



**LOCALIZAÇÃO VISUAL INERCIAL COM AUTO-CALIBRAÇÃO  
DOS PARÂMETROS INTERNOS DE CÂMERA MONOCULAR**

**ARTUR BORGES PIO**

**DISSERTAÇÃO DE MESTRADO EM ENGENHARIA DE SISTEMAS  
ELETRÔNICOS E AUTOMAÇÃO  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**FACULDADE DE TECNOLOGIA  
UNIVERSIDADE DE BRASÍLIA**



UNIVERSIDADE DE BRASÍLIA  
FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

**INERTIAL VISUAL LOCATION WITH SELF-CALIBRATION OF  
INTERNAL PARAMETERS OF MONOCULAR CAMERA**

**LOCALIZAÇÃO VISUAL INERCIAL COM AUTO-CALIBRAÇÃO  
DOS PARÂMETROS INTERNOS DE CÂMERA MONOCULAR**

**ARTUR BORGES PIO**

**ORIENTADOR: GEOVANY ARAÚJO BORGES**

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA DE  
SISTEMAS ELETRÔNICOS E AUTOMAÇÃO

PUBLICAÇÃO: PPGEA.DM-725/2019

BRASÍLIA/DF: AGOSTO - 2019

**UNIVERSIDADE DE BRASÍLIA  
FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**LOCALIZAÇÃO VISUAL INERCIAL COM AUTO-CALIBRAÇÃO DOS  
PARÂMETROS INTERNOS DE CÂMERA MONOCULAR**

**ARTUR BORGES PIO**

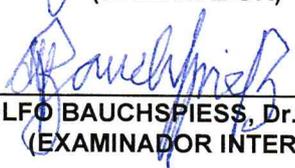
DISSERTAÇÃO DE MESTRADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE.

APROVADA POR:



---

**GEOVANY ARAÚJO BORGES, Dr., ENE/UNB  
(ORIENTADOR)**



---

**ADOLFO BAUCHSPIESS, Dr., ENE/UNB  
(EXAMINADOR INTERNO)**



---

**JOÃO YOSHIYUKI ISHIHARA, Dr., ENE/UNB  
(SUPLENTE)**

Brasília, 12 de agosto de 2019.

## FICHA CATALOGRÁFICA

PIO, ARTUR;

Localização Visual Inercial com Auto-Calibração dos Parâmetros Internos de Câmera Monocular [Distrito Federal] 2019.

ix+109 p., 210 x 297 mm (ENE/FT/UnB, Engenheiro, Engenharia de Sistemas Eletrônicos e Automação, 2019).

DISSERTAÇÃO DE MESTRADO – Universidade de Brasília, Faculdade de Tecnologia.

Departamento de Engenharia Elétrica

1. Localização

2. Calibração de Câmera

3. Auto-Calibração

4. Localização Visual Inercial

I. ENE/FT/UnB

II. Título (série)

## REFERÊNCIA BIBLIOGRÁFICA

PIO, A. (2019). Localização Visual Inercial com Auto-Calibração dos Parâmetros Internos de Câmera Monocular, DISSERTAÇÃO DE MESTRADO, Publicação PPGEA.DM-725/2019, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, ix+109.

## CESSÃO DE DIREITOS

AUTOR: Artur Borges Pio

TÍTULO: Localização Visual Inercial com Auto-Calibração dos Parâmetros Internos de Câmera Monocular.

GRAU: Engenheiro ANO: 2019

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa dissertação de mestrado pode ser reproduzida sem autorização por escrito do autor.

---

Artur Borges Pio

Departamento de Eng. Elétrica (ENE) - FT

Universidade de Brasília (UnB)

Campus Darcy Ribeiro

CEP 70919-970 - Brasília - DF - Brasil

*À toda a minha família.*

## AGRADECIMENTOS

*Primeiramente gostaria de agradecer ao meu orientador, professor Geovany Araújo Borges, que foi quem me ajudou a colocar os pensamentos e ideias em ordem. Também quero agradecer ao meu primeiro mentor que tive na minha vida acadêmica, George Andrew Brindeiro, que foi quem me deu as primeiras lições de como se trabalhar em um ambiente de pesquisa.*

*Também gostaria de deixar o meu muito obrigado para todos aqueles que me acompanharam ao longo de todos os anos de mestrado: meus amigos. Camila, Letícia, Helena, Rodrigo, Johnny, Renato, Buzz, Crepe, Giordano, Daniel, Humberto, Kyth, Ricardo e Thiago, vocês me acompanham desde os anos da graduação e não faltaram encontros, conversas, almoços e até mesmo discussões que serviram para criarmos laços que tenho certeza que vão durar muitos anos. Nos anos de mestrado também fiz novas amizades no ambiente de pesquisa. Cris, Eric, Roberto, Bruno e todos os demais amigos que fiz ao longo da realização das matérias mais difíceis, noites viradas no laboratório e até mesmo em congresso, obrigado por tornarem essa jornada mais divertida.*

*Ao longo desses anos também pude contar com o apoio da minha "prima" Alana, que com seu constante bom humor e alto astral sempre conseguiu me animar até mesmo em semanas de provas ou entregas de trabalhos. Também agradeço a todos os encontros que tive com minha amiga Bia, que sempre me proporcionou conversas e reflexões importantes.*

*Claro, não posso deixar de agradecer à minha namorada Natália. Durante todos esses anos sempre pude contar com seu apoio e amor. Seja para celebrar momentos de euforia por bons resultados obtidos, para confessar nervosismos com apresentações ou mesmo para compartilhar frustrações, você sempre me deu atenção e ouvidos. Obrigado por estar ao meu lado em todas essas etapas. Você foi fundamental para que mais esse desafio fosse superado.*

*E não podia deixar de agradecer ao suporte, apoio e carinho infinito que recebo da minha família. Agradeço aos meus avós, que sempre torceram muito por mim e que sempre foram excelentes fontes de sabedoria. Aos meus tios, que sempre foram minhas referências e exemplos de esforço, aventura e diversão. Ao meu primo de muitos anos, que sempre esteve na posição de meu grande melhor amigo. Ao meu novo primo, que mesmo com sua pequena estatura já preenche boa parte dos corações de todos. Ao meu irmão mais velho, que sempre foi uma referência direta me despertando, desde cedo, a vontade de questionar e ter curiosidade pela ciência. Ao meu irmão mais novo, que compartilha, desde a infância, momentos de diversão, crescimento e aprendizado comigo. Ao meu pai, que sempre foi um exemplo a ser seguido, que desde o meu nascimento me proporciona absolutamente tudo que preciso e quem me ensina constantemente a viver. E claro, à minha querida mãe, que sem sombra de dúvidas é quem maior impacto teve na construção de quem sou hoje e a quem eu mais devo obrigados na vida.*

