:

$$\begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} \arctan\left(\frac{2(\eta\varepsilon_1 + \varepsilon_2\varepsilon_3)}{1 - 2(\varepsilon_1^2 + \varepsilon_2^2)}\right) \\ \arctan(2(\eta\varepsilon_2 + \varepsilon_3\varepsilon_1)) \\ \arctan\left(\frac{2(\eta\varepsilon_3 + \varepsilon_1\varepsilon_2)}{1 - 2(\varepsilon_2^2 + \varepsilon_3^2)}\right) \end{bmatrix}, \quad (2.7)$$

e \mathbf{q} pode ser calculado através dos Ângulos de Euler α , β e γ pela seguinte expressão:

$$\mathbf{q} = \begin{bmatrix} \cos\left(\frac{\alpha}{2}\right)\cos\left(\frac{\beta}{2}\right)\cos\left(\frac{\gamma}{2}\right) - \sin\left(\frac{\alpha}{2}\right)\sin\left(\frac{\beta}{2}\right)\sin\left(\frac{\gamma}{2}\right) \\ \sin\left(\frac{\alpha}{2}\right)\cos\left(\frac{\beta}{2}\right)\cos\left(\frac{\gamma}{2}\right) - \cos\left(\frac{\alpha}{2}\right)\sin\left(\frac{\beta}{2}\right)\sin\left(\frac{\gamma}{2}\right) \\ \cos\left(\frac{\alpha}{2}\right)\sin\left(\frac{\beta}{2}\right)\cos\left(\frac{\gamma}{2}\right) - \sin\left(\frac{\alpha}{2}\right)\cos\left(\frac{\beta}{2}\right)\sin\left(\frac{\gamma}{2}\right) \\ \cos\left(\frac{\alpha}{2}\right)\cos\left(\frac{\beta}{2}\right)\sin\left(\frac{\gamma}{2}\right) - \sin\left(\frac{\alpha}{2}\right)\sin\left(\frac{\beta}{2}\right)\cos\left(\frac{\gamma}{2}\right) \end{bmatrix}. \quad (2.8)$$

Figura 1 – Representação gráfica de uma rotação utilizando quaternion no espaço 3D.



Fonte: Autor(2020).

2.2 Plataforma Stewart

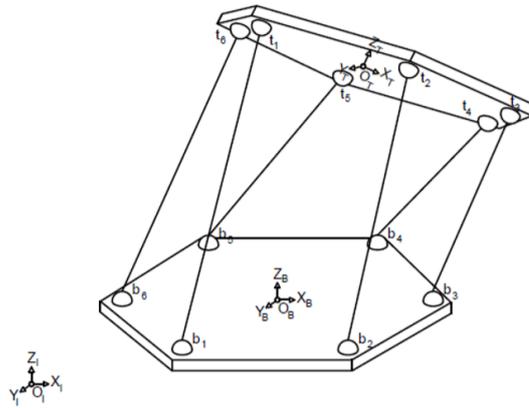
A Plataforma Stewart é formada por uma base fixa e outra móvel, conectadas entre si através de seis hastes ou atuadores lineares, o que torna este sistema capaz de se movimentar em todos os 6-DOF. Segundo Dasgupta e Mruthyunjaya (2000), a Plataforma Stewart, por ser um robô do tipo paralelo, apresenta diversas vantagens em relação à robôs com estrutura em série, quais sejam: rigidez mecânica; elevada precisão e baixa demanda de torque em relação à carga empregada sobre a base móvel (KUCUK, 2012). Devido às juntas utilizadas em ambas as bases – fixa e móvel – apenas restringir deslocamentos, não há adição de forças de flexão. Ademais, o fato do peso da carga ser distribuído entre as seis hastes, minimiza a potência e o tamanho necessário dos atuadores, bem como a rigidez estrutural.

2.2.1 Geometria da Plataforma de Stewart

Na Plataforma Stewart são definidos três sistemas de coordenadas: O_I , sistema de coordenada global ou de interesse; O_B , sistema de referência situado no centro da base fixa e O_T , sistema de coordenada localizado no topo da plataforma. Nesse sentido, a Figura 2 representa graficamente todos os sistemas de coordenadas definidos.

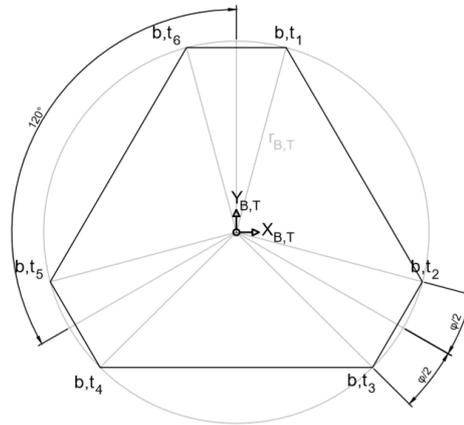
O manipulador possui $i = (1..6)$ hastes no qual cada uma das juntas b_i e t_i - arranjadas em pares e espaçadas por 120° entre si - pertencem a um vértice de um

Figura 2 – Plataforma Stewart e os sistemas de referências.



Fonte: Faria et al. (2016)

Figura 3 – Relações geométricas.



Fonte: Faria et al. (2016)

hexágono contido em um círculo de raio r_B e r_T , como observado na Figura 3 (FARIA et al., 2016). Desta forma, φ_B e φ_T também definem a distância entre as juntas, assim como o tamanho da borda da plataforma.

Na sequência, fora definida a posição das juntas b_i e t_i por meio dos vetores $\mathbf{B}_i \in \mathbb{R}^3$ para a base da plataforma e $\mathbf{T}_i \in \mathbb{R}^3$ para o topo da plataforma em relação a seus respectivos sistemas de coordenadas locais, os quais são expressos da seguinte forma:

$$\mathbf{B}_i = \begin{bmatrix} r_B \cos(\lambda_i) \\ r_B \sin(\lambda_i) \\ 0 \end{bmatrix}, \quad \mathbf{T}_i = \begin{bmatrix} r_T \cos(\psi_i) \\ r_T \sin(\psi_i) \\ 0 \end{bmatrix}, \quad (2.9)$$

em que,

$$\lambda_i = \begin{cases} \frac{i\pi}{3} - \frac{\varphi_B}{2} & i = (1, 3, 5), \\ \lambda_{i-1} - \varphi_B & i = (2, 4, 6), \end{cases} \quad (2.10)$$

$$\psi_i = \begin{cases} \frac{i\pi}{3} - \frac{\varphi_T}{2} & i = (1, 3, 5), \\ \psi_{i-1} - \varphi_T & i = (2, 4, 6). \end{cases}$$

2.2.2 Dinâmica da Plataforma Stewart

Para descrever a dinâmica da Plataforma Stewart é comumente utilizado o formalismo lagrangiano - baseado na energia do sistema - e o formalismo Newton-Euler, que utiliza as forças que envolvem o movimento da planta. Porém, neste trabalho utiliza-se o formalismo Newton-Euler baseado em quaternions. Desta forma, será utilizada uma descrição genérica de um corpo rígido tridimensional com relação ao sistema de coordenada cujo a origem coincide com o centro de massa do corpo. Sendo assim, considerando a abordagem feita na seção anterior, a equação que representa o movimento de rotação do topo da plataforma baseado em quaternion é dado por (MARKLEY; CRASSIDIS, 2014):

$$\dot{\mathbf{q}}(\eta, \boldsymbol{\varepsilon}, \boldsymbol{\omega}) = \frac{1}{2} \begin{bmatrix} -\boldsymbol{\varepsilon}^T \\ \eta \mathbf{I}_3 + S(\boldsymbol{\varepsilon}) \end{bmatrix} \boldsymbol{\omega}, \quad (2.11)$$

$$\boldsymbol{\tau}(\boldsymbol{\omega}, \dot{\boldsymbol{\omega}}) = \mathbf{I}_m \dot{\boldsymbol{\omega}} + S(\boldsymbol{\omega}) \mathbf{I}_m \boldsymbol{\omega},$$

onde, $\boldsymbol{\tau} \in \mathbb{R}^3$ é o vetor de torque, $\mathbf{I}_m \in \mathbb{R}^3$ é a matriz de inércia do corpo, $\boldsymbol{\omega} \in \mathbb{R}^3$ é o vetor de velocidade angular e $(\dot{\quad})$ representa a derivada no tempo.

Por fim, as equações da dinâmica de translação do topo da plataforma são definidas como:

$$\dot{\mathbf{p}}(\mathbf{v}) = \mathbf{v}, \quad (2.12)$$

$$\mathbf{F}(\dot{\mathbf{v}}) = M \dot{\mathbf{v}},$$

em que $\dot{\mathbf{p}} = [\dot{p}_x \ \dot{p}_y \ \dot{p}_z]^T \in \mathbb{R}^3$ é o vetor de velocidade do topo da plataforma em relação ao sistema de coordenada global, com \dot{p}_x , \dot{p}_y e \dot{p}_z relacionados à x -, y - e z -eixo, respectivamente. $M \in \mathbb{R}^{3 \times 3}$ é a matriz de massa da plataforma, $\mathbf{F} \in \mathbb{R}^3$ o vetor de forças inerciais aplicado ao topo da plataforma e $\dot{\mathbf{v}} \in \mathbb{R}^3$ o vetor de aceleração, sendo estes dois últimos também representados em relação ao sistema de coordenada global.

Destarte, adicionando a força de gravidade no modelo, a dinâmica de posição e

orientação do topo da plataforma é formulada como:

$$\begin{bmatrix} \dot{\mathbf{q}} \\ \dot{\boldsymbol{\omega}} \\ \dot{\mathbf{p}} \\ \dot{\mathbf{v}} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \begin{bmatrix} -\boldsymbol{\varepsilon}^T \\ \eta I_3 + S(\boldsymbol{\varepsilon}) \end{bmatrix} \boldsymbol{\omega} \\ I_m^{-1}(\mathbf{u}_\tau + \boldsymbol{\tau}_{ext} - S(\boldsymbol{\omega})I_m\boldsymbol{\omega}) \\ \mathbf{v} \\ M^{-1}(\mathbf{u}_F + \mathbf{F}_{ext}) + \mathbf{g} \end{bmatrix}, \quad (2.13)$$

no qual, $\mathbf{u}_\tau \in \mathbb{R}^3$ e $\boldsymbol{\tau}_{ext} \in \mathbb{R}^3$ são os torques de entrada e perturbação relacionados ao sistema de coordenada local, $\mathbf{u}_F \in \mathbb{R}^3$ e $\mathbf{F}_{ext} \in \mathbb{R}^3$ são as forças de entrada e perturbação relacionadas ao sistema de coordenada global e \mathbf{g} é o vetor de gravidade. Aqui observa-se a principal vantagem de parametrizar a orientação de um corpo rígido por quaternion, o qual permite a obtenção de um modelo dinâmico livre de singularidades, assim como nenhuma relação trigonométrica.

2.3 Modelo do Giroscópio e Acelerômetro

Uma vez que é possível ter uma orientação diferente entre o sistema de coordenadas globais e o sistema de coordenadas locais do sensor, se faz necessário um modelo capaz de relacionar a velocidade angular $\boldsymbol{\omega}_G$ medida pelo sensor com a velocidade angular $\boldsymbol{\omega}$ da plataforma, que tem como referência as coordenadas globais O_I . Tendo em vista as considerações abordadas e a parametrização da orientação com quaternions, pode-se definir a matriz de rotação R_{imu} para formalizar a relação entre a velocidade angular nas coordenadas globais $\boldsymbol{\omega}$ e as medidas do sensor $\boldsymbol{\omega}_G$. Deste modo, representa-se o modelo do giroscópio da seguinte forma:

$$\boldsymbol{\omega}_G = R_{imu}\boldsymbol{\omega} + \mathbf{b}_G + \boldsymbol{\delta}_G, \quad (2.14)$$

no qual R_{imu} descreve a orientação da IMU com relação à plataforma, $\mathbf{b}_G \in \mathbb{R}^3$ uma constante da diferença entre o valor medido e o valor real, representando o *offset* do sensor, e por fim, $\boldsymbol{\delta}_G$ um processo estocástico representando o ruído branco Gaussiano do sensor, com média zero e covariância Q_G .

Por sua vez, o acelerômetro se faz necessário para medir as acelerações relativas as translações que atuam no sistema em conjunto com a aceleração da gravidade. Para a correta descrição do modelo matemático do sensor, consideram-se as forças inerciais decorrentes de uma rotação quando o mesmo está localizado fora da origem do sistema (OLENICK; APOSTOL; GOODSTEIN, 2008), denominadas de força centrífuga e a força

de Coriolis. A dinâmica completa do acelerômetro é dado por:

$$\mathbf{a}_A = R_{imu}R^T(\mathbf{q})((\mathbf{a} + \mathbf{g}) + H(\boldsymbol{\omega}, \boldsymbol{\alpha})\mathbf{O}_{imu}) + \mathbf{b}_A + \boldsymbol{\delta}_A, \quad (2.15)$$

onde temos $R_{imu}R^T(\mathbf{q})$ é a matriz de rotação responsável pela transformação do sistema de coordenadas globais da plataforma para o sistema de coordenadas locais do sensor, $\mathbf{a} \in \mathbb{R}^3$ é a aceleração no centro do corpo rígido, e a função $H(\boldsymbol{\omega}, \boldsymbol{\alpha})$ representa os efeitos das força inerciais, dadas por:

$$H(\boldsymbol{\omega}, \boldsymbol{\alpha}) = S(\boldsymbol{\omega})^2 + S(\boldsymbol{\alpha}), \quad (2.16)$$

$\mathbf{O}_{imu} \in \mathbb{R}^3$ representa o *offset* do sensor em relação à origem, e finalmente, $\mathbf{b}_A \in \mathbb{R}^3$ é o *offset* das medidas do sensor e $\boldsymbol{\delta}_A$ um processo estocástico representando o ruído branco Gaussiano do sensor, com média zero e covariância Q_A .

2.4 Modelo da Câmera

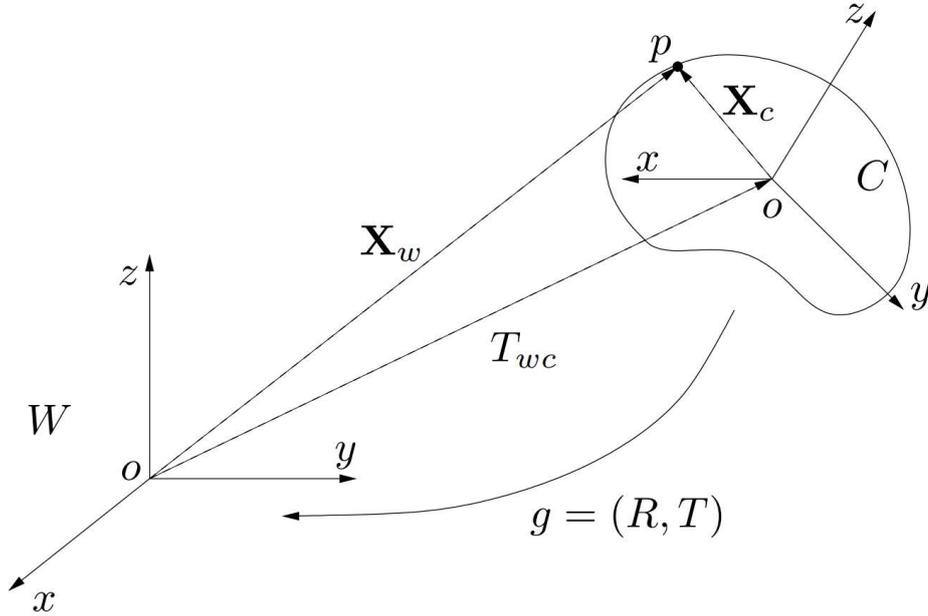
Antes de apresentar o Filtro de Kalman Estendido e seu funcionamento, o qual necessita de dois passos para sua implementação: (1) a predição, que é baseada no modelo do sistema; e (2) a correção, que tem como objetivo de corrigir a estimativa fornecida pelo modelo no passo anterior. Devemos abordar a correção dos estados estimados pelo Filtro de Kalman Estendido, que neste trabalho será realizado por uma câmera. Neste sentido, definimos abaixo algumas relações matemáticas de modo a extrair as informações das imagens. Além disso, para a correta modelagem matemática da câmera devemos considerar seus parâmetros intrínsecos e extrínsecos.

2.4.1 Parâmetros Extrínsecos

Os parâmetros extrínsecos indicam a posição e orientação da câmera nas coordenadas globais e são definidos como qualquer arranjo de parâmetros geométricos que identifiquem a transformação entre o sistema de coordenadas locais da câmera e o sistema de coordenadas globais (TRUCCO; VERRI, 1998). Sabendo que esta transformação de sistemas de coordenadas de um corpo rígido é de natureza não linear e afins, devemos converter a transformação afim para uma representação linear. Portanto, a ferramenta que deve-se utilizar é as Coordenadas Homogêneas, que consiste em uma extensão do espaço euclidiano \mathbb{E}^3 ao hiperplano \mathbb{R}^4 (MA et al., 2012).

Considere a Figura 4, que representa o movimento de um objeto no espaço euclidiano \mathbb{E}^3 . Para utilizar as coordenadas homogêneas, basta adicionar o elemento 1 no vetor $\mathbf{X} = [X_1, X_2, X_3]^T \in \mathbb{R}^3$ de um ponto $\mathbf{p} \in \mathbb{E}^3$, desta forma obtemos um vetor $\tilde{\mathbf{X}} \in \mathbb{R}^4$.

Figura 4 – Movimento de um corpo rígido entre o sistema de coordenadas da câmera C e o sistema de coordenadas global W.



Fonte: (MA et al., 2012).

Assim, conseguimos fazer a transformação de uma representação afim para uma representação linear, que na forma matricial é definida como:

$$\tilde{\mathbf{X}} = \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \in \mathbb{R}^4. \quad (2.17)$$

Ainda segundo Ma et al. (2012), podemos definir a orientação e posição de um objeto a partir desta mesma abordagem - das coordenadas homogêneas - através da relação:

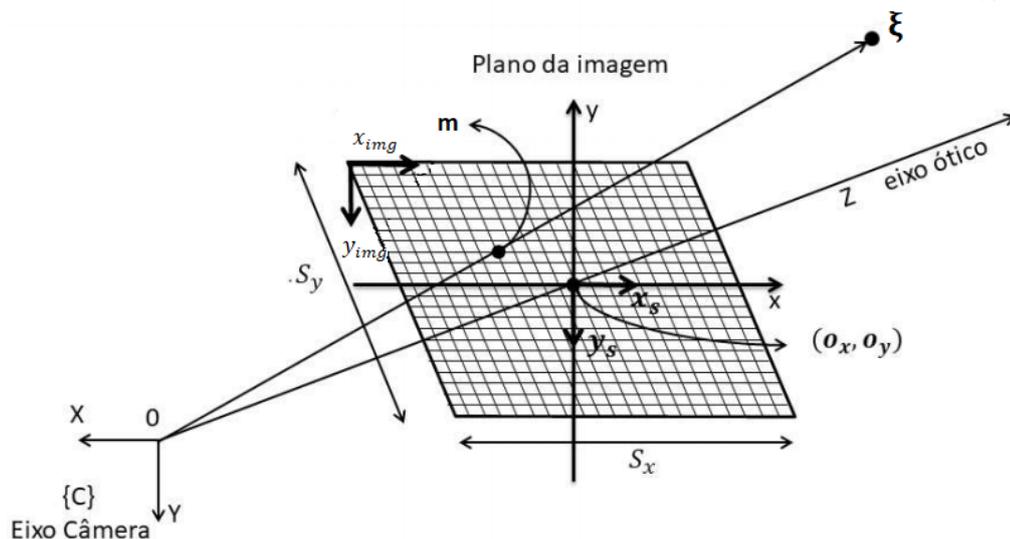
$$\tilde{\mathbf{X}}_w = \begin{bmatrix} \mathbf{X}_w \\ 1 \end{bmatrix} = \bar{g}_{wc} \tilde{\mathbf{X}}_c = \begin{bmatrix} R_{wc} & \mathbf{T}_{wc} \\ 0_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{X}_c \\ 1 \end{bmatrix} \in \mathbb{R}^4, \quad (2.18)$$

onde $R_{wc} \in \mathbb{R}^{3 \times 3}$ é a matriz de rotação do sistema de coordenadas locais C para o sistema de coordenadas globais W, $\mathbf{T}_{wc} \in \mathbb{R}^3$ é o vetor de translação entre a origem do sistema de coordenadas W e o sistema de coordenadas da câmera C, e $\mathbf{X}_c \in \mathbb{R}^3$ o vetor de posição do objeto em relação ao sistema de coordenadas da câmera.

De modo geral a representação homogênea $\bar{g} \in \mathbb{R}^{4 \times 4}$ de um corpo rígido pode ser representado como:

$$\bar{g} = \begin{bmatrix} R & \mathbf{T} \\ 0_{1 \times 3} & 1 \end{bmatrix} \in \mathbb{R}^{4 \times 4}. \quad (2.19)$$

Figura 5 – Relação entre o plano da retina da câmera e a matriz de pixels do plano de imagem.



Fonte: Adaptado de (VASQUEZ et al., 2015).

2.4.2 Parâmetros Intrínsecos

Quando o objetivo é uma medida mais precisa, precisamos considerar no modelo matemático as características óticas, geométricas e digitais da câmera utilizada (VASQUEZ et al., 2015). Considerando a Figura 5, temos: (o_x, o_y) sendo o ponto central do plano de imagem; (s_x, s_y) os valores de escalonamento da imagem no eixo x, y ; (x_{img}, y_{img}) as novas coordenadas calculadas em *pixels*; $\mathbf{m} \in \mathbb{R}^2$ o ponto projetado no plano de imagem do ponto $\xi = (X, Y, Z)$; e Z sendo o eixo ótico que intersecciona o centro da imagem.

Para calcular as coordenadas \mathbf{m} no plano de imagem, deve-se, principalmente, especificar a unidade ao longo dos eixos (x_{img}, y_{img}) . Ma et al. (2012) definem uma matriz de escalonamento no seguinte formato:

$$\begin{bmatrix} x_s \\ y_s \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \quad (2.20)$$

Contudo, as coordenadas (x_s, y_s) ficam especificadas em relação ao ponto de intersecção do eixo ótico e o plano de imagem, logo, a solução é aplicar uma translação na origem da referência da imagem e inverter os eixos:

$$\begin{aligned} \mathbf{x}_{img} &= -(x_s - o_x), \\ \mathbf{y}_{img} &= -(y_s - o_y). \end{aligned} \quad (2.21)$$

Aplicando a equação 2.20 em 2.21 e empregando a coordenada homogênea temos:

$$\begin{bmatrix} \mathbf{x}_{img} \\ \mathbf{y}_{img} \\ 1 \end{bmatrix} = \begin{bmatrix} -s_x & 0 & o_x \\ 0 & -s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \quad (2.22)$$

Desta forma a equação 2.22 pode ser reescrita como a matriz $K_s \in \mathbb{R}^{3 \times 3}$ considerando o ângulo de inclinação θ entre os eixos (x_s, y_s) :

$$K_s = \begin{bmatrix} -s_x & -s_\theta & o_x \\ 0 & -s_y & o_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (2.23)$$

onde o termo $s_\theta \in \mathbb{R}$ é proporcional à $\cot(\theta)$, ou seja, quando temos $\theta = 0$ temos um pixel retangular, portanto, $s_\theta = 0$.

Outro fator importante para ser considerado junto a estes parâmetros são a distância focal f , que representa a distância entre a origem do sistema de coordenadas da câmera e o ponto de intersecção do eixo ótico Z com o plano de imagem. Ma et al. (2012) definem a matriz $K_f \in \mathbb{R}^{3 \times 4}$ como sendo:

$$K_f = \begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (2.24)$$

Em suma os parâmetros intrínsecos de uma câmera podem ser representados como:

$$K_{int} = K_s K_f = \begin{bmatrix} -s_x & -s_\theta & o_x \\ 0 & -s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} f s_x & f s_\theta & o_x & 0 \\ 0 & f s_y & o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (2.25)$$

2.4.3 Modelo da Câmera *Pinhole*

Segundo Ma et al. (2012), se adicionarmos um sistema de coordenadas ao centro de projeção de imagem, com o eixo ótico alinhado com o eixo Z e utilizando o modelo de câmera *Pinhole*, a projeção das coordenadas \mathbf{X} para \mathbf{x} é dado por:

$$\mathbf{m} = \begin{bmatrix} x_{img} \\ y_{img} \end{bmatrix} = -\frac{f}{Z} \begin{bmatrix} X \\ Y \end{bmatrix}, \quad (2.26)$$

onde f representa a distância focal da câmera, X, Y, Z são coordenadas do ponto $\boldsymbol{\xi}$ no espaço tridimensional e x_{img}, y_{img} são as coordenadas do ponto \mathbf{m} projetado no plano de imagem. Deste modo, a representação homogênea é:

$$\left[f \frac{X}{Z}, f \frac{Y}{Z}, 1 \right] \in \mathbb{R}^3. \quad (2.27)$$

Podemos reescrever a equação 2.26 utilizando a representação homogênea como:

$$Z \begin{bmatrix} x_{img} \\ y_{img} \\ 1 \end{bmatrix} = \begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad (2.28)$$

em que $\mathbf{X} \doteq [X, Y, Z, 1]^T$ é a representação homogênea do ponto $\boldsymbol{\xi}$ no espaço tridimensional e $\bar{\mathbf{m}} \doteq [x \ y \ 1]^T$ é a projeção homogênea do ponto \mathbf{m} no plano de imagem. O parâmetro Z é geralmente desconhecido, por conseguinte fazemos uma troca de variável $Z = \lambda$, onde $\lambda \in \mathbb{R}_+$.

Considerando a transformação de coordenadas abordadas na seção de parâmetros extrínsecos, temos que:

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} R_{cam} & \mathbf{T} \\ 0_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix}, \quad (2.29)$$

onde $R_{cam} \in \mathbb{R}^{3 \times 3}$ é matriz de rotação que descreve a orientação da câmera em relação a orientação do corpo rígido.

Uma vez que para ter uma representação fidedigna das coordenadas globais $\mathbf{X}_0 = [X_0 \ Y_0 \ Z_0 \ 1]^T$ para as coordenadas \mathbf{m} (em *pixels*) correspondentes no plano de imagem, deve-se introduzir as características óticas, geométricas e digitais da câmera utilizada. Deste modo, reescrevendo a equação 2.28, a representação ideal para a câmera é dado por:

$$\lambda \begin{bmatrix} x_{im} \\ y_{im} \\ 1 \end{bmatrix} = \begin{bmatrix} fs_x & fs_\theta & o_x & 0 \\ 0 & fs_y & o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{cam} & \mathbf{T} \\ 0_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix}, \quad (2.30)$$

ou em sua forma matricial,

$$\lambda \bar{\mathbf{m}} = K_{int} \bar{g} \bar{\mathbf{X}}_0. \quad (2.31)$$

Considerando o sistema em estudo, temos quatro marcadores com posições conhecidas no sistema de coordenadas do corpo, que são representados pelo vetor $\boldsymbol{\xi}_i \in \mathbb{R}^3$ e sua projeção no plano de imagem pelo vetor \mathbf{m}_i representado em pixels. Utilizando a abordagem apresentada nas seções acima, podemos parametrizar a orientação da câmera através dos quaternions. De acordo com Mariottini e Prattichizzo (2005), o modelo que representa a formação da imagem projetada de um ponto $\boldsymbol{\xi}_i$ por uma câmera com a orientação \mathbf{q} e posição $\mathbf{p} \in \mathbb{R}^3$ é dado pela seguinte expressão:

$$\lambda \bar{\mathbf{m}}_i = K_{int} R_{cam} R^T(\mathbf{q})(\boldsymbol{\xi}_i - \mathbf{p}). \quad (2.32)$$

Como λ é um termo desconhecido e temos interesse em calcular a posição \mathbf{p} , utilizamos o operador $S(\cdot)$ para eliminarmos este termo da equação, pois $S(\bar{\mathbf{m}}_i)\bar{\mathbf{m}}_i = 0$. Considerando o processo estocástico δ_{cam} , a equação 2.32 pode ser reescrita da seguinte forma:

$$0 = S(\bar{\mathbf{m}}_i)K_{int}R_{cam}R^T(\mathbf{q})(\xi_i - \mathbf{p}) + \delta_{cam}. \quad (2.33)$$

2.5 Filtro de Kalman Estendido

O Filtro de Kalman foi proposto inicialmente por Kalman (1960), como um conjunto de equações matemáticas que proporcionam uma solução computacional eficiente para o método dos mínimos quadrados (WELCH; BISHOP et al., 1995). Como demonstrado por Welch, Bishop et al. (1995), ainda que não temos um modelo matemático preciso da natureza do sistema, podemos estimar os estados passado, presente e futuro da planta de forma precisa.