## RESUMO

**Título:** Localização Visual Inercial com Auto-Calibração dos Parâmetros Internos de Câmera Monocular

**Autor:** Artur Borges Pio

**Orientador:** Geovany Araújo Borges

Na robótica móvel, uma das habilidades mais importantes dos robôs é saber se localizar no espaço. Seja no caso de um carro autônomo que precisa dirigir por uma cidade, um VANT que auxilia uma equipe de resgate ou mesmo um simples robô de laboratório é crucial que todos eles consigam saber onde estão, em outras palavras, precisam de um sistema de Localização para realizarem suas tarefas. Existem diversas técnicas e algoritmos de se resolver o problema da localização de robôs. Um destes sistemas que vem ganhando certo destaque nas pesquisas dos últimos tempos é a Localização Visual Inercial. Este tipo de técnica consiste em realizar a Localização de um agente robótico a partir da combinação das leituras obtidas por uma Unidade de Medição Inercial com um sistema de Odometria Visual. Esta combinação dos sensores pode ser realizada utilizando técnicas de filtragem estocástica, como uma variação do Filtro de Kalman, para desempenhar a chamada fusão sensorial. Para que a fusão seja bem sucedida e entregue boas estimativas, é de extrema importância que os sensores façam leituras não enviesadas e que a câmera esteja bem calibrada. A calibração de câmera é o procedimento executado para se calcular os parâmetros internos da câmera que são utilizados para que se possa construir um modelo matemático da mesma. As técnicas de calibração mais difundidas são executadas de maneira offline e exigem certo conhecimento do ambiente em que a câmera se encontra. Como existem situações que podem vir a alterar os valores dos parâmetros internos da câmera, a calibração precisa ser constantemente realizada para que sempre se possua os valores corretos dos parâmetros internos. Algoritmos de visão computacional como a Odometria Visual, que precisam dos parâmetros internos durante a sua execução, podem vir a ser prejudicados caso ocorra alguma alteração nos valores destes parâmetros durante a sua execução. Sendo assim, o presente trabalho apresenta uma possível solução para este tipo de situação. Ao longo da pesquisa aqui reportada, foi desenvolvido um sistema de Auto-Calibração de Câmera com auxílio de uma UMI. Este sistema consiste em um estimador de parâmetros internos da câmera baseado em um FKE que opera de maneira *online* e sem qualquer conhecimento prévio a respeito do ambiente em que a câmera se encontra. Este método permite a sistemas como a Odometria Visual operarem sempre com os valores corretos dos parâmetros internos da câmera. A partir do auto-calibrador desenvolvido, foi ainda projetado um sistema de Localização Visual Inercial com Auto-Calibração de Câmera (LOVIACC). Este sistema LOVIACC foi projetado como o conjunto de dois FKUs que operam com o intuito de estimar a posição e orientação de um agente robótico, que carrega uma câmera e uma UMI, ao mesmo tempo em que faz as calibração da câmera utilizada. A dissertação apresenta ainda alguns experimentos realizados a fim de se validar os dois métodos propostos.



# ABSTRACT

**Title:** Inertial Visual Location with Self-Calibration of Internal Parameters of Monocular Camera

**Author:** Artur Borges Pio

**Supervisor:** Geovany Araújo Borges

In mobile robotics, one of the most important skills of robots is knowing how to locate itself in space. Whether it's a self-driving car driving in a city, a UAV that assists a rescue team or even a simple lab robot, it's crucial that they all know where they are, in other words, they need a Localization system in order to solve their tasks. There are several techniques and algorithms to solve the robot localization problem. One of these systems that has been gaining some prominence in the surveys of recent times is Visual Inertial Localization. This type of technique consists in performing the Localization of a robotic agent from the combination of the readings obtained by an Inertial Measurement Unit with a Visual Odometry system. This combination of sensors can be performed using stochastic filtering techniques, such as a variation of the Kalman Filter, to perform the so called sensor fusion. For the fusion to be successful and provide good estimates, it is extremely important that the sensors make unbiased readings and that the camera is well calibrated. Camera calibration is the procedure performed to calculate the internal parameters of the camera that are used so that a mathematical model can be constructed. The most widespread calibration techniques are performed offline and require some knowledge of the environment in which the camera is located. As there are situations that may change the values of the internal parameters of the camera, the calibration needs to be constantly performed so that the correct values of the internal parameters are always known. Computer vision algorithms such as Visual Odometry, which need the internal parameters during its execution, can be impaired if there is any change in the values of these parameters during its execution. Therefore, the present work presents a possible solution to this type of situation. Throughout the research reported here, a Self-Calibration Camera system was developed with the aid of an UMI. This system consists of an internal camera parameter estimator based on a EKF that operates in an online manner and without any prior knowledge of the environment in which the camera is located. This method allows systems such as Visual Odometry to always operate with the correct values of the camera's internal parameters. From the self-calibrator developed, an Inertial Visual Localization system with Camera Self-Calibration was also designed. This second system was designed as the set of two UKFs that operate to estimate the position and orientation of a robotic agent, which carries a camera and a UMI, while calibrating the camera used. The dissertation also presents some experiments carried out in order to validate the two proposed methods.

## SUMÁRIO

---

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>1</b>
1.1	MOTIVAÇÃO .....	1
1.2	CONTEXTUALIZAÇÃO .....	2
1.3	DEFINIÇÃO DO PROBLEMA E OBJETIVOS .....	3
1.4	PROPOSTAS E CONTRIBUIÇÕES .....	5
1.5	APRESENTAÇÃO DO MANUSCRITO .....	5
<b>2</b>	<b>LOCALIZAÇÃO VISUAL INERCIAL .....</b>	<b>7</b>
2.1	INTRODUÇÃO .....	7
2.2	NAVEGAÇÃO INERCIAL .....	9
2.2.1	GIROSCÓPIO .....	11
2.2.2	ACELERÔMETRO .....	15
2.2.3	ALGORITMO DA NAVEGAÇÃO INERCIAL .....	20
2.2.3.1	RASTREANDO A ORIENTAÇÃO .....	20
2.2.3.2	RASTREANDO A POSIÇÃO .....	23
2.2.3.3	PROPAGAÇÃO DE ERROS .....	24
2.3	ODOMETRIA VISUAL .....	24
2.3.1	FORMULAÇÃO DO PROBLEMA .....	25
2.3.2	TEORIA .....	27
2.3.3	IMPLEMENTAÇÃO .....	30
2.4	FUSÃO SENSORIAL .....	33
2.4.1	CLASSIFICAÇÕES .....	33
2.4.2	FILTRO DE KALMAN COMO TÉCNICA DE FUSÃO SENSORIAL .....	34
2.4.3	VANTAGENS DA FUSÃO SENSORIAL VISUAL E INERCIAL .....	35
2.5	CALIBRAÇÃO DE CÂMERA .....	36
2.5.1	MODELO DA CÂMERA .....	36
2.5.2	IMPLEMENTAÇÃO DA CALIBRAÇÃO .....	39
<b>3</b>	<b>ABORDAGENS DESENVOLVIDAS .....</b>	<b>41</b>
3.1	INTRODUÇÃO .....	41
3.2	AUTO-CALIBRAÇÃO DE CÂMERA COM AUXÍLIO DE UMA UMI .....	42
3.2.1	DEFINIÇÃO DO PROBLEMA .....	42
3.2.2	ALGORITMO .....	43
3.2.2.1	ESTIMADOR .....	44
3.2.2.2	VETOR DE ESTADOS .....	46
3.2.2.3	FUNÇÃO DE PROPAGAÇÃO DE ESTADOS .....	47
3.2.2.4	MODELO DE MEDIÇÃO .....	47
3.2.2.5	JACOBIANOS .....	52
3.2.2.6	CONSIDERAÇÕES FINAIS .....	56
3.3	LOCALIZAÇÃO VISUAL INERCIAL COM AUTO-CALIBRAÇÃO DE CÂMERA (LOVIACC) .....	56
3.3.1	DEFINIÇÃO DO PROBLEMA .....	56
3.3.2	ALGORITMO .....	57