O filtro faz a fusão de um modelo com os sensores, funcionando em dois passos distintos: predição e correção. De acordo com Anderson e Moore (2012) a predição dos estados se baseiam no modelo do sistema, sendo este representado por um sistema linear, que está sujeito a um ruído Gaussiano aditivo,

$$\begin{aligned} \mathbf{x}_k &= A_k\mathbf{x}_{k-1} + B_k\mathbf{u}_k + \boldsymbol{\epsilon}_k, \\ \mathbf{z}_k &= C_k\mathbf{x}_k + \boldsymbol{\delta}_k, \end{aligned} \quad (2.34)$$

sendo $\mathbf{x}_k \in \mathbb{R}^n$ os estados do sistema, $\mathbf{u}_k \in \mathbb{R}^m$ as entradas do sistema, $\mathbf{z}_k \in \mathbb{R}^p$ as medições dos sensores, as matrizes A_k , B_k e C_k descrevem o sistema, e os sinais $\boldsymbol{\delta}_k$ e $\boldsymbol{\epsilon}_k$ são ruídos gaussianos com média zero e covariância Q_z inerente ao sensor e a incerteza do modelo que descreve o sistema com covariância Q_x , respectivamente.

A Tabela 1 descreve matematicamente o funcionamento do Filtro de Kalman, retornando a estimativa dos estados bem como a covariância desta estimativa, que representa a confiança da estimação \mathbf{x}_k .

Tabela 1 – Algoritmo do Filtro de Kalman.

Predição
1: $\bar{\mathbf{x}}_k = A_k\mathbf{x}_{k-1} + B_k\mathbf{u}_k$
2: $\bar{P}_k = A_kP_{k-1}A_k^T + Q_x$
Correção
3: $K_k = \bar{P}_kC_k^T(C_k\bar{P}_kC_k^T + Q_z)^{-1}$
4: $\mathbf{x}_k = \bar{\mathbf{x}}_k + K_k(\mathbf{z}_k - C(\bar{\mathbf{x}}_k))$
5: $P_k = (I - K_kC_k)\bar{P}_k$

Predição: as linhas 1 e 2 do algoritmo são utilizadas para prever o estado atual do sistema e a sua covariância antes de realizar uma medição. Elas se baseiam apenas no modelo do sistema, portanto, quanto melhor for o modelo, melhor será esta predição.

Correção: a correção acontece entre as linhas 3 e 5, onde o chamado “ganho de Kalman” K_k é calculado. É através deste ganho que a medição \mathbf{z}_k é incorporada na estimativa, o que ocorre na linha 4. O erro entre a predição $\bar{\mathbf{x}}_k$ e a medição \mathbf{z}_k é corrigido por este ganho. Por fim, uma nova covariância P_k é calculada e armazenada para o próximo passo do algoritmo.

Contudo, este trabalho aborda o modelo não linear do sistema, sendo assim, é necessário a utilização de uma extensão deste filtro, o Filtro de Kalman Estendido, pois este relaxa a suposição de que as equações do sistema são lineares, deixando a descrição do espaço de estados não mais limitada à (2.34) e se torna:

$$\begin{aligned}\mathbf{x}_k &= f(\mathbf{u}_k, \mathbf{x}_{k-1}) + \boldsymbol{\epsilon}_k \\ \mathbf{z}_k &= h(\mathbf{x}_k) + \boldsymbol{\delta}_k\end{aligned}\quad (2.35)$$

onde a função não linear $f(\cdot)$ relaciona o estado no passo $(k-1)$ com o estado no passo (k) , incluindo como parâmetro qualquer função \mathbf{u}_k , e a função não linear $h(\cdot)$ relaciona o estado \mathbf{x}_k com as medições \mathbf{z}_k (WELCH; BISHOP et al., 1995).

Para poder trabalhar com esta classe de sistema, utiliza-se a aproximação de Taylor de primeira ordem, isto é, aproxima-se a função não linear $f(\cdot)$ por,

$$f(\mathbf{u}_k, \mathbf{x}_{k-1}) \approx f(\mathbf{u}_k, \bar{\mathbf{x}}_{k-1}) + \underbrace{\frac{\partial f(\mathbf{u}_k, \bar{\mathbf{x}}_{k-1})}{\partial \bar{\mathbf{x}}_{k-1}}}_{=: F_k} (\mathbf{x}_{k-1} - \bar{\mathbf{x}}_{k-1}) \quad (2.36)$$

sendo as derivadas parciais F_k chamadas de Jacobiana e seus valores dependem de \mathbf{u}_k e $\bar{\mathbf{x}}_{k-1}$, desta forma essa matriz não é constante. Esta mesma aproximação é utilizada para

Tabela 2 – Algoritmo do Filtro de Kalman Estendido.

Predição
1: $\bar{\mathbf{x}}_k = f(\mathbf{u}_k, \bar{\mathbf{x}}_{k-1})$
2: $\bar{P}_k = F_k P_{k-1} F_k^T + Q_x$
Correção
3: $K_k = \bar{P}_k H_k^T (H_k \bar{P}_k H_k^T + Q_z)^{-1}$
4: $\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + K_k (\mathbf{z}_k - h(\hat{\mathbf{x}}_k))$
5: $P_k = (I - K_k H_k) \bar{P}_k$

as não linearidades envolvendo os sensores,

$$h(\mathbf{x}_k) \approx h(\bar{\mathbf{x}}_k) + \underbrace{\frac{\partial h(\bar{\mathbf{x}}_k)}{\partial \bar{\mathbf{x}}_k}}_{=: H_k} (\mathbf{x}_k - \bar{\mathbf{x}}_k) \quad (2.37)$$

O algoritmo resultante é apresentado na Tabela 2 e pode ser observado que é uma extensão do Filtro de Kalman original para sistema linearizados.

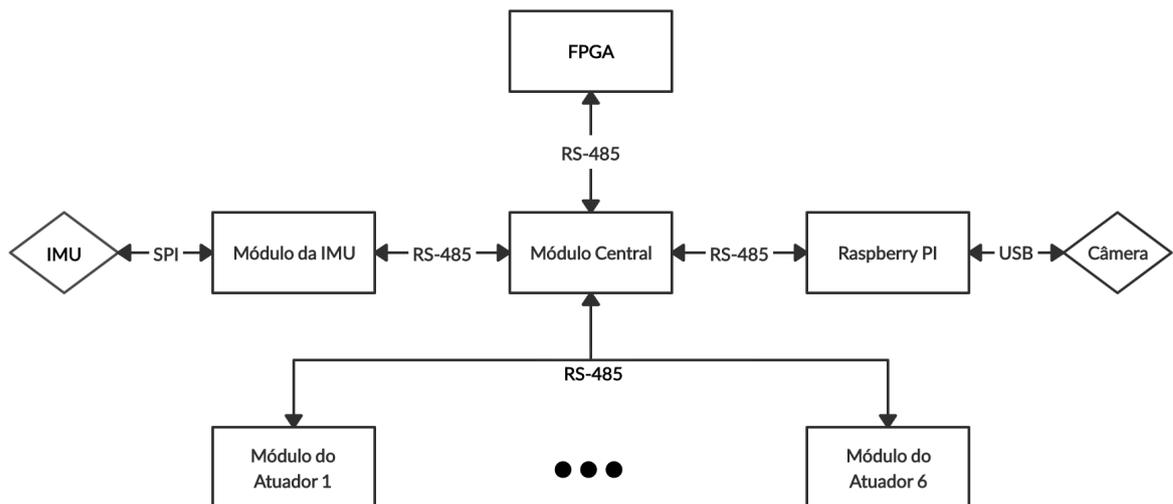
3 Sensoreamento e *Hardware*

Nesta seção serão vistos alguns conceitos relativos a aquisição de dados, sensoreamento e validação dos estados estimados, assim como descrito o hardware que forma o sistema da Plataforma Stewart.

3.1 Aquisição de Dados

A Plataforma Stewart utilizada está equipada com uma IMU e uma câmera para realizar a estimação da pose do topo da plataforma. Para inferir a pose através da cinemática direta de modo a comparar com os dados estimados, utiliza-se *encoders* dispostos junto aos motores DCs, responsáveis por acionarem os atuadores. A integração dos sensores e módulos pode ser vista na Figura 6.

Figura 6 – Processo de integração.



Fonte: Autor (2020).

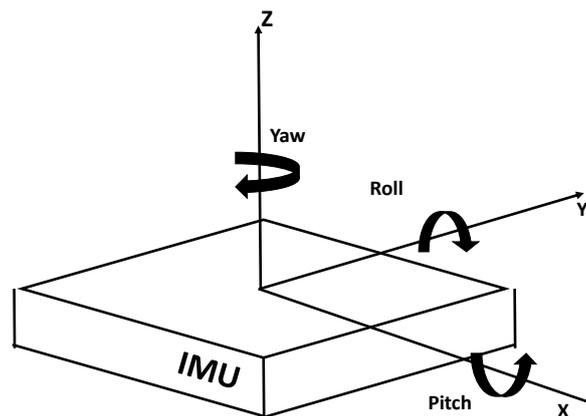
Toda comunicação entre os dispositivos responsáveis pelo processamento do filtro, controle de atuadores e processamento de imagem é gerenciada pelo módulo central e realizada através de um barramento RS-485. Portanto, foi desenvolvido um protocolo de comunicação para o controle de fluxo de dados. A sequência de aquisição é a seguinte: primeiramente é adquirido dados da IMU localizado no topo da plataforma; em seguida, as posições dos marcadores localizados na base da plataforma; e então os dados de posição dos atuadores através dos *encoders*. Uma vez realizada a aquisição, o módulo central envia os dados pré-processados para o FPGA realizar a estimação da pose do topo da plataforma.

Após cada iteração do EKF, é criado um *LOG* com todos os dados disponíveis, ou seja, dados da IMU, câmera, *encoders* e a pose estimada pelo filtro. Assim é possível fazer uma avaliação do EKF embarcado à posteriori.

3.1.1 IMU – Unidade de Medida Inercial

A Unidade de Medida Inercial é um dispositivo que une sensores capazes de medir velocidade angular, força de aceleração e também a orientação de um corpo rígido. Neste sentido, é utilizado geralmente um giroscópio, acelerômetro e um magnetômetro, respectivamente. Neste contexto, a orientação de um sistema de interesse pode ser obtida através da integração das medidas de velocidade angular inferida pelo giroscópio. Assim como a orientação, a velocidade e a posição linear podem ser obtidas através da integração do sinal proveniente do acelerômetro, sendo que a posição é preciso integrar o sinal duas vezes. A Figura 7 representa os graus de liberdade medidos pela IMU.

Figura 7 – Representação gráfica da IMU.



Fonte: Autor (2020).

Não obstante, estes dispositivos são capazes de trabalhar com uma frequência maior que 1kHz. Isso demonstra que são sensores bastante sensíveis, o que leva a ter uma medida com um ruído aditivo e um *offset*, também conhecido como *bias* (LI et al., 2015). Assim, quando integramos os sinais provenientes da IMU, temos uma divergência entre a pose inferida através do sensor e a pose real, o que se deve ao acúmulo de erros durante a aquisição de dados. Este fenômeno é conhecido como *Drift* do sinal e pode ser tratado utilizando outros sensores em conjunto, através de técnicas de fusão de sensores (XU; LI, 2007).

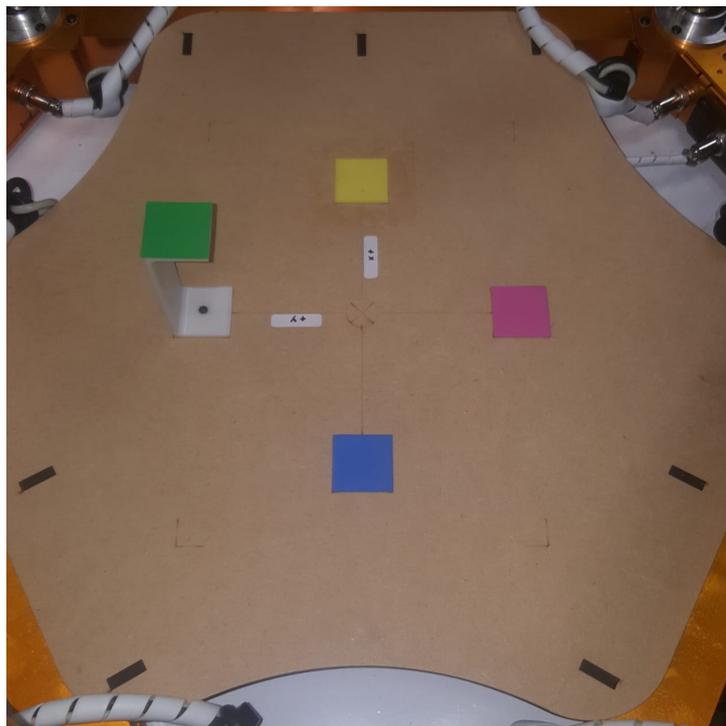
3.1.2 Visão Computacional

Trabalhos envolvendo topologias de sensoramento através de câmeras são muito abordados na literatura devido a grande quantidade de informações que uma imagem

fornece. O grande problema para realizar o processamento e análise de uma imagem é a demanda por um grande esforço computacional. A principal ferramenta e mais utilizada atualmente para lidar com imagens é o conjunto de bibliotecas OpenCV, que possuem algoritmos otimizados com diversas aplicações, como reconhecimento de faces e ações humanas, identificação de objetos, extração de modelos 3D, aplicação de filtros, etc. O presente trabalho considera o emprego de uma câmera para estimação da pose da plataforma.

Os dados provenientes da câmera são utilizados no passo de correção do EKF. Neste sentido, optou-se por utilizar uma *webcam* Logitech C270, com resolução HD de forma a ter a menor covariância possível nas medidas. Foram utilizados marcadores com cores diferentes e forma simples buscando diminuir o esforço computacional envolvido. A Figura 8 mostra como ficou a disposição dos marcadores, onde temos três marcadores no mesmo plano, entretanto em posições diferentes. Além disso, de forma a enriquecer a imagem com informações, foi alocado outro marcador com posicionamento e plano diferente dos demais.