3.3.2.1 ODOMETRIA VISUAL .....	58
3.3.2.2 FILTRO DE KALMAN UNSCENTED .....	60
3.3.2.3 FUSÃO SENSORIAL DA ODOMETRIA VISUAL COM NAVEGAÇÃO INERCIAL .....	62
3.3.2.4 ESTIMADOR DE PARÂMETROS INTERNOS DA CÂMERA .....	67
3.3.2.5 CONSIDERAÇÕES FINAIS .....	72
<b>4 ANÁLISE DE DESEMPENHO .....</b>	<b>73</b>
4.1 INTRODUÇÃO .....	73
4.2 VALIDAÇÃO DA AUTO-CALIBRAÇÃO DE CÂMERA COM AUXÍLIO DE UMA UMI .....	73
4.2.1 DESCRIÇÃO DO EXPERIMENTO .....	74
4.2.2 RESULTADOS DAS ESTIMAÇÕES .....	75
4.3 VALIDAÇÃO DO SISTEMA LOVIACC .....	79
4.3.1 SIMULAÇÃO .....	79
4.3.2 ESTIMAÇÃO DOS PARÂMETROS .....	85
4.3.2.1 VARIAÇÃO DOS PARÂMETROS SEGUINDO UM DEGRAU .....	86
4.3.2.2 VARIAÇÃO DOS PARÂMETROS SEGUINDO UMA SENOIDE .....	88
4.3.3 ESTIMAÇÃO DA LOCALIZAÇÃO .....	90
4.3.4 CONSIDERAÇÕES FINAIS .....	96
<b>5 CONCLUSÕES .....</b>	<b>97</b>
5.1 CONSIDERAÇÕES FINAIS .....	97
5.2 PROPOSTAS PARA TRABALHOS FUTUROS .....	100
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>102</b>

## LISTA DE FIGURAS

---

2.1	Esquemático da combinação do sensoriamento visual e inercial. ....	8
2.2	Distinção de {G} e de {SNI}. ....	9
2.3	UMI de plataforma fixa. ....	10
2.4	Um giroscópio mecânico convencional. ....	11
2.5	Efeito Sagnac. As linhas pontilhadas representam o caminho percorrido pelo feixe de luz viajando na direção da rotação enquanto que as linhas sólidas representam o caminho percorrido pelo feixe de luz que viaja na direção oposta a rotação. $\theta$ é o ângulo que o giroscópio se movimentou enquanto os feixes de luz viajavam. ....	12
2.6	Giroscópio MEMS de massa vibratória. ....	13
2.7	Esquemático simplificado de um acelerômetro mecânico. ....	16
2.8	Esquemático de um acelerômetro do tipo SAW. ....	17
2.9	Ouvido de um ser humano. ....	19
2.10	O algoritmo da Navegação Inercial. ....	20
2.11	Representação de uma cena de múltiplas vistas, com câmeras em posições e orientações diferentes. A presença de pontos do objeto em imagens diferentes é utilizada na SFM para se estimar a posição 3D do ponto, bem como as posições e orientações das câmeras que capturaram as imagens. ....	25
2.12	Deslocamento de pixels correspondentes em dois <i>frames</i> consecutivos devido a movimentação da câmera. ....	29
2.13	Diagrama de blocos mostrando os principais componentes de um sistema de Odometria Visual. .	30
2.14	As quatro possíveis soluções para a reconstrução de $R$ e $t$ a partir de $E$ . ....	32
2.15	Esquemático da projeção de um ponto 3D em uma câmera <i>pinhole</i> . ....	37
2.16	Imagens capturadas para a calibração de uma câmera. ....	40
3.1	Esquemático da estimação dos parâmetros internos da câmera. ....	43
3.2	Esquemático do sistema de Auto-Calibração de Câmera com Auxílio de uma UMI. ....	54
3.3	Esquemático do algoritmo de LOVIACC proposto. ....	58
3.4	Esquemático de fluxo de dados do sistema de Odometria Visual utilizado no LOVIACC proposto. ....	59
3.5	Esquemático da fusão sensorial entre a Odometria Visual e as leituras da UMI. ....	63
3.6	Representação gráfica da linha do tempo mostrando possíveis momentos em que ocorrem capturas de novas imagens pela Câmera ou leituras dos sensores da UMI. ....	67
4.1	Montagem dos sensores utilizados na realização do experimento. Tem-se uma câmera Firefly FMVU-03MTM/C <i>global shutter</i> com a UMI Memsense Nano IMU fixamente acoplada ao corpo da câmera. ....	74
4.2	Imagens capturadas pela câmera enquanto a mesma era submetida a um movimento circular sobre uma mesa de laboratório. ....	74
4.3	Resultados da aplicação do FKE para as estimativas dos parâmetros internos da câmera ao longo do número do <i>frame</i> capturado. (a) Estimação do parâmetro $\alpha$ ; (b) Estimação do parâmetro $\beta$ ; (c) Estimação do parâmetro $\phi$ ; (d) Estimação do parâmetro $k$ . ....	76

4.4	Reconstrução da trajetória percorrida pela câmera enquanto foram realizadas as capturas das imagens. ....	78
4.5	Simulação do percurso realizado pelo agente robótico. (a) Representação 3D do percurso; (b) Apresentação das três componentes ( $x$ , $y$ e $z$ ) do percurso simulado. ....	80
4.6	Simulação da velocidade e aceleração da movimentação realizada pelo robô durante o experimento. (a) Componentes da velocidade simulada ao longo do tempo; (b) Componentes da aceleração simulada ao longo do tempo. ....	81
4.7	Ângulos de rotação ao longo dos três eixos principais ( $x$ , $y$ e $z$ ) que correspondem a orientação do centro de referência da UMI simulada. ....	82
4.8	Simulação das leituras dos acelerômetros da UMI. Gráficos apresentam as componentes da aceleração percebidas pelos três acelerômetros. ....	82
4.9	Simulação das leituras dos giroscópios da UMI. Gráficos apresentam as componentes da velocidade angular percebidas pelos três giroscópios. ....	83
4.10	Geração de imagem sintética utilizada nos experimentos. (a) Nuvem de pontos 3D posicionados no espaço para simular pontos de interesse em uma imagem; (b) Imagem sintética gerada a partir da projeção dos pontos 3D do espaço no plano de imagem. ....	84
4.11	Resultados das estimações dos parâmetros com os valores reais seguindo uma função degrau. Em azul estão representados os valores reais dos parâmetros e, em vermelho, os valores estimados com o FKU. (a) Estimação do parâmetro $\alpha$ ; (b) Estimação do parâmetro $\beta$ ; (c) Estimação do parâmetro $\phi$ ; (d) Estimação do parâmetro $u_0$ ; (e) Estimação do parâmetro $v_0$ . ....	87
4.12	Resultados das estimações dos parâmetros com os valores reais seguindo uma função seno. Em azul estão representados os valores reais dos parâmetros e, em vermelho, os valores estimados com o FKU. (a) Estimação do parâmetro $\alpha$ ; (b) Estimação do parâmetro $\beta$ ; (c) Estimação do parâmetro $\phi$ ; (d) Estimação do parâmetro $u_0$ ; (e) Estimação do parâmetro $v_0$ . ....	89
4.13	Resultado das estimações da fusão sensorial para a posição e velocidade. Em azul está representado o <i>ground truth</i> da movimentação e, em vermelho, as estimações. (a) Evolução das estimativas das componentes da posição do robô ao longo do tempo; (b) Evolução das estimativas das componentes da velocidade do robô ao longo do tempo. ....	91
4.14	Estimativas das componentes da posição do agente robótico sendo rastreado apenas com o uso de um Sistema de Navegação Inercial. Foram utilizadas as mesmas leituras da UMI simuladas para o experimento. ....	92
4.15	Representação 3D da comparação entre o percurso estimado e o simulado. Em azul se encontra a trajetória real do robô e, em vermelho, a trajetória estimada. ....	93
4.16	Evolução das estimativas das componentes da orientação do agente robótico. Em azul está a verdadeira angulação simulada e, em vermelho, os valores estimados para cada uma das rotações ao redor dos três eixos. ....	93
4.17	Resultado da estimação da escala $\lambda$ da Odometria Visual obtida com a fusão sensorial. (a) Evolução da estimação de $\lambda$ ao longo do tempo. Em azul está o valor real da escala e, em vermelho, o valor estimado; (b) Comparativo do erro da estimação, representado em vermelho, com $\pm 3\sigma$ , apresentado em verde, referente a incerteza da estimação. ....	95

## LISTA DE TABELAS

---

2.1	Resumo das fontes de Erro em Giroscópios MEMS .....	15
2.2	Resumo das fontes de Erro em acelerômetros MEMS .....	19
4.1	Tabela resumo dos parâmetros. ....	77
4.2	Valores dos parâmetros utilizados na sintetização de imagens ao longo da simulação do primeiro experimento. ....	86

## LISTA DE SÍMBOLOS

---

### Símbolos Latinos

$F_c$	Força Coriolis
$m$	Massa
$v$	Velocidade Linear
$t$	Tempo em segundos
$E(\cdot)$	Operador de Esperança
$Var(\cdot)$	Operador de Variância
$Cov(\cdot)$	Operador de Covariância
$F$	Força
$s$	Posição linear
$N_i$	$i$ -ésima variável aleatória em uma sequência
$v_g$	Velocidade linear em relação ao {G}
$v_s$	Velocidade linear em relação ao {SNI}
$R$	Matriz de Rotação
$a_s$	Aceleração em relação ao {SNI}
$a_g$	Aceleração em relação ao {G}
$g_g$	Aceleração da gravidade em relação ao {G}
$I_i$	$i$ -ésima imagem em uma sequência de capturas
$T_{k,k-1}$	Transformação de corpo rígido entre instantes $k$ e $k - 1$
$R_{k,k-1}$	Matriz de rotação que compõe $T_{k,k-1}$
$t_{k,k-1}$	Vetor de translação que compõe $T_{k,k-1}$
$T_{1:n}$	Conjunto de transformações de corpo rígido
$C_{0:n}$	Conjunto de poses da câmera
$C_i$	$i$ -ésima pose da câmera
$P$	Ponto 3D no {CAM}
$X, Y, Z$	Coordenadas de $P$
$p$	Projeção de $P$ na imagem da câmera
$u, v$	Coordenadas, em pixel, de $p$ na imagem
$f$	Distância focal da câmera
$u_0, v_0$	Coordenadas do ponto principal da câmera em pixels
$E$	Matriz Essencial
$\hat{p}$	Coordenadas normalizadas do ponto $p$
$x$	Vetor de estados de um sistema dinâmico
$y$	Vetor de medidas de um sistema dinâmico
$P_t$	Matriz de covariância associada aos estados de um sistema dinâmico no instante $t$
$K_t$	Ganho de Kalman no instante $t$
$u_t$	Vetor de controle no instante $t$
$I$	Matriz identidade
$0$	matriz nula
$R_g^{cam}$	Matriz de rotação entre {G} e {CAM}

$\mathbf{t}_g^{cam}$	Vetor de translação entre {G} e {CAM}
$\mathbf{K}$	Matriz da câmera
$\mathbf{R}_{cam}^{umi}$	Matriz de rotação entre {CAM} e {SNI}
$\mathbf{t}_{cam}^{umi}$	Vetor de translação entre {CAM} e {SNI}
$k_n$	$n$ -ésimo elemento do polinômio $\xi$
$\mathbf{x}_t$	Vetor de estados no instante $t$
$f(\mathbf{x}_t)$	Função de propagação de estados
$\mathbf{y}_t$	Vetor de medições no instante $t$
$h(\mathbf{x}_t)$	Modelo de medições
$\hat{\mathbf{x}}_t'$	Estimação a priori
$\hat{\mathbf{x}}_t$	Estimação a posteriori
$F_t$	Jacobiano de $f(x_t)$
$H_t$	Jacobiano de $h(x_t)$
$e_u$	Coordenada $u$ da epípola
$e_v$	Coordenada $v$ da epípola
$\mathbf{R}_{g,t}^{cam}$	Matriz de rotação entre {G} e {CAM} no instante $t$
$\mathbf{t}_{g,t}^{cam}$	Vetor de translação entre {G} e {CAM} no instante $t$
$\mathbf{q}_t$	Quatérnio unitário que representando a orientação no instante $t$
$\mathcal{X}$	Conjunto de pontos sigma do FKU
$T$	Intervalo de tempo entre amostras da UMI