Figura 8 – Disposição dos marcadores.



Fonte: Autor (2020).

O procedimento para aquisição da posição dos marcadores é realizado da seguinte forma: após fazer a aquisição da imagem é feito uma conversão de RGB¹ para HSV²; então

¹ Do inglês *Red*, *Green* e *Blue*, é um padrão baseado em três cores primárias: vermelho, verde e azul onde através da combinação destas três cores em diferentes intensidades é possível representar todas as outras cores que um olho humano é capaz de perceber.

² Do inglês *Hue*, *Saturation* e *Value*, é um padrão representado em função de tonalidade, saturação e

o marcador de interesse é selecionado de acordo com os valores de tonalidade, saturação e brilho; após este procedimento, a detecção de contorno pode ser realizada. Se a imagem possuir contornos, ou seja, se um marcador está no campo de visão da câmera, é possível então obter a coordenada do marcador através das propriedades de momentos invariantes da imagem.

Segundo Teague (1980) o plano de irradiação de uma imagem binária, denotado por $f(x, y)$, pode ser descrito em função de seus momentos invariantes, em que:

$$M_{00} = \sum_x \sum_y f(x, y), \quad (3.1)$$

representa o momento de ordem zero. Uma vez que queremos determinar o centroide dos marcadores no plano de imagem, devemos utilizar também o momento de ordem um, que é representado como:

$$M_{10} = \sum_x \sum_y xf(x, y), \quad (3.2)$$

$$M_{01} = \sum_x \sum_y yf(x, y). \quad (3.3)$$

Desta forma, para obter o centroide (\bar{x}, \bar{y}) , temos que:

$$\bar{x} = \frac{M_{10}}{M_{00}}, \quad \bar{y} = \frac{M_{01}}{M_{00}}. \quad (3.4)$$

No contexto deste trabalho, os valores dos centróides calculados são as posições dos marcadores em pixels no plano de imagem.

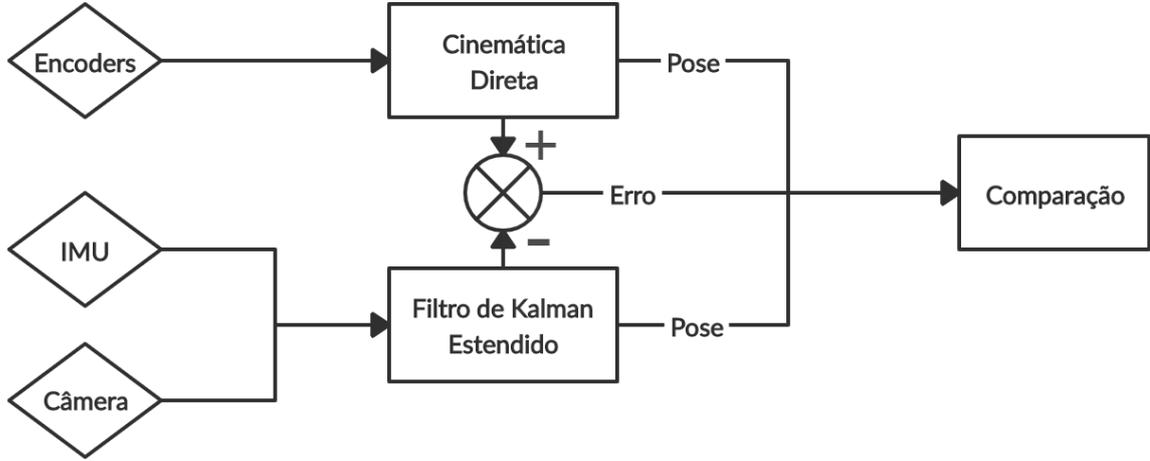
3.2 Validação dos Estados Estimados

Para realizar a validação da proposta deste trabalho, foi realizada uma comparação entre os dados estimados da pose do topo da Plataforma Stewart através do Filtro de Kalman Estendido com a pose inferida através da cinemática direta. A Figura 9 ilustra o processo de comparação entre estes dois sinais.

A cinemática direta é uma abordagem que considera a geometria do sistema de interesse, contudo, como resultado esta técnica leva a múltiplas soluções. Portanto, é necessário modelar a plataforma em função das restrições de movimento de cada parte móvel, e então utilizar métodos numéricos para encontrar a solução única (CARDONA, 2015). Este trabalho considera o uso do método numérico *Nelder-Mead* para realizar a

brilho, se assemelhando mais com percepção humana para representação do espectro de cores.

Figura 9 – Processo de validação dos estados estimados.



Fonte: Autor (2020).

minimização de uma função custo J do erro do valor medido em relação ao valor estimado para o tamanho dos atuadores, com objetivo de encontrar a solução única da cinemática direta.

Para realizar o processo de minimização foi utilizada a função `fminsearch` do *software* MATLAB, que a partir de um valor inicial é capaz de realizar a minimização de uma função custo. Isto posto, como entrada para a função é necessário fornecer um valor inicial de posição e orientação, que podem ser obtidas através das seguintes equações:

$$\begin{aligned} \mathbf{B}_{b,i}(\mathbf{p}_k, \mathbf{q}_k) &= \mathbf{B}_i R(\mathbf{q}_B) + \mathbf{p}_k^T, \\ \mathbf{T}_{b,i}(\mathbf{p}_k, \mathbf{q}_k) &= \mathbf{T}_i R(\mathbf{q}_T) + \mathbf{p}_k^T, \end{aligned} \quad (3.5)$$

onde $R(\mathbf{q}_B)$ e $R(\mathbf{q}_T)$ são as matrizes de rotações baseadas em quaternion entre a coordenada global e as coordenadas locais da base e do topo da plataforma, respectivamente. Os vetores \mathbf{B}_i e \mathbf{T}_i podem ser encontrados a partir de (2.9) e (2.10). Assim, pode-se inferir o valor estimado de tamanho dos atuadores \hat{L}_i , como:

$$\hat{L}_i = \sqrt{(\mathbf{T}_{b,i} - \mathbf{B}_{b,i})^T (\mathbf{T}_{b,i} - \mathbf{B}_{b,i})}. \quad (3.6)$$

Deste modo, temos

$$\boldsymbol{\theta}_k = \arg \min_{\boldsymbol{\theta}_k} J(\boldsymbol{\theta}_k) = \arg \min_{\boldsymbol{\theta}_k} \sum_{i=1}^6 (\mathbf{L}_i - \hat{L}_i(\boldsymbol{\theta}_k))^2, \quad (3.7)$$

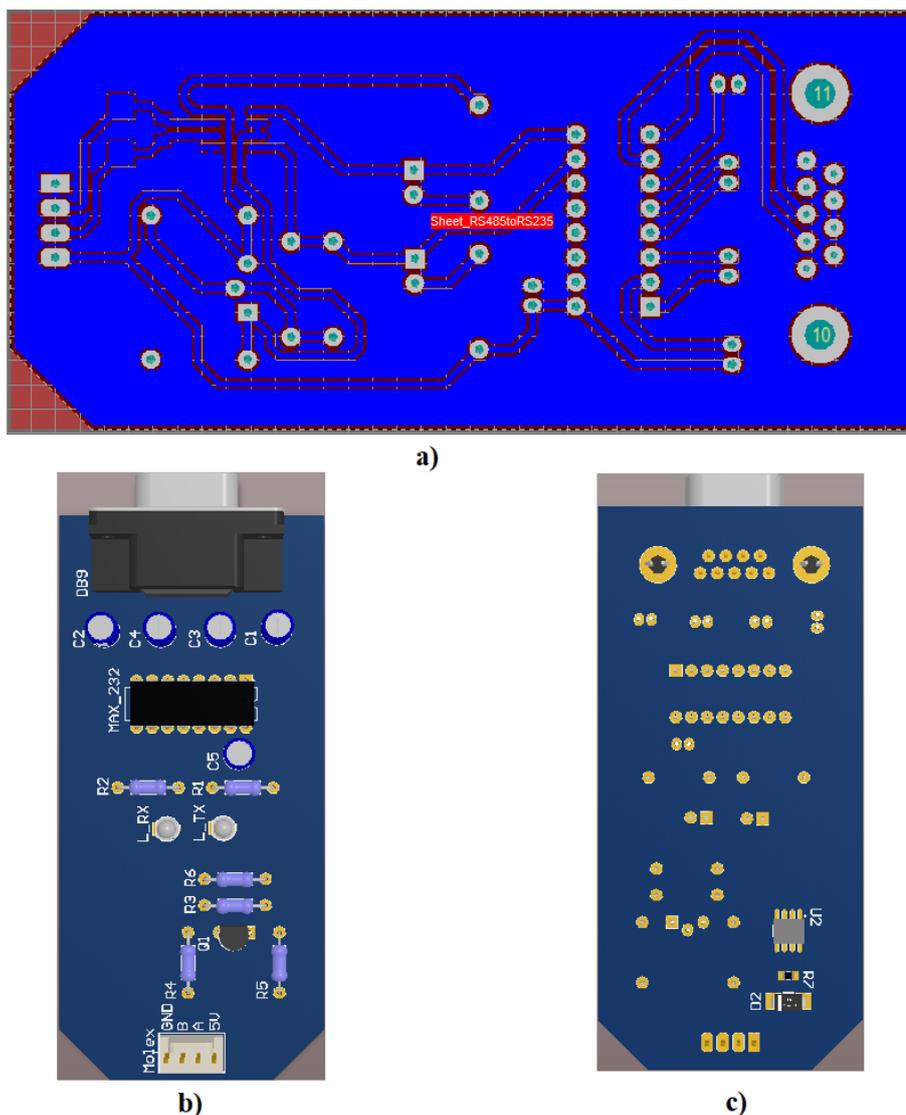
onde $\boldsymbol{\theta}$ se refere a pose estimada do topo da plataforma, ou seja, $\boldsymbol{\theta}_k = \begin{bmatrix} \hat{\mathbf{p}}_k^T & \hat{\mathbf{q}}_k^T \end{bmatrix}$. Por se

tratar de um problema de otimização não linear, em cada iteração o algoritmo considera a solução anterior θ_{k-1} como o ponto inicial para calcular θ_k .

3.3 Descrição do *Hardware*

Para embarcar o filtro desenvolvido neste trabalho, foi escolhido um processador comum implementado em um FPGA. Como citado anteriormente, toda comunicação entre os módulos é feita através de um barramento de comunicação RS-485. Neste sentido, para a integração do FPGA com a plataforma foi necessário o desenvolvimento de um conversor RS-232/RS-485. Foram utilizados um MAXIM232 e um ST485EC para o projeto do conversor e o resultado final pode ser visto na Figura 10.

Figura 10 – Placa de circuito impresso do conversor RS-232/RS-485: a) bottom layer, b) renderização do top layer, c) renderização do bottom layer.



Fonte: Autor (2020).

Os parâmetros geométricos da Plataforma Stewart utilizados neste trabalho são definidos conforme a Tabela 3.

Tabela 3 – Parâmetros Geométricos da Plataforma Stewart.

Descrição	Valor
r_B (Raio plataforma fixa)	0.35 [m]
r_T (Raio plataforma móvel)	0.25 [m]
φ_B (Ângulo de cada junta da plataforma fixa)	$\pi/6$ [rad]
φ_T (Ângulo de cada junta da plataforma móvel)	$\pi/2$ [rad]

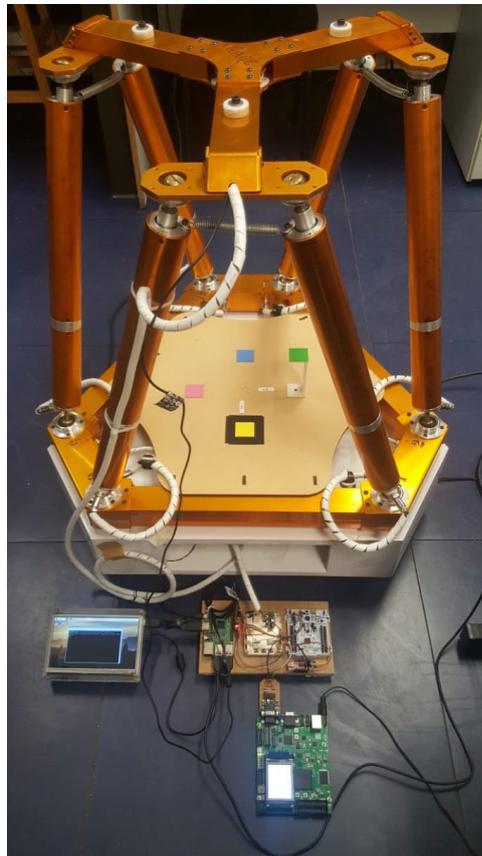
A lista de componentes utilizados neste trabalho podem ser vistas na Tabela 4.

Tabela 4 – Componentes Utilizados

Componente	Descrição
Câmera	Logitech C270 HD 720p
IMU	LSM6DS3H
FPGA	V2M-MPS2-0318C
Processador	ARM Cortex-M3 25MHz

A configuração do *hardware* e o sistema completo são mostrados na Figura 11.

Figura 11 – Experimental setup.



Fonte: Autor (2020).

4 Método de Estimação Proposto

Considerando os conceitos e definições vistos nos capítulos anteriores, nesta seção será apresentada todas as relações matemáticas necessárias para a formulação do Filtro de Kalman Estendido e o processo para embarcá-lo em um FPGA, utilizando um *soft-core* (ARM Cortex-M3).

4.1 Formulação do Filtro de Kalman Estendido

Para implementação do Filtro de Kalman Estendido, foram considerados os seguintes estados:

$$\hat{\mathbf{x}} := \left[\hat{\mathbf{q}}^T \quad \hat{\boldsymbol{\omega}}^T \quad \hat{\boldsymbol{\alpha}}^T \quad \hat{\mathbf{p}}^T \quad \hat{\mathbf{v}}^T \quad \hat{\mathbf{a}}^T \quad \hat{\mathbf{b}}^g \quad \hat{\mathbf{b}}^a \right]^T \in \mathbb{R}^{25} \quad (4.1)$$

onde,

- $\hat{\mathbf{q}} \in \mathbb{R}^4$ é o quaternion de orientação da plataforma;
- $\hat{\boldsymbol{\omega}} \in \mathbb{R}^3$ é o vetor de velocidade angular da plataforma;
- $\hat{\boldsymbol{\alpha}} \in \mathbb{R}^3$ é o vetor de aceleração angular da plataforma;
- $\hat{\mathbf{p}} \in \mathbb{R}^3$ é o vetor de posição da plataforma;
- $\hat{\mathbf{v}} \in \mathbb{R}^3$ é o vetor de velocidade linear da plataforma;
- $\hat{\mathbf{a}} \in \mathbb{R}^3$ é o vetor de aceleração linear da plataforma;
- $\hat{\mathbf{b}}^g \in \mathbb{R}^3$ é o vetor de *bias* do giroscópio;
- $\hat{\mathbf{b}}^a \in \mathbb{R}^3$ é o vetor de *bias* do acelerômetro.

Todos os estados à cima estão relacionados às coordenadas globais do sistema. Para simplificar a representação, assumimos que:

$$\dot{\hat{\mathbf{v}}} = \hat{\mathbf{a}}, \quad \dot{\hat{\mathbf{p}}} = \hat{\mathbf{v}}, \quad \dot{\hat{\boldsymbol{\omega}}} = \hat{\boldsymbol{\alpha}}, \quad \dot{\hat{\mathbf{q}}} = F(\hat{\boldsymbol{\omega}})\hat{\mathbf{q}} = G(\hat{\mathbf{q}})\hat{\boldsymbol{\omega}}. \quad (4.2)$$

onde,

$$F(\hat{\boldsymbol{\omega}}) = \frac{1}{2} \begin{bmatrix} 0 & -\hat{\boldsymbol{\omega}}^T \\ \hat{\boldsymbol{\omega}} & -S(\hat{\boldsymbol{\omega}}) \end{bmatrix} \quad e \quad G(\hat{\mathbf{q}}) = \frac{1}{2} \begin{bmatrix} -\hat{\boldsymbol{\epsilon}}^T \\ \hat{\eta}I_3 + S(\hat{\boldsymbol{\epsilon}}) \end{bmatrix}. \quad (4.3)$$

Isto posto, a equação de transição do modelo contínuo é descrito como:

$$\dot{\mathbf{x}} = f^c(\mathbf{x}) + \boldsymbol{\delta}_x^c, \quad (4.4)$$

onde $\dot{\mathbf{x}} \in \mathbb{R}^{25}$ é o vetor de estados de processo, f^c é o modelo do processo determinístico e $\boldsymbol{\delta}_x^c \in \mathbb{R}^{25}$ é o processo estocástico que modela as incertezas introduzidas na equação de transição com média zero e covariância Q_x , sendo:

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{q}} \\ \hat{\boldsymbol{\omega}} \\ \hat{\boldsymbol{\alpha}} \\ \hat{\mathbf{p}} \\ \hat{\mathbf{v}} \\ \hat{\mathbf{a}} \\ \hat{\mathbf{b}}^g \\ \hat{\mathbf{b}}^a \end{bmatrix} \quad e \quad f^c(\hat{\mathbf{x}}) = \hat{\mathbf{x}} + \begin{bmatrix} F(\hat{\boldsymbol{\omega}})\hat{\mathbf{q}} \\ \hat{\boldsymbol{\alpha}} \\ 0_{3 \times 1} \\ \hat{\mathbf{v}} \\ \hat{\mathbf{a}} \\ 0_{9 \times 1} \end{bmatrix}. \quad (4.5)$$

A matriz de covariância Q_x do processo contínuo é dado por:

$$Q_x^c = 10^{-5} I_{25 \times 25}. \quad (4.6)$$

Como já visto em seções anteriores, a dinâmica de orientação é intrinsecamente não linear e o Filtro de Kalman Estendido precisa que o modelo do sistema seja linearizado, portando, a cada iteração é feito uma linearização do modelo, que é dado pela matriz Jacobiana $F \in \mathbb{R}^{25 \times 25}$,

$$F(\hat{\mathbf{x}}) = \frac{\partial f(\hat{\mathbf{x}})}{\partial \hat{\mathbf{x}}} = \text{diag}\{\Phi(\hat{\mathbf{x}}), \Psi, 0_{9 \times 6}\}, \quad (4.7)$$

onde

$$\Phi(\hat{\mathbf{x}}) = \begin{bmatrix} \mathbb{F}(\hat{\boldsymbol{\omega}}) & \mathbb{G}(\hat{\mathbf{q}}) & 0_{4 \times 3} \\ 0_{3 \times 4} & 0_{3 \times 3} & I_{3 \times 3} \\ 0_{3 \times 4} & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix}, \quad \Psi = \begin{bmatrix} 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} \quad (4.8)$$

e

$$\mathbb{G}(\hat{\mathbf{q}}) = \frac{1}{2} \begin{bmatrix} -\hat{\boldsymbol{\epsilon}}^T \\ \hat{\eta}I + S(\hat{\boldsymbol{\epsilon}}) \end{bmatrix}. \quad (4.9)$$

Para seguir a proposta deste trabalho e embarcar o filtro é necessário discretizar estas relações matemáticas com um período de tempo T determinado. Para realizar a

discretização foi escolhido o método *Euler–Forward Discretization*, e o modelo discreto da equação de transição é dado por:

$$\hat{\mathbf{x}}_{k+1} = f(\hat{\mathbf{x}}_k) + \hat{\boldsymbol{\epsilon}}_{x,k}. \quad (4.10)$$

O modelo do processo determinístico discretizado e a matriz de covariância discreta é representada por:

$$f(\hat{\mathbf{x}}_k) = \hat{\mathbf{x}}_k + T \begin{bmatrix} F(\hat{\boldsymbol{\omega}}_k)\hat{\mathbf{q}}_k \\ \hat{\boldsymbol{\alpha}}_k \\ 0_{3 \times 1} \\ \hat{\mathbf{v}}_k \\ \hat{\mathbf{a}}_k \\ 0_{9 \times 1} \end{bmatrix}, \quad Q_x = T^2 Q_x^c. \quad (4.11)$$

Por fim, discretizando a equação (4.7) temos:

$$F(\hat{\mathbf{x}}_k) = \frac{\partial f(\hat{\mathbf{x}}_k)}{\partial \hat{\mathbf{x}}_k} = I_{25 \times 25} + T \text{diag}\{\Phi(\mathbf{x}_k), \Psi, 0_{9 \times 3}\}. \quad (4.12)$$

Cabe também definir as relações matemáticas em relacionadas as medições e como elas são reconstruídas a partir dos estados estimados. Isto posto, as medições utilizadas no passo de correção do filtro são:

- $\boldsymbol{\omega}_G \in \mathbb{R}^3$ é as medidas do giroscópio;
- $\mathbf{a}_A \in \mathbb{R}^3$ é as medidas do acelerômetro;
- $\boldsymbol{\xi}_1 \in \mathbb{R}^2$ é a posição em *pixels* do marcador 1 no plano de imagem;
- $\boldsymbol{\xi}_2 \in \mathbb{R}^2$ é a posição em *pixels* do marcador 2 no plano de imagem;
- $\boldsymbol{\xi}_3 \in \mathbb{R}^2$ é a posição em *pixels* do marcador 3 no plano de imagem;
- $\boldsymbol{\xi}_4 \in \mathbb{R}^2$ é a posição em *pixels* do marcador 4 no plano de imagem.

O modelo não linear das medições $z \in \mathbb{R}^{14}$ é expresso como:

$$\mathbf{z} = \mathbf{h}(\hat{\mathbf{x}}) + \boldsymbol{\delta}_z^c, \quad (4.13)$$

onde $h(\hat{\mathbf{x}})$ é formado pelas equações da dinâmica dos sensores (2.14, 2.15 e 2.32), e $\boldsymbol{\delta}_z^c \in$

\mathbb{R}^{14} é o processo estocástico que modela as incertezas introduzidas na equação de medição com média zero e covariância Q_z , sendo:

$$z = \begin{bmatrix} \boldsymbol{\omega}_G \\ \mathbf{a}_A \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \mathbf{h}_k(\mathbf{x}) = \begin{bmatrix} R_{imu}\hat{\boldsymbol{\omega}} + \hat{\mathbf{b}}_G \\ R_{imu}R^T(\hat{\mathbf{q}})((\hat{\mathbf{a}} + \hat{\mathbf{g}}) + H(\hat{\boldsymbol{\omega}}, \hat{\boldsymbol{\alpha}})\mathbf{O}_{imu}) + \hat{\mathbf{b}}_A \\ CC \cdot S(\bar{\mathbf{m}}_1)K_{int}R_{cam}R^T(\hat{\mathbf{q}})(\boldsymbol{\xi}_1 - \hat{\mathbf{p}}) \\ CC \cdot S(\bar{\mathbf{m}}_2)K_{int}R_{cam}R^T(\hat{\mathbf{q}})(\boldsymbol{\xi}_2 - \hat{\mathbf{p}}) \\ CC \cdot S(\bar{\mathbf{m}}_3)K_{int}R_{cam}R^T(\hat{\mathbf{q}})(\boldsymbol{\xi}_3 - \hat{\mathbf{p}}) \\ CC \cdot S(\bar{\mathbf{m}}_4)K_{int}R_{cam}R^T(\hat{\mathbf{q}})(\boldsymbol{\xi}_4 - \hat{\mathbf{p}}) \end{bmatrix} \quad (4.14)$$

e

$$\boldsymbol{\delta}_z = [\boldsymbol{\delta}_G \quad \boldsymbol{\delta}_A \quad \boldsymbol{\delta}_{c1} \quad \boldsymbol{\delta}_{c2} \quad \boldsymbol{\delta}_{c3} \quad \boldsymbol{\delta}_{c4}]^T \quad (4.15)$$

, onde $CC = \begin{bmatrix} I_{2 \times 2} & 0_{2 \times 1} \end{bmatrix}$.

A matriz de covariância Q_z associada a todas as incertezas introduzidas pelas medições é composta pela covariância de cada sensor utilizado, desta forma a covariância é definida pela seguinte relação:

$$Q_z = \text{diag}\{Q_G, Q_A, Q_{c1}, \dots, Q_{c4}\}, \in \mathbb{R}^{14 \times 14} \quad (4.16)$$

A fim de calcular o ganho de Kalman, assim como anteriormente, precisamos linearizar o modelo de saída dos sensores aplicando as derivadas parciais, ou seja, a Jacobiana $H \in \mathbb{R}^{14 \times 25}$ dada por:

$$H(\hat{\mathbf{x}}) := \frac{\partial \mathbf{h}(\hat{\mathbf{x}})}{\partial \hat{\mathbf{x}}} = [H^{\hat{\mathbf{q}}} \quad H^{\hat{\boldsymbol{\omega}}, \hat{\boldsymbol{\alpha}}} \quad H^{\hat{\mathbf{p}}} \quad H^{\hat{\mathbf{v}}, \hat{\mathbf{a}}} \quad H^{\hat{\mathbf{b}}}], \quad (4.17)$$

onde $H^{\hat{\mathbf{q}}}$, $H^{\hat{\boldsymbol{\omega}}, \hat{\boldsymbol{\alpha}}}$, $H^{\hat{\mathbf{p}}}$, $H^{\hat{\mathbf{v}}, \hat{\mathbf{a}}}$ and $H^{\hat{\mathbf{b}}}$ são as Jacobianas relacionadas com a dinâmica de orientação, velocidade e aceleração angular, posição, velocidade e aceleração linear e os *bias*, respectivamente. Os resultados das derivadas parciais resultam em:

$$H^{\hat{\mathbf{q}}} := \begin{bmatrix} 0_{3 \times 4} \\ R_{imu}\mathbb{J}(\hat{\mathbf{q}}, \hat{\mathbf{a}} - \mathbf{g}) \\ CC \cdot S(\bar{\mathbf{m}}_1)K_{int}R_{cam}\mathbb{J}(\hat{\mathbf{q}}, \boldsymbol{\xi}_1 - \hat{\mathbf{p}}) \\ CC \cdot S(\bar{\mathbf{m}}_2)K_{int}R_{cam}\mathbb{J}(\hat{\mathbf{q}}, \boldsymbol{\xi}_2 - \hat{\mathbf{p}}) \\ CC \cdot S(\bar{\mathbf{m}}_3)K_{int}R_{cam}\mathbb{J}(\hat{\mathbf{q}}, \boldsymbol{\xi}_3 - \hat{\mathbf{p}}) \\ CC \cdot S(\bar{\mathbf{m}}_4)K_{int}R_{cam}\mathbb{J}(\hat{\mathbf{q}}, \boldsymbol{\xi}_4 - \hat{\mathbf{p}}) \end{bmatrix}, \quad (4.18)$$

$$H^{\hat{\boldsymbol{\omega}}, \hat{\boldsymbol{\alpha}}} := \begin{bmatrix} R_{imu}R(\hat{\mathbf{q}}) & 0_{3 \times 4} \\ -R_{imu}R(\hat{\mathbf{q}})\mathbb{W}(\hat{\boldsymbol{\omega}}, \mathbf{O}_{imu}) & -R_{imu}R(\hat{\mathbf{q}})S(\mathbf{O}_{imu}) \\ 0_{8 \times 3} & 0_{8 \times 3} \end{bmatrix}, \quad (4.19)$$

$$H_k^{\hat{p}} := \begin{bmatrix} 0_{6 \times 3} \\ -CC \cdot S(\bar{\mathbf{m}}_1) K_{int} R_{cam} R(\hat{\mathbf{q}}) \\ -CC \cdot S(\bar{\mathbf{m}}_2) K_{int} R_{cam} R(\hat{\mathbf{q}}) \\ -CC \cdot S(\bar{\mathbf{m}}_3) K_{int} R_{cam} R(\hat{\mathbf{q}}) \\ -CC \cdot S(\bar{\mathbf{m}}_4) K_{int} R_{cam} R(\hat{\mathbf{q}}) \end{bmatrix}, \quad (4.20)$$

$$H_k^{\hat{p}, \hat{a}} := \begin{bmatrix} 0_3 & 0_3 \\ 0_3 & R_{imu} R(\hat{\mathbf{q}}) \\ 0_{8 \times 3} & 0_{8 \times 3} \end{bmatrix}, \quad (4.21)$$

e $H^{\hat{b}} = [I_6 \ 0_{6 \times 14}]^T$. As funções auxiliares $\mathbb{J}(\mathbf{q}, \mathbf{u})$ e $\mathbb{W}(\mathbf{y}, \mathbf{u}) \ \forall \mathbf{y}, \mathbf{u} \in \mathbb{R}^3$ são definidas como:

$$\mathbb{J}(\mathbf{q}, \mathbf{u}) := \frac{\partial R(\mathbf{q}) \mathbf{u}}{\partial \mathbf{q}} = \frac{\partial (I_3 - 2\eta S(\boldsymbol{\varepsilon}) + 2S^2(\boldsymbol{\varepsilon})) \mathbf{u}}{\partial \begin{bmatrix} \eta \\ \boldsymbol{\varepsilon} \end{bmatrix}} = 2 \left[-S(\boldsymbol{\varepsilon}) \mathbf{u} S(\mathbf{u}) \eta + \mathbb{W}(\boldsymbol{\varepsilon}, \mathbf{u}) \right] \quad (4.22)$$

e

$$\mathbb{W}(\mathbf{y}, \mathbf{u}) := \frac{\partial S^2(\mathbf{y}) \mathbf{u}}{\partial \mathbf{y}} = \mathbf{y} \mathbf{u}^T - 2\mathbf{u} \mathbf{y}^T + \mathbf{y}^T \mathbf{u} I_3 \quad (4.23)$$

Com todas as relações apresentadas podemos utilizar o algoritmo do EKF apresentado no Capítulo 2, e combinando com as relações apresentadas a cima, temos o algoritmo mostrado na Tabela 5.