## Símbolos Gregos

$\omega$	Velocidade angular
$\epsilon$	Viés de sensor Inercial
$\theta$	Posição angular
$\sigma$	Desvio padrão
$\delta_t$	Intervalo entre leituras de uma UMI
$\omega_s$	Velocidade angular em relação ao {SNI}
$\delta_{\theta_x}, \delta_{\theta_y}, \delta_{\theta_z}$	Pequenas rotações em relação aos eixos $x$ , $y$ e $z$
$\alpha$	Parâmetro interno da câmera proporcional a $f$ na direção $u$
$\beta$	Parâmetro interno da câmera proporcional a $f$ na direção $v$
$\phi$	Angulo entre eixos $u$ e $v$ do sensor da câmera
$\xi$	Função polinomial que modela a distorção radial da lente
$\varepsilon_{x,t}$	Ruído gaussiano branco que perturba a propagação de estados
$\varepsilon_{y,t}$	Ruído gaussiano branco que perturba o modelo de medições
$\Delta u_t$	Deslocamento da coordenada $u$ de um pixel
$\Delta v_t$	Deslocamento da coordenada $v$ de um pixel
$\Lambda_t$	Variável auxiliar que representa $X$ de {G} no {CAM} no instante $t$
$\mu_t$	Variável auxiliar que representa $Y$ de {G} no {CAM} no instante $t$
$\Gamma_t$	Variável auxiliar que representa $Z$ de {G} no {CAM} no instante $t$
$\lambda$	Escala da Odometria Visual
$\Sigma_t$	Matriz de covariância da estimativa do FKU
$\boldsymbol{\mu}_t$	Esperança da estimativa do FKU
$\tau$	Variável auxiliar no cálculo dos pontos sigma do FKU

$\rho$	Variável auxiliar no cálculo dos pontos sigma do FKU
$\kappa$	Variável auxiliar no cálculo dos pontos sigma do FKU
$\omega_m^{[i]}$	Peso utilizado no calculo da média para o $i$ -ésimo ponto sigma do FKU
$\omega_c^{[i]}$	Peso utilizado no calculo da covariância para o $i$ -ésimo ponto sigma do FKU
$\gamma$	Leituras dos acelerômetros
$\Omega$	Leituras dos giroscópios
$\delta_{\theta,t}$	Ângulos de rotação ao redor dos eixos do {G}

## Siglas

GPS	<i>Global Positioning System</i>
FKE	Filtro de Kalman Estendido
FKU	Filtro de Kalman Unscented
OV	Odometria Visual
FK	Filtro de Kalman
UMI	Unidade de Medição Inercial
CMOS	<i>Complementary Metal Oxide Semiconductor</i>
MEMS	<i>Micro-Electro-Mechanical Systems</i>
NI	Navegação Inercial
LIDAR	<i>Light Detection And Ranging</i>
LOVIACC	Localização Visual Inercial com Auto-Calibração da Câmera
LVI	Localização Visual Inercial
SNI	Sistema de Navegação Inercial
{G}	Sistema de coordenadas Globais
{CAM}	Sistema de coordenadas da Câmera
{IMG}	Sistema de coordenadas da Imagem
FOG	<i>Fiber Optic Gyroscope</i>
SFM	<i>Structure From Motion</i>

## NOTAÇÃO

---

Neste trabalho vetores são representados por letras minúsculas em negrito. Matrizes são representadas por letras maiúsculas em negrito.

## 1.1 MOTIVAÇÃO

Nesta seção serão apresentadas algumas situações nas quais o leitor poderá perceber a relevância do estudo apresentado nesta dissertação. A ideia é que o leitor se sinta motivado, ou instigado, a se aprofundar no estudo da localização de robôs móveis fazendo uso de sensores visuais e inercias.

Imagine-se em um mundo em que todos os carros que circulam nas ruas das cidades são autônomos, isto é, são carros que dirigem sozinhos, sem qualquer necessidade de motorista ou piloto à distância. Neste cenário, as cidades, as pessoas que nelas vivem e até mesmo o meio ambiente poderiam usufruir de diversas vantagens[1]. O número de carros poderia ser reduzido, uma vez que o uso de carros compartilhados seria mais difundido, o que por si só já reduziria os trânsitos e congestionamentos, reduzindo o tempo gasto em transporte para todos. Com menos tempo gasto no deslocamento, ou mesmo podendo fazer outras atividades enquanto o carro dirige para elas, as pessoas poderiam aproveitar mais seus momentos de lazer. As cidades e o meio ambiente podem desfrutar dos benefícios da redução do número de veículos circulando, que vão desde menos manutenção para a estrutura viária urbana até uma redução drástica da emissão de poluentes. O número de acidentes seria reduzido a quase zero, uma vez que os carros poderiam se comunicar entre si, compartilhando sua posição, velocidade e rota (seria o fim das batidas de carros ocasionadas pela falta de seta por parte de um motorista), evitando assim quase todas as colisões. Além disso, não haveria mais necessidade de fiscalização ou multas de trânsito, já que os carros sempre seguiriam as regras de condução e jamais dirigiram embriagados. As vantagens não param por aí, pode-se ainda perceber que os espaços destinados a estacionamento poderiam ser reaproveitados para outros fins, uma vez que os carros não precisariam mais ficar estacionados sem utilidade durante todo o dia de trabalho de seu proprietário, poderia circular pelas ruas facilitando o deslocamento de outras pessoas. Esse mundo dos carros autônomos não está muito distante. As pesquisas têm avançado muito nos últimos anos, e já existem testes sendo feitos com este tipo de veículos em cidades reais[2, 3]. Mas para estes carros autônomos, que nada mais são que robôs móveis com a finalidade de realizar transporte, funcionarem, é de extrema importância que os mesmos tenham conhecimento de onde eles estão, por onde estão passando, a que velocidade estão se deslocando e etc. Em outras palavras, é necessário que eles tenham como rastrear a sua Localização. Pode-se pensar facilmente que estes robôs poderiam fazer uso do GPS (*Global Positioning System*) para realizar esta localização e obter as demais informações a respeito de sua movimentação. Mas é sabido que existem certas condições para o funcionamento correto do GPS, condições estas que não são satisfeitas, por exemplo, quando o robô, ou carro, passa por um túnel e perde a comunicação com o satélite. Como o robô faria para se localizar em situações como esta?

Outra situação em que a robótica móvel pode ajudar bastante o ser humano é com o auxílio a missões de resgate[4]. Imagine que uma cidade foi assolada por um terremoto de grande magnitude. As estradas foram destruídas ou obstruídas por destroços, mapas das ruas já não ajudam tanto no deslocamento, e é necessário fazer uma varredura por todos os escombros para buscar por sobreviventes. Robôs podem ser utilizados para auxiliar nestas buscas trabalhando de maneira muito mais rápida, eficiente e sem pausas noturnas, o que poderia salvar muitas vidas. Agora, da mesma maneira que o exemplo dos carros autônomos, estes robôs de resgate, que podem ser terrestres, aquáticos ou mesmo aéreos, precisam saber por onde andam, precisam saber onde

estão. E em situações de desastre como a descrita, dificilmente o robô contará com auxílio de mapas ou pontos de referência devidamente conhecidos para auxiliar na sua localização. E sinal de GPS nem sempre estará disponível ou será de grande ajuda, por exemplo, quando o robô precisar realizar uma varredura nos escombros de um prédio. Mais uma vez fica a questão: como o robô pode saber onde está?

Felizmente, para contornar essas adversidades e tornar possíveis todas estas aplicações de robôs, os projetistas e pesquisadores de robótica móvel já apresentam diversas soluções de localização que precisam apenas de sensores embarcados nos próprios robôs para funcionar. Por vezes, estas soluções envolvem mais de um sensor, e saber combinar as informações deles é crucial, seja fazendo uso das vantagens de um sensor para corrigir as desvantagens de outros ou mesmo para inferir informações que nenhum sensor poderia medir sozinho.

## 1.2 CONTEXTUALIZAÇÃO

Como apresentado na seção anterior, dar ao robô a capacidade de se localizar é de extrema importância. E para que o mesmo tenha maior autonomia e robustez, é crucial existir certa redundância a respeito de seu sensoriamento, para o caso de um sensor de localização vir a falhar, ele ainda é capaz de fazer inferências a respeito de seu posicionamento com seus outros sensores[5].

Executar a combinação das leituras de dois ou mais sensores pode ainda ser uma possível forma de se realizar a localização de um robô móvel. Esta combinação de sensores é chamada de fusão sensorial[6]. Ela pode ser empregada em casos em que se deseja obter uma maior confiança em relação as leituras do que as provenientes dos sensores isolados, ou ainda, encontrar estimativas de grandezas que não podem ser diretamente medidas por um único sensor. Dentre as maneiras de se realizar esta fusão, destaca-se a técnica por meio da Filtragem Estocástica[7].

O Filtro de Kalman é uma poderosa ferramenta estocástica muito utilizada na robótica móvel. Ele é empregado em diversas situações em que se deseja realizar estimações. Seu algoritmo apresenta um formato muito bem estruturado e definido; dividido em duas etapas principais: Predição e Medição. Na etapa de Predição, o filtro faz uso do conhecimento do modelo matemático do sistema em análise para realizar a predição da evolução das grandezas estimadas. Na etapa de Medição, o filtro faz uso de leituras de outras grandezas, que podem ser as mesmas que estão sendo estimadas ou grandezas a elas correlatas, para executar uma espécie de correção sobre a estimativa obtida na etapa de Predição. Esta correção se dá por meio de uma “ponderação estocástica” que atribui “pesos” às estimativas ou leituras baseadas no grau de confiança que elas possuem[8, 7].

Este algoritmo descrito permite interpretar o método de estimação do Filtro de Kalman como uma combinação das suas duas etapas. Esta combinação é muitas vezes utilizada como o método de se desempenhar a fusão sensorial entre sensores para se realizar a localização de robôs móveis. O Filtro de Kalman só pode ser empregado em situações nas quais a modelagem do sistema em questão é linear, porém existem variações do Filtro de Kalman que podem atender eventuais necessidades de filtragem em casos não lineares, como Filtro de Kalman Estendido (FKE) e o Filtro de Kalman Unscented (FKU)[8].

Sabendo que se pode realizar esta importante combinação de sensores para se estimar a localização de robôs, resta saber quais sensores seriam estes, pois, como observa-se no caso do GPS, existem situações em que sensores podem simplesmente não funcionar, ou fornecer leituras não muito precisas[5]. Sendo assim, é importante realizar a escolha de sensores que possam complementar a leitura um do outro, para que as vantagens de um possa compensar os defeitos dos outros sensores.

Uma possível combinação de sensores que podem fornecer ao robô a sua posição é a combinação entre os sensores inerciais, como acelerômetros e giroscópios, e sensores visuais[9, 10, 11, 12, 13, 14]. Acelerômetros e giroscópios são sensores que medem a aceleração e a orientação, respectivamente, de um agente que os porte. A partir de medidas destes sensores ao longo do tempo, e das formulações da cinemática, é possível reconstruir a trajetória desempenhada pelo agente.

Já sensores visuais são basicamente câmeras e suas “medidas” são capturas de imagens do ambiente em que estão. Só que, a partir de uma técnica da visão computacional chamada Odometria Visual[15, 16], é possível, a partir de uma sequência de imagens capturadas por uma câmera, fazer a reconstrução da trajetória percorrida pela câmera enquanto a mesma capturava a sequência de imagens, sendo assim, se o robô se movimenta carregando uma câmera, a partir das imagens capturadas por ela, é possível descobrir qual foi o trajeto percorrido pelo robô.

Esses dois métodos de reconstrução da trajetória percorrida pelo robô podem ser fundidos com uma variação do Filtro de Kalman[9]. E se isto ocorrer, será possível obter uma estimativa da posição do robô mais confiável do que a obtida se estivesse sendo usado apenas um destes sensores. Isso se dá pois existem situações nas quais os sensores inerciais não conseguem operar corretamente e existem grandezas que a Odometria Visual simplesmente não é capaz de recuperar se fizer uso apenas de imagens capturadas por uma única câmera[15].

Essa localização obtida pela combinação destes sensores é chamada de Localização Visual Inercial[10]. E pode ser empregada nas mais diversas situações. Voltando ao exemplo dos carros autônomos, pode-se pensar que, quando o carro estiver passando por um túnel e perder o seu sinal de GPS, sua estimativa de posição pode ser desempenhada pela Localização Visual Inercial. Um VANT (Veículo Aéreo Não Tripulado) autônomo empregado em situações de resgate pode fazer uso da Localização Visual Inercial dentro de prédios ou subsolos nos quais uma localização baseada em satélites não seria possível. E claro, não é impossível imaginar que a Localização Visual Inercial pode ainda ser combinada a outros sensores a fim de se obter uma estimativa a respeito da posição do robô ainda mais confiável. Para isso, vale a máxima de que “dois sensores são melhores do que um (desde que sejam não enviesados)”[7, 5].