Tabela 5 – Algoritmo do EKF para Fusão de Sensores.

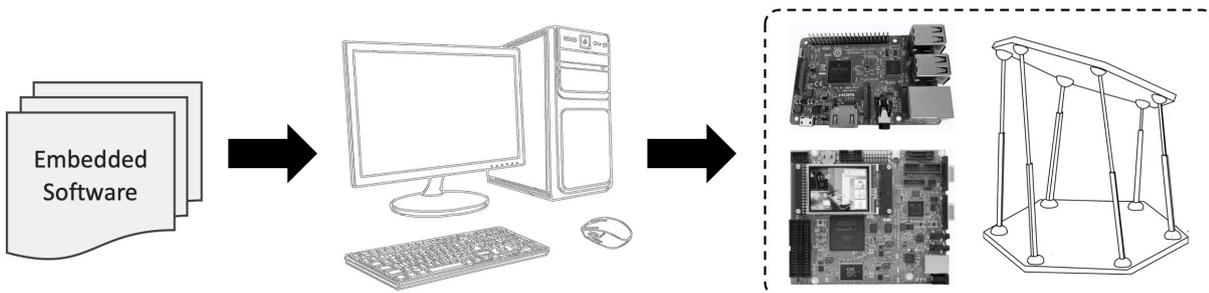
Etapa de Predição:
Calcula F_k de \mathbf{x}_k como em (4.12)
1: $\bar{\mathbf{x}}_k = f(\mathbf{x}_k)$;
2: $\bar{P}_k = F_k P_{k-1} F_k^T + Q_x$
Normalizar o quaternion em \mathbf{x}_k
Etapa de Correção:
Calcula H_k de \mathbf{z}_k como em (4.17)
3: $K_k = \bar{P}_k H_k^T (H_k \bar{P}_k H_k^T + Q_z)^{-1}$
4: $\hat{\mathbf{x}}_{k+1} = \bar{\mathbf{x}}_k + K_k (\mathbf{z}_k - h(\bar{\mathbf{x}}_k))$
5: $\hat{P}_{k+1} = (I - K_k H_k) \bar{P}_k$
Normalizar o quaternion em \mathbf{x}_{k+1}

4.2 Proposta do Software Embarcado

Devido à crescente diversidade de ambientes em que os sistemas 6-DOF são empregados, precisamos necessariamente desenvolver novas abordagens econômicas que melhorem o desempenho e também reduzam os custos para a indústria. Nesse sentido, além de demonstrar matematicamente como podemos melhorar o algoritmo EKF, apresentado na Seção 4.1, este trabalho propôs sua implementação através de um software embarcado em um FPGA que utiliza processadores comerciais, facilitando a reprodução e o uso de nossa abordagem nos mais diversos sistemas 6-DOF encontrados pelo mundo.

A Figura 12 mostra o fluxo de desenvolvimento dividido em três fases. A primeira fase foi dedicada ao desenvolvimento do software embarcado. Inicialmente, um aplicativo MATLAB foi desenvolvido para modelar a estimativa de pose teórica, apresentada na Seção 4.1. Esta é a referência para validar o software embarcado e garantir que as relações matemáticas estão corretas.

Figura 12 – Fluxograma do desenvolvimento do *Software* embarcado em um FPGA.



Fonte: Autor (2020).

Em seguida, o software embarcado foi desenvolvido na linguagem C para calcular a estimativa de pose. Para manipular os dados e implementar o algoritmo mostrado na Tabela 5, foram desenvolvidas bibliotecas para lidar com operações com quaternions e matrizes, por exemplo, funções para calcular a matriz inversa, multiplicações, divisões, matriz jacobiana, norma euclidiana do quaternion, derivada de vetores, produto cruzado, entre outras funções. A Tabela 6 mostra o número de funções e linhas implementadas para cada biblioteca do aplicativo.

Tabela 6 – Bibliotecas Desenvolvidas para Implementação do Algoritmo de Estimação de Pose.

Biblioteca	Número de Funções	Número de linhas de código (LOC)
EKF	2	496
Quaternion	7	141
Protocolo de Comunicação	12	420
Matriz	24	406

Na segunda fase, avaliamos nosso aplicativo em um PC *desktop* padrão baseado no processador x86. Esta etapa nos permitiu garantir a implementação correta do nosso

estimador de pose e também visa facilitar a disseminação de nossa aplicação a ser empregada a outros sistemas de 6-DOF. Nesta etapa, todos os dados de entrada dos sensores – *encoders*, IMU e câmera – foram adquiridos e pré-processados *offline*, possibilitando ajustar as covariâncias e os ganhos do EKF. Para validar nossa aplicação, os estados estimados foram comparados com os dados reais, calculados a partir dos *encoders* usando cinemática direta. Após a validação, aumentamos o suporte a um novo processador, no caso, da família ARM Cortex-M, a fim de abranger a grande maioria dos dispositivos encontrados atualmente e prepará-lo para ser embarcados em um FPGA.

A terceira fase trata da integração do FPGA com a Plataforma Stewart, a fim de que possamos dar suporte a aplicativos *online*, isto é, aplicativos executados em tempo real. O pré-processamento de imagem é feito por uma placa Raspberry PI 3 Modelo B, onde os dados extraídos de cada marcador são agrupados com os dados da IMU e enviados ao FPGA por meio de um protocolo desenvolvido. O objetivo em dividir o problema em tarefas menores é tornar a implementação mais barata ou até mesmo aproveitar, no futuro, do suporte a multitarefa de um processador multiprocessado, isto é, que tenha mais de um processador.

A Tabela 7 mostra o formato do pacote de dados desenvolvido para a comunicação entre os módulos (FPGA, Raspberry PI e sensores), que inclui um cabeçalho com 4 campos (ou seja, início, endereço, comando e número de dados); uma carga útil de dados com tamanho variado; uma verificação de redundância cíclica (CRC) para validar o pacote inteiro; e um byte de parada. Observe que cada módulo é identificado com um endereço exclusivo, permitindo o controle individual de cada atuador. Além disso, os módulos são conectados através do barramento de comunicação RS-485, tornando-o um sistema de comunicação serial.

Tabela 7 – Protocolo de Comunicação.

INÍCIO	ENDEREÇO	COMANDO	N DADOS	DADOS	CRC16	PARADA
1 Byte	1 Byte	1 Byte	1 Byte	N Bytes	2 Byte	1 Byte

Após cada ensaio, que consiste em várias iterações, um *DATALOG* é criado com a estimativa de pose calculada pelo FPGA com todos os dados do sensor (ou seja, *encoders*, IMU e câmera). Em seguida, esse filtro é validado comparando os dados estimados com os dados reais calculados a partir da cinemática direta, mostrando a eficácia de nossa abordagem.

5 Resultados Experimentais

Neste capítulo será apresentado os resultados obtidos através do fluxo desenvolvido para validação do filtro proposto neste trabalho. Primeiramente, é apresentado a simulação dos sensores comparando com as medições obtidas, em seguida, são mostrados os valores de posição e orientação da Plataforma Stewart inferidos através da cinemática direta. Por fim, é apresentado a comparação dos resultados obtidos com o Filtro de Kalman Estendido assim como uma análise estatística dos dados. Para mostrar a eficiência do filtro é feito um ensaio onde a perda dos marcadores é forçada.

Vale ressaltar que o conceito está sendo validado por meio de um protótipo FPGA, que foi implementado com um *soft-core* ARM Cortex-M3. Nesse sentido, os experimentos foram realizados com um tempo de amostragem de baixa frequência ($0,833Hz$) e, portanto, a dinâmica de plataforma foi cerca de 100 vezes mais lenta.

5.1 Simulação dos Sensores

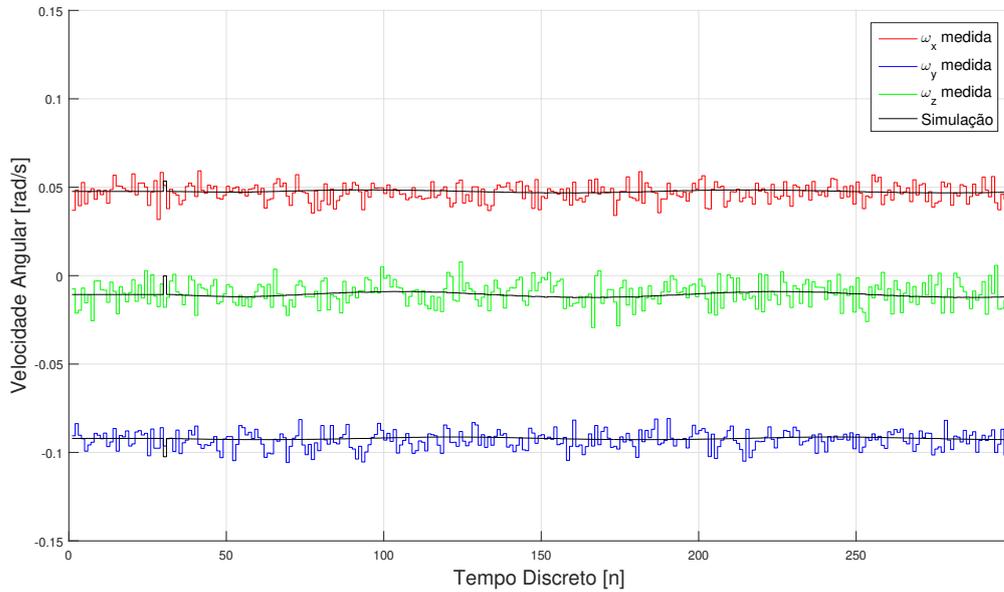
Para a simulação dos sensores foram consideradas as relações apresentadas no Capítulo 2, mais precisamente nas as equações (2.14), (2.15) e (2.33). Todos os processos descritos anteriormente foram considerados e o pós-processamento dos dados foi realizado no *software* MATLAB.

A Figura 13 apresenta a comparação entre a simulação do sensor giroscópio inferida através da equação (2.14) em relação aos dados medidos. Aqui nota-se uma característica deste sensor, o ruído em baixas frequências. Por outro lado, a Figura 14 apresenta a comparação entre a simulação do sensor acelerômetro em relação aos dados medidos. Ao contrário do giroscópio, o acelerômetro apresenta ruídos em altas frequências, por isto temos uma curva do sinal medido bastante suave, como mostrado na Figura 14.

A Figura 15 mostra a simulação da câmera em relação aos dados medidos, podendo ser observado a baixa covariância em suas medidas, sendo esta uma das principais características deste sensor. A análise dos resultados mostra que os modelos representam uma descrição fiel dos sensores, provando que o ambiente de simulação condiz matematicamente com o sistema real.

Os dados utilizados para realizar a simulação dos sensores foram obtidos através dos dados adquiridos do primeiro ensaio, ou seja, o ensaio representado pela Seção 5.2. A Figura 16 mostra os valores obtidos pelo *encoder*, representado pelo gráfico de tamanho dos atuadores, e a posição e orientação inferida pela cinemática direta. Os dados aqui computados serão utilizados para comparação dos dados estimados pelo filtro no

Figura 13 – Simulação do sensor giroscópio em relação aos dados medidos.



Fonte: Autor (2020).

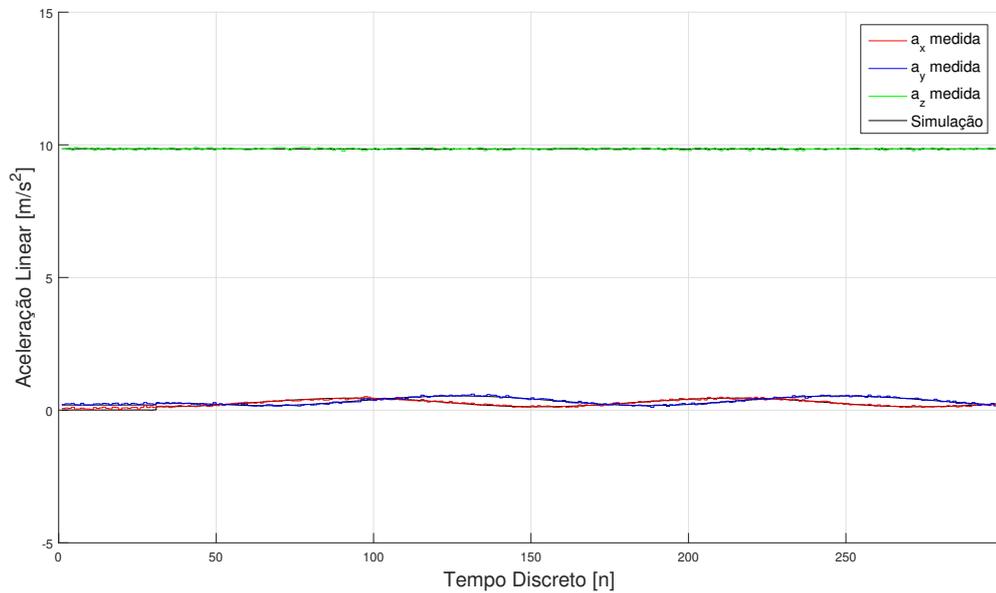
experimento 1.