### **1.3 DEFINIÇÃO DO PROBLEMA E OBJETIVOS**

Como mostrado na seção anterior, a Localização Visual Inercial é uma possível solução para o problema de localização de robôs móveis. Como apresentado, este sistema de navegação faz a combinação de sensores inerciais com as estimativas obtidas com uma Odometria Visual.

O processo de recuperação da trajetória percorrida pela câmera executado pela Odometria Visual é baseado na procura de pontos comuns a duas imagens consecutivas. Espera-se que, devido ao movimento desempenhado pela câmera entre uma captura e outra, estes pontos estejam em locais diferentes na imagem, isto é, em coordenadas de pixel distintas. É justamente a partir deste deslocamento de pixels em conjunto com o conhecimento da modelagem matemática que descreve a câmera que a Odometria Visual consegue recuperar a movimentação ocorrida entre as capturas[15, 4, 17].

Esta modelagem da câmera consiste em conseguir enunciar como, de maneira matemática, se dão as projeções de pontos tridimensionais do mundo em pontos bidimensionais da imagem, deixando clara a correlação entre as coordenadas do ponto e da sua projeção[18].

Para que seja possível enunciar como se dá esta projeção, ou melhor, para se modelar a câmera, é necessário ter conhecimento de alguns entes físicos que são diretamente relacionados à formação da imagem da câmera. Por exemplo, a distância focal, o tamanho e formato do sensor que capta cada pixel da imagem são alguns destes entes físicos necessários[19, 18]. É comum se referir a estes elementos físicos da câmera como Parâmetros Internos. Sendo assim, para que a Odometria Visual seja efetiva é necessário saber, com a maior exatidão possível, quais são os parâmetros internos da câmera utilizada.

A maneira mais utilizada para medir quais são os parâmetros internos de uma câmera consiste em se realizar um procedimento chamado de Calibração de Câmera[20, 21]. A calibração é normalmente realizada de maneira *offline*, com o auxílio de alguns elementos conhecidos no ambiente em que a câmera se encontra e antes de se utilizar a câmera para a aplicação que se deseja (como por exemplo a Odometria Visual). Ou seja, a calibração é realizada, os parâmetros são mensurados e utilizados na aplicação como se fossem constantes.

Só que existe um problema com este procedimento tradicional de calibração de câmeras: há situações que podem alterar os parâmetros internos da câmera, como a câmera sofrer algum choque mecânico ou o robô que a porta sofrer uma trepidação muito intensa[22]. Quando isso acontece durante a execução de um algoritmo de Odometria Visual, o modelo da câmera utilizado deixa de ser fiel às projeções que estão acontecendo nas capturas de imagens e, conseqüentemente, as estimações de movimentação não são mais confiáveis. A maneira de voltar a se ter uma estimativa confiável é submeter a câmera novamente ao procedimento de calibração para que se obtenha os novos valores dos parâmetros internos.

Imagine novamente o robô de resgate em ação. Não é de se estranhar que possam acontecer deslizamentos ou quedas de fragmentos que possam por ventura acertar a câmera do robô. No caso dos carros autônomos, é bem provável que em algumas estradas pelas quais o carro passe a situação do pavimento esteja ruim, o que faria o carro, juntamente com todos os seus sensores, trepidar bastante. Essas são situações pelas quais robôs móveis que carregam câmera podem passar e que podem alterar os parâmetros internos das mesmas.

Sistemas que utilizem os parâmetros internos da câmera, como a Odometria Visual e, conseqüentemente, a Localização Visual Inercial, precisam ser submetidos a constantes calibrações de câmera[22]. Isso limita bastante o grau de confiança que se dá em suas estimativas, uma vez que o método mais comum de calibração se dá de maneira *offline* e precisa de preparação do ambiente para ser executado. Dessa forma, fica evidente que um novo método de se realizar a calibração de câmera de maneira *online* e sem que haja qualquer necessidade de conhecimento prévio do ambiente em que a câmera se encontra é muito bem vindo. Com tal técnica, o robô de resgate, o carro autônomo ou qualquer outro robô móvel que precise de uma câmera, pode realizar a calibração dos parâmetros internos a todo instante, o que fará com que estimativas provenientes de sistemas como a Odometria Visual e a Localização Visual Inercial sejam muito mais confiáveis.

Este é justamente um dos objetivos desta dissertação. Ao longo do trabalho serão apresentadas duas maneiras de se estimar os parâmetros internos da câmera de forma *online* e sem conhecimento prévio do ambiente. A pesquisa reportada neste trabalho iniciou-se com a pergunta: “é possível realizar a auto-calibração da câmera, isto é, será possível, a partir de informações da imagem capturada pela própria câmera, realizar a calibração?”. A resposta positiva veio com a primeira técnica desenvolvida, onde a partir das imagens sequenciais de uma câmera em conjunto com leituras de uma UMI (Unidade de Medição Inercial), acoplada ao corpo da câmera, foi possível realizar a calibração.

A segunda técnica desenvolvida e apresentada neste trabalho, dá um passo além de simplesmente calibrar a câmera. Aproveitando que houve a utilização de uma UMI para realizar a calibração, expandiu-se a primeira técnica a fim de se realizar um sistema de Localização Visual Inercial com Auto-Calibração de Câmera, isto é, desenvolveu-se um método de se realizar a Localização Visual Inercial em paralelo com a Auto-Calibração da

Câmera. O que confere à técnica, uma maior confiança nas suas estimativas quando comparada a uma simples Localização Visual Inercial.

## 1.4 PROPOSTAS E CONTRIBUIÇÕES

Dados os objetivos de se propor uma técnica que fosse capaz de realizar a calibração de uma câmera de maneira *online*, a pesquisa reportada nesta dissertação caminhou em direção a mostrar que esta auto-calibração pode ser feita por um Filtro de Kalman Estendido. Neste tipo de formulação, os parâmetros internos da câmera são justamente os parâmetros a serem estimados pelo filtro. Claro, foi necessário estudar bem o modelo da câmera para que fosse possível projetar o filtro de maneira adequada.