5.2 Aplicação do Filtro de Kalman Estendido

Dois experimentos foram realizados para avaliar o algoritmo EKF embarcados no FPGA, ambos com trajetórias sinusoidais: o primeiro tendo os marcadores presentes durante todo o ensaio, e o segundo experimento forçando a perda dos marcadores para demonstrar a eficiência do filtro proposto. Como dito na seção anterior, os resultados são validados com os estados reconstruídos através da cinemática direta baseada nos valores dos *encoders*. A relação Euler–Quaternion representado pela equação (2.7), foi utilizada para computar os Ângulos de Euler (*roll*, *pitch* e *yaw*), uma vez que a visualização da orientação de um corpo rígido por Ângulos de Euler é mais intuitiva em relação a orientação baseada em quaternion.

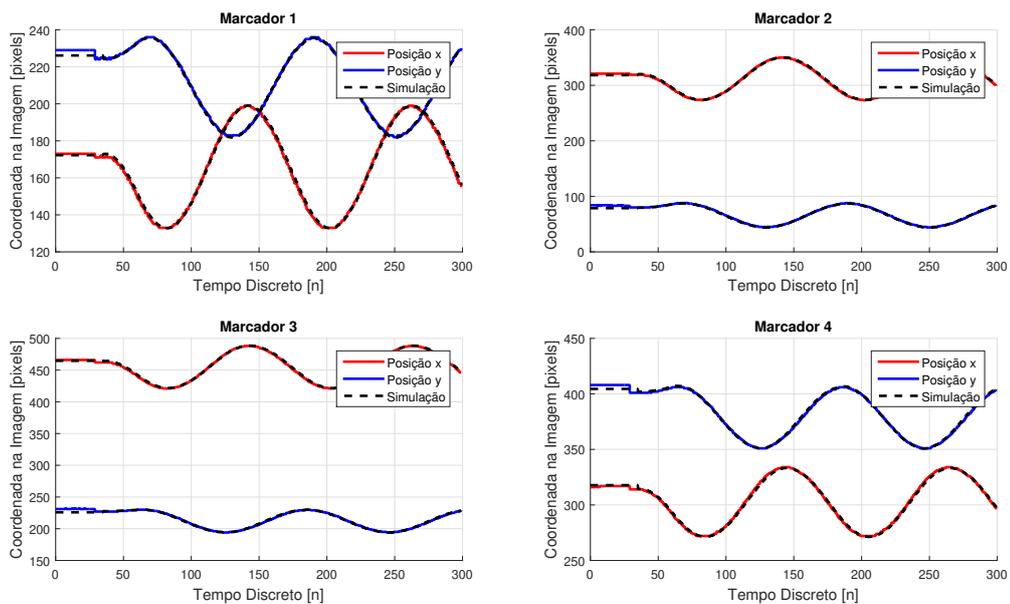
A covariância inicial em ambos os experimentos para a matriz P é uma matriz nula, ou seja, $P = [0_{25 \times 25}]$. Já a matriz de covariância dos sensores de Q_z e seus respectivos *bias* são calculados a cada inicialização do aplicativo, ou seja, por um período de tempo, um pacote de dados do sensor é adquirido com a plataforma em repouso e, em seguida, o *bias* e covariância de cada sensor são calculados, conforme mostrado na Tabela 8. Por outro lado, a covariância do modelo foi previamente definida empiricamente e consiste em uma matriz dada por $Q_x = \begin{bmatrix} 10^{-5} \times I_{19 \times 19} & 0_{19 \times 6} \\ 0_{6 \times 19} & 0_{6 \times 6} \end{bmatrix}$.

Figura 14 – Simulação do sensor acelerômetro em relação aos dados medidos.



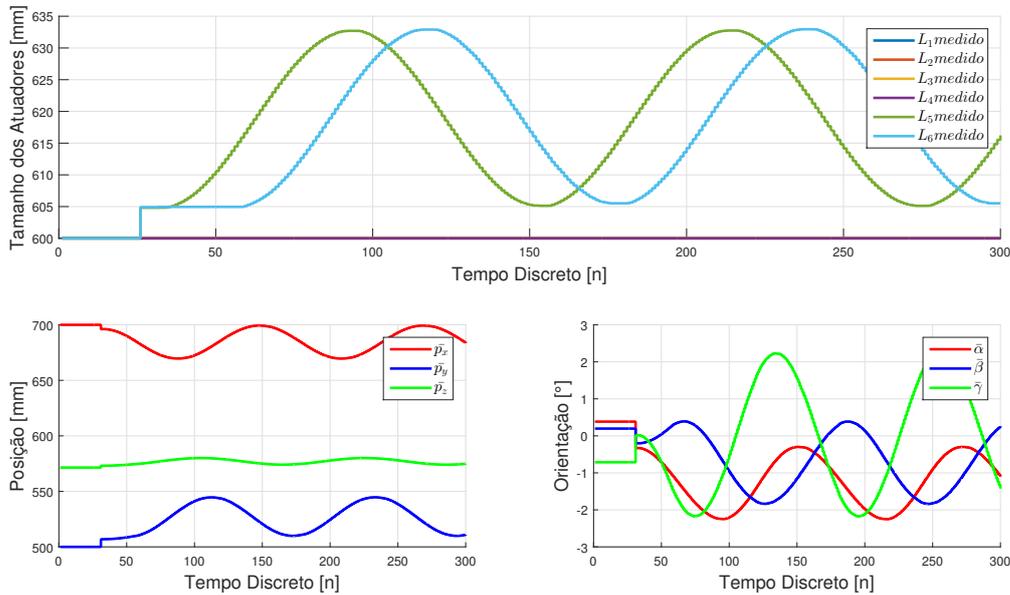
Fonte: Autor (2020).

Figura 15 – Simulação da câmera em relação aos dados medidos.



Fonte: Autor (2020).

Figura 16 – Orientação e posição inferida pela cinemática direta.



Fonte: Autor (2020).

Tabela 8 – Covariância Associada aos Sensores.

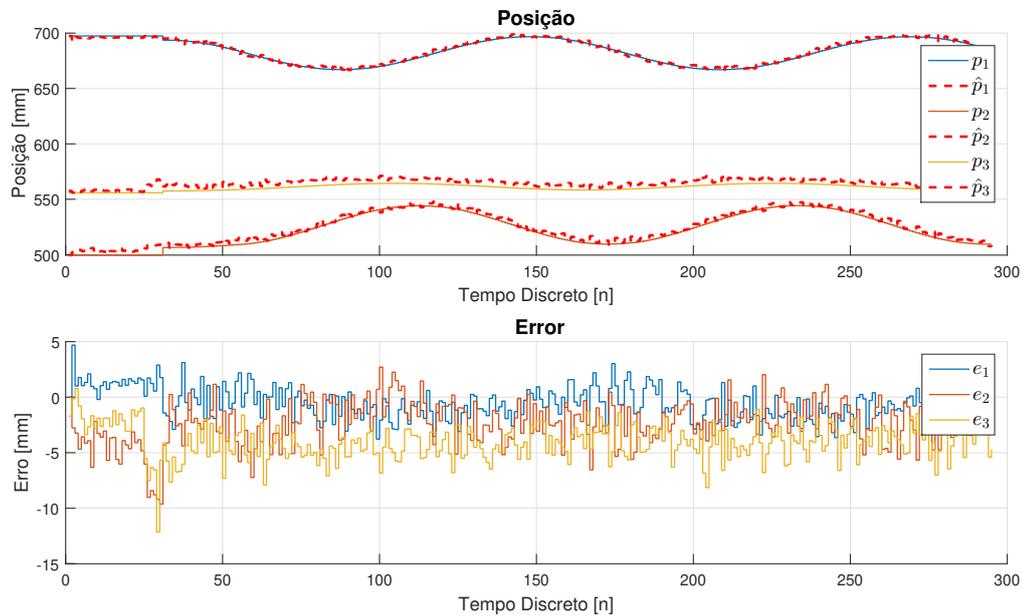
Sensor	Experimento 1	Experimento 2
Accelerometro	$\text{diag}\{0.02793, 0.01708, 0.0465\}$	$\text{diag}\{0.01158, 0.00806, 0.01276\}$
Giroscópio	$10^{-3} \times \text{diag}\{0.7401, 0.697, 1.136\}$	$10^{-3} \times \text{diag}\{0.9007, 1.2848, 0.3798\}$
Câmera	$10^{-3} \times I_8$	$10^{-3} \times I_8$

As Figuras 17–18 ilustram os resultados adquiridos com a aplicação funcionando em tempo real. Os dados estimados estão em pontilhado e os dados reais são as linhas contínuas. É possível observar que as trajetórias estimadas pelo filtro se comportam conforme o esperado em relação aos estados reais, o que é comprovado pelo baixo erro mostrado na parte inferior das Figuras 17–18.

Além disso, a Tabela 9 apresenta uma análise estatística do primeiro experimento, contendo a média, desvio padrão (σ) e o maior valor absoluto dos erros de estimativa para cada eixo de orientação e eixo de posição dos dados adquiridos no FPGA. Isso mostra que o filtro desenvolvido conseguiu estimar a posição com um erro abaixo de 8,14 mm (17) e orientação com um erro de 0,63° (18), o que comprova a qualidade do filtro desenvolvido.

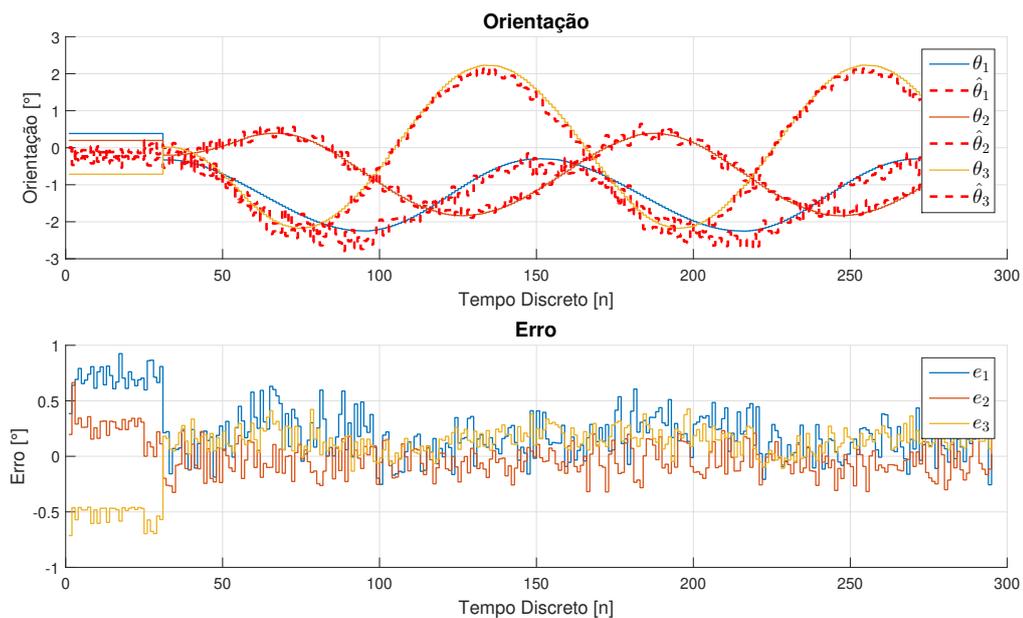
O segundo experimento proposto visa avaliar o desempenho do filtro e seu poder de recuperação, neste sentido, forçamos a perda de dados proveniente da câmera. Para lidar com a perda de marcadores, o algoritmo do filtro implementado reutiliza os dados da câmera do instante anterior. As Figuras 19–20 mostram a performance do filtro nessas condições. Essa escolha de *design* foi feita para aumentar a confiabilidade do filtro desenvolvido. Nesse sentido, o segundo experimento força a perda de marcadores em dois

Figura 17 – Experimento 1: estimativa da posição inferida pelo Filtro de Kalman Estendido e o erro em relação ao valor real.



Fonte: Autor (2020).

Figura 18 – Experimento 1: estimativa da orientação inferida pelo Filtro de Kalman Estendido e o erro em relação ao valor real.



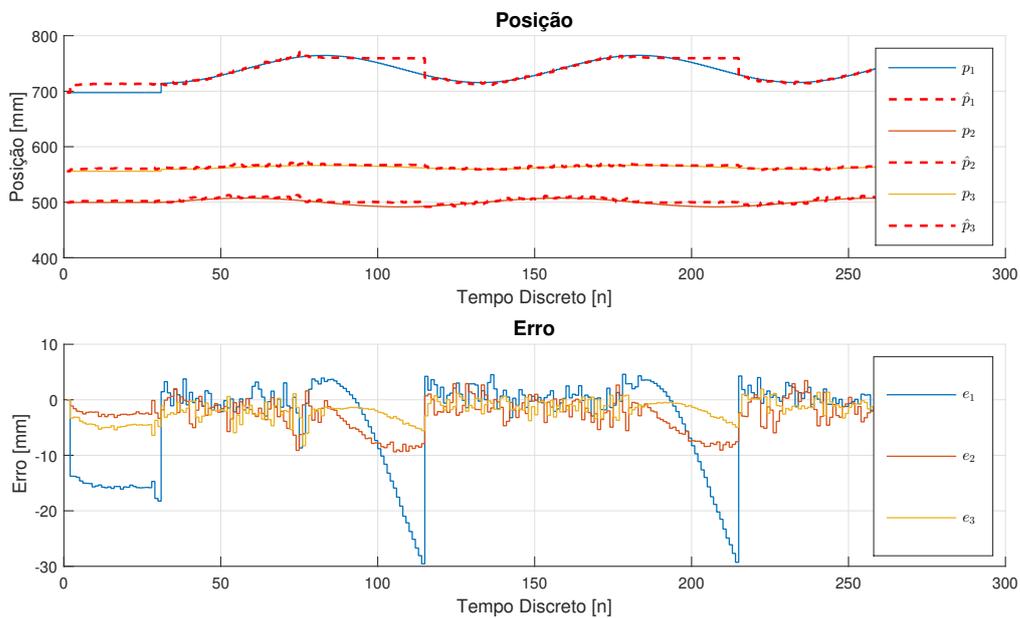
Fonte: Autor (2020).

Tabela 9 – Análise Estatística do Experimento Realizado.

Descrição	Erro de posição [mm]	Erro de orientação [°]
	Experimento 1	Experimento 2
média x	-0.7119	0.1630
média y	-2.0394	-0.0452
média z	-4.1676	0.1456
σ_x	1.3512	0.1792
σ_y	1.7648	0.1289
σ_z	1.3701	0.1054
$\max(x)$	3.7803	0.6304
$\max(y)$	7.2305	0.3254
$\max(z)$	8.1471	0.4268

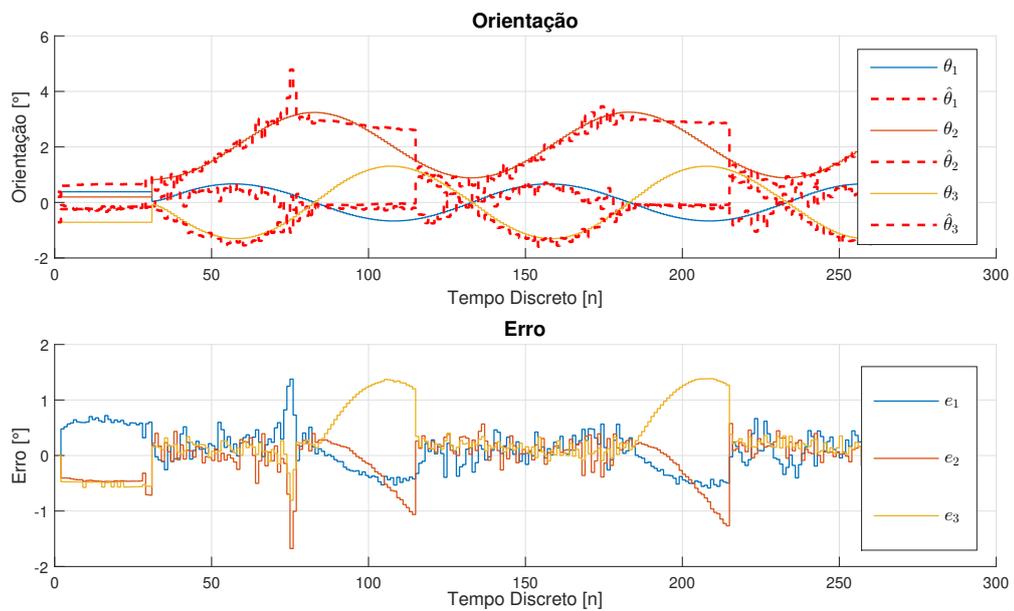
momentos distintos. Observamos que quando os marcadores são perdidos (amostras 80 e 180), o filtro se desvia dos dados reais adquiridos da plataforma. No entanto, quando os marcadores retornam ao campo de visão da câmera, o filtro converge para os valores de estimativa de pose corretos, demonstrando que (1) o algoritmo depende de todos os dados e (2) o algoritmo é capaz de recuperar e convergir para o valor ideal de estimativa de pose.

Figura 19 – Experimento 2: estimativa da posição inferida pelo Filtro de Kalman Estendido e o erro em relação ao valor real.



Fonte: Autor (2020).

Figura 20 – Experimento 2: estimativa da orientação inferida pelo Filtro de Kalman Estendido e o erro em relação ao valor real.



Fonte: Autor (2020).

6 Conclusões

O presente trabalho propôs uma nova formulação de um estimador de posição e orientação para um sistema de seis graus de liberdade visando aplicações *online*. Para isso, na fusão dos sensores, utilizou-se o Filtro de Kalman Estendido baseado em quaternions. A topologia escolhida para sensoriamento foi a *sensor-visão*, que contém: uma IMU com um giroscópio e um acelerômetro; e uma câmera que usa marcadores com cores diferentes e geometria simples para reduzir o esforço computacional. Assim, este estudo considerou o uso de um microcontrolador comercial de baixo custo e alta disponibilidade no mercado, facilmente encontrado em sistemas embarcados.

A implementação e avaliação do fluxo de desenvolvimento proposto se deu por meio da utilização de uma Plataforma Stewart. Havendo, portanto, a necessidade do desenvolvimento de bibliotecas na linguagem C para lidar com as operações matemáticas expostas no Capítulo 4. Sobre este ponto, a Tabela 6 apresentou quais funções foram desenvolvidas, assim como a quantidade de funções e números de linhas. Ademais, considerando o *software* desenvolvido, observou-se necessário também o desenvolvimento de um conversor RS-232/RS-485, dado que o FPGA tinha como porta serial o tipo RS-232 e como meio de comunicação disponível para integração o tipo RS-485.

Por fim, no que tange aos resultados apresentados no Capítulo 5, ressalta-se que o filtro desenvolvido mostrou um nível de excelência, apresentando um baixo erro de posição e orientação, de 8,14 mm e 0,63°, respectivamente. Com o intuito de demonstrar a eficiência do filtro proposto, foi realizado um experimento no qual forçou-se a perda de dados dos marcadores em dois momentos distintos: Figura 19–20. Os resultados mostram que há divergência do sinal no momento em que há a perda de dados dos marcadores. Contudo, no instante em que a disponibilidade dos dados é reestabelecida, o filtro converge rapidamente para os valores esperados.

6.1 Perspectivas de Trabalhos Futuros

De modo a destacar as perspectivas para trabalhos futuros, elencam-se os seguintes itens:

- Tratar os pontos críticos do algoritmo através da abordagem de arquitetura paralela;
- Aplicar o controlador proposto por Faria et al. (2016), considerando os estados estimados pelo filtro;

- Implementar outras metodologias para fusão de sensores e estimação de orientação e posição, como, por exemplo, o Filtro de Partículas.

Referências

- ABOUDONIA, A.; EL-BADAWY, A.; RASHAD, R. Active anti-disturbance control of a quadrotor unmanned aerial vehicle using the command-filtering backstepping approach. *Nonlinear Dynamics*, v. 90, n. 1, p. 581–597, July 2017. Citado na página 21.
- ANDERSON, B. D.; MOORE, J. B. *Optimal filtering*. [S.l.]: Courier Corporation, 2012. Citado na página 36.
- CARDONA, M. A new approach for the forward kinematics of general stewart-gough platforms. In: IEEE. *2015 IEEE Thirty Fifth Central American and Panama Convention (CONCAPAN XXXV)*. [S.l.], 2015. p. 1–6. Citado na página 42.
- CHUNYANG, Z. et al. A dual quaternion based fusion framework for imu data with 6 dof pose. *Journal of Electronics and Information Science*, Clausius Scientific Press, v. 1, n. 1, p. 22–31, 2016. Citado na página 21.
- DASGUPTA, B.; MRUTHYUNJAYA, T. The stewart platform manipulator: a review. *Mechanism and machine theory*, Elsevier, v. 35, n. 1, p. 15–40, 2000. Citado na página 27.
- DEBRUIN, J. Control systems for mobile satcom antennas. *IEEE Control Systems Magazine*, v. 28, n. 1, p. 86–101, Feb 2008. Citado na página 22.
- DIEBEL, J. Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix*, v. 58, n. 15-16, p. 1–35, Out 2006. Citado 2 vezes nas páginas 25 e 26.
- EUSTON, M. et al. A complementary filter for attitude estimation of a fixed-wing uav. In: IEEE. *2008 IEEE/RSJ international conference on intelligent robots and systems*. [S.l.], 2008. p. 340–345. Citado na página 21.
- FARIA, P. F. de et al. Quaternion-based dynamic control of a 6-dof stewart platform for periodic disturbance rejection. In: IEEE. *Control Applications (CCA), 2016 IEEE Conference on*. [S.l.], 2016. p. 1191–1196. Citado 2 vezes nas páginas 28 e 63.
- GARCÍA, S. et al. Indoor slam for micro aerial vehicles control using monocular camera and sensor fusion. In: IEEE. *2016 international conference on autonomous robot systems and competitions (ICARSC)*. [S.l.], 2016. p. 205–210. Citado na página 21.
- GUO, H. et al. Implementation of ekf for vehicle velocities estimation on fpga. *IEEE Transactions on Industrial Electronics*, v. 60, n. 9, p. 3823–3835, Sep. 2012. Citado na página 23.
- KALMAN, R. E. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, American Society of Mechanical Engineers, v. 82, n. 1, p. 35–45, 1960. Citado na página 36.
- KUCUK, S. *Serial and Parallel Robot Manipulators: Kinematics, Dynamics, Control and Optimization*. [S.l.]: BoD–Books on Demand, 2012. Citado na página 27.

- KUMAR, P. R.; BEHERA, A. K.; BANDYOPADHYAY, B. Robust finite-time tracking of stewart platform: A super-twisting like observer-based forward kinematics solution. *IEEE Transactions on Industrial Electronics*, IEEE, v. 64, n. 5, p. 3776–3785, 2017. Citado na página 22.
- LEE, C.; SALCIC, Z. A fully-hardware-type maximum-parallel architecture for kalman tracking filter in fpgas. In: *International Conference on Information, Communications and Signal Processing (ICICSP)*. [S.l.: s.n.], 1997. v. 2, p. 1243–1247. Citado na página 23.
- LI, J. et al. A combined vision-inertial fusion approach for 6-DoF object pose estimation. In: VERIKAS, A. et al. (Ed.). *Seventh International Conference on Machine Vision (ICMV 2014)*. SPIE, 2015. v. 9445, p. 238 – 249. Disponível em: <<https://doi.org/10.1117/12.2180574>>. Citado na página 40.
- MA, Y. et al. *An invitation to 3-d vision: from images to geometric models*. [S.l.]: Springer Science & Business Media, 2012. v. 26. Citado 4 vezes nas páginas 31, 32, 33 e 34.
- Mariottini, G. L.; Prattichizzo, D. Egt for multiple view geometry and visual servoing: robotics vision with pinhole and panoramic cameras. *IEEE Robotics Automation Magazine*, v. 12, n. 4, p. 26–39, Dec 2005. ISSN 1070-9932. Citado na página 35.
- MARKLEY, F. L.; CRASSIDIS, J. L. *Fundamentals of spacecraft attitude determination and control*. [S.l.]: Springer, 2014. v. 33. Citado na página 29.
- MARKUS, E. D.; AGEE, J. T.; JIMOH, A. A. Flat control of industrial robotic manipulators. *Robotics and Autonomous Systems*, v. 87, n. C, p. 226–236, Jan. 2017. Citado na página 21.
- MATTILA, J. et al. A survey on control of hydraulic robotic manipulators with projection to future trends. *IEEE/ASME Transactions on Mechatronics*, v. 22, n. 2, p. 669–680, April 2017. Citado na página 21.
- MILETOVIĆ, I. et al. Improved stewart platform state estimation using inertial and actuator position measurements. *Control Engineering Practice*, Elsevier, v. 62, p. 102–115, 2017. Citado na página 22.
- MIRZAEI, F. M.; ROUMELIOTIS, S. I. A kalman filter-based algorithm for imu-camera calibration: Observability analysis and performance evaluation. *IEEE transactions on robotics*, IEEE, v. 24, n. 5, p. 1143–1156, 2008. Citado na página 22.
- NXP. Lpc1800 series: High-performance microcontrollers (mcus) based on arm cortex-m3 cores. Available at <[https://www.nxp.com/products/processors-and-microcontrollers/arm-microcontrollers/general-purpose-mcus/lpc1800-cortex-m3:MC_1403790776032#/
>](https://www.nxp.com/products/processors-and-microcontrollers/arm-microcontrollers/general-purpose-mcus/lpc1800-cortex-m3:MC_1403790776032#/)>, 2015. Citado na página 23.
- OLENICK, R.; APOSTOL, T.; GOODSTEIN, D. *The Mechanical Universe: Introduction to Mechanics and Heat*. [S.l.]: Cambridge University Press, 2008. Citado na página 30.
- OSHMANN, Y.; CARMÍ, A. Attitude estimation from vector observations using a genetic-algorithm-embedded quaternion particle filter. *Journal of Guidance, Control, and Dynamics*, v. 29, n. 4, p. 879–891, 2006. Citado na página 21.

- SHARMA, U.; RAWAL, V.; TAYAL, V. An overview on multiple unmanned aerial vehicle control through single controller. *International Journal of Computer Applications*, v. 180, n. 49, p. 8–12, June 2018. Citado na página 21.
- TEAGUE, M. R. Image analysis via the general theory of moments. *JOSA*, Optical Society of America, v. 70, n. 8, p. 920–930, 1980. Citado na página 42.
- TI. Omap4 mobile applications platform. Available at <<http://www.ti.com/lit/ml/swpt034b/swpt034b.pdf>>, 2011. Citado na página 23.
- TRUCCO, E.; VERRI, A. *Introductory techniques for 3-D computer vision*. [S.l.]: Prentice Hall Englewood Cliffs, 1998. v. 201. Citado na página 31.
- VASQUEZ, X. M. Z. et al. Posicionamento de múltiplos objetos a partir de visão estéreo. Pontifícia Universidade Católica do Rio Grande do Sul, 2015. Citado na página 33.
- WELCH, G.; BISHOP, G. et al. An introduction to the kalman filter. 1995. Citado 2 vezes nas páginas 36 e 37.
- XIANG, X.; YU, C.; ZHANG, Q. Robust fuzzy 3D path following for autonomous underwater vehicle subject to uncertainties. *Computers & Operations Research*, v. 84, n. C, p. 165–177, Aug. 2017. Citado na página 21.
- XU, D.; LI, Y. F. A new pose estimation method based on inertial and visual sensors for autonomous robots. In: IEEE. *2007 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. [S.l.], 2007. p. 405–410. Citado na página 40.
- YANG, C. et al. Real-time state estimation for spatial six-degree-of-freedom linearly actuated parallel robots. *Mechatronics*, Elsevier, v. 19, n. 6, p. 1026–1033, 2009. Citado na página 22.
- Zhang, S. et al. Adaptive neural control for robotic manipulators with output constraints and uncertainties. *IEEE Transactions on Neural Networks and Learning Systems*, v. 29, n. 11, p. 5554–5564, Nov. 2018. Citado na página 21.
- ZHENG, X.; GUO, Y. Research on real time 6dof robot localization based on visual and inertial fusion. In: ATLANTIS PRESS. *2017 5th International Conference on Machinery, Materials and Computing Technology (ICMMCT 2017)*. [S.l.], 2017. Citado na página 21.



Pontifícia Universidade Católica do Rio Grande do Sul
Pró-Reitoria de Graduação
Av. Ipiranga, 6681 - Prédio 1 - 3º. andar
Porto Alegre - RS - Brasil
Fone: (51) 3320-3500 - Fax: (51) 3339-1564
E-mail: prograd@pucrs.br
Site: www.pucrs.br