Esta primeira linha de pesquisa que fez uso do FKE para realizar a estimação, acarretou na escrita e publicação do artigo:

- **Online Inertial-Aided Monocular Camera Self-Calibration [23] - Resumo:** Neste artigo é apresentado um método de se realizar a calibração de uma câmera de maneira *online* e sem qualquer conhecimento prévio a respeito do ambiente. Esta técnica é baseada nas leituras de uma UMI, fixamente acoplada ao corpo da câmera, e no rastreamento de um único ponto 3D projetado nas imagens capturadas pela câmera enquanto a mesma se movimenta livremente no espaço. Este método foi projetado para ser utilizado em conjunto com técnicas de Odometria Visual ou de Navegação Visual Inercial pois, nestes sistemas, é necessário o conhecimento dos parâmetros internos da câmera utilizada. Normalmente, este tipo de sistema faz uso de métodos *offline* de calibração de câmera e têm de tomar seus resultados como constantes. Porém, é sabido que existem situações que podem alterar os parâmetros internos da câmera e, conseqüentemente, prejudicam sistemas que dependem do conhecimento destes parâmetros. O método proposto neste artigo pode ser implementado em conjunto com estes sistemas para que se possa sempre ter uma estimativa atual dos parâmetros. Além disso, o método fornece uma estimativa a respeito da movimentação realizada pela câmera enquanto o método está em execução. A técnica é baseada em um FKE que faz as estimativas dos parâmetros. A eficiência do método é comprovada com a realização de um experimento em mundo real que demonstra ser possível calibrar uma câmera com a técnica proposta.

Como já mencionado, o método se mostrou eficaz na tarefa de realizar a auto-calibração da câmera e deu o pontapé inicial no desenvolvimento da pesquisa que visava o projeto de um sistema de Localização Visual Inercial com Auto-Calibração de Câmera.

## 1.5 APRESENTAÇÃO DO MANUSCRITO

Esta dissertação está organizada em cinco capítulos. O primeiro destes cinco foi exatamente esta introdução ao trabalho e tinha como objetivo motivar o leitor e fazê-lo perceber a importância do conteúdo estudado. Um aprofundamento detalhado a respeito da Localização Visual Inercial é apresentado no Capítulo 2. Este aprofundamento visa deixar o leitor a par do funcionamento de cada sensor utilizado, conhecendo desde os princípios físicos que tornam possível o sensor realizar as medidas até as causas de seus problemas de funcionamento, o capítulo ainda apresenta como realizar a fusão sensorial e quais as vantagens de se utilizar este tipo de técnica

para melhorar as estimativas que podem ser realizadas a partir das leituras dos sensores. Ainda no capítulo 2 é apresentada como funcionam as técnicas tradicionais de calibração de câmera e seus problemas práticos. Resumindo, este capítulo tem como objetivo dar toda uma base teórica em conjunto com uma revisão bibliográfica do tema ao leitor, para que ele siga o restante da dissertação sem qualquer defasagem.

O Capítulo 3 é onde são apresentadas as duas abordagens desenvolvidas. Inicialmente, será apresentado o primeiro método desenvolvido para realizar a auto-calibração de uma câmera, que resultou na escrita e publicação do artigo. Na sequência, é apresentado o algoritmo do sistema de Localização Visual Inercial com Auto-Calibração de Câmera desenvolvido a partir de todos os resultados da pesquisa obtidos até então.

No capítulo 4 são apresentados os resultados e análises de experimentos realizados a fim de se validar os dois métodos propostos. Por fim, no capítulo 5 são apresentadas as conclusões a que se espera que o leitor tenha conseguido chegar ao longo de toda a leitura.

# 2

## LOCALIZAÇÃO VISUAL INERCIAL

---

### 2.1 INTRODUÇÃO

No reino animal, os animais sempre desenvolveram formas de perceber o mundo para ajudá-los nas mais diversas tarefas vitais[24, 25]. Ornitorrincos, por exemplo, conseguem localizar suas presas, mesmo quando em águas turvas, se utilizando de sensores eletrorreceptores presentes em seu bico; Algumas aves orientam o curso de voo a partir de uma percepção do campo magnético da Terra e morcegos são capazes de detectar obstáculos e o ambiente a sua volta a partir do seu sistema de ecolocalização[24].

No caso dos seres humanos, é comum que se diga que existem cinco sentidos: visão, audição, paladar, tato e olfato. Mas estes não são os únicos meios pelos quais os seres humanos percebem o mundo. Os humanos ainda são capazes, por exemplo, de combinar uma série de sensores biológicos, como as estruturas responsáveis pela noção de equilíbrio e movimento presentes do ouvido interno, a percepção do posicionamento articular e a força exercida pelos músculos para formar a chamada propriocepção, também conhecida como cinestesia[25]. A cinestesia é a capacidade de perceber o movimento e posição do próprio corpo, sem mesmo precisar da visão.

Quando se fala de robótica móvel, é crucial que os robôs, assim como os seres humanos, percebam a sua posição, ou ainda, tenham um senso de localização e movimentação para que possam executar as suas tarefas para as quais foram projetados[26, 5, 6]. Para isso, os robôs contam com diversos sensores e técnicas de fusão sensorial que foram desenvolvidos a certa semelhança dos sentidos presentes e das técnicas praticadas pelos animais na natureza. Giroscópios e acelerômetros podem ser utilizados para dar ao robô informações a cerca de sua movimentação, um sensor ultrassônico pode fornecer uma informação de distância de maneira semelhante a que os morcegos utilizam e uma câmera pode proporcionar dados similares ao que a visão apresenta para os seres humanos. Os robôs contam ainda com alguns sensores que não se tem conhecimento de sentido análogo presente em animais na natureza, como por exemplo o GPS (*Global Positioning System*) e o LIDAR (*Light Detection And Ranging*), para auxiliá-los na sua tarefa de localização[25].

Neste capítulo, será apresentada uma espécie de propriocepção para a robótica. Mais precisamente, será discutida uma forma de dar ao robô informações a respeito de sua localização e orientação a partir de dados coletados dos sensores de uma UMI (Unidade de Medição Inercial), ou seja, de giroscópios e acelerômetros, e das pistas visuais obtidas com o auxílio de uma câmera, não surpreende portanto que este tipo de técnica recebe o nome de Localização Visual Inercial[9, 10]. Serão mostrados os dois sensores (visual e inercial), cobrindo desde o funcionamento dos sensores análogos presentes na natureza aos princípios físicos que os tornam complementares entre si, cada um cobrindo as deficiências do outro, na tarefa de fornecer ao robô a sua localização.

Analisar a ordem derivativa que os sensores conseguem perceber o espaço é uma ótima maneira de classificá-los e também de saber suas características[25, 27, 28]. Por exemplo, o GPS é um sensor que consegue, através de uma triangulação de satélites, entregar ao robô dados relacionados diretamente a sua posição no espaço, isto é, trata-se de um sensor de ordem derivativa 0. Técnicas de visão computacional empregadas com os dados coletados de uma câmera também estão nesta categoria de sensores, pois, assim como o GPS, conseguem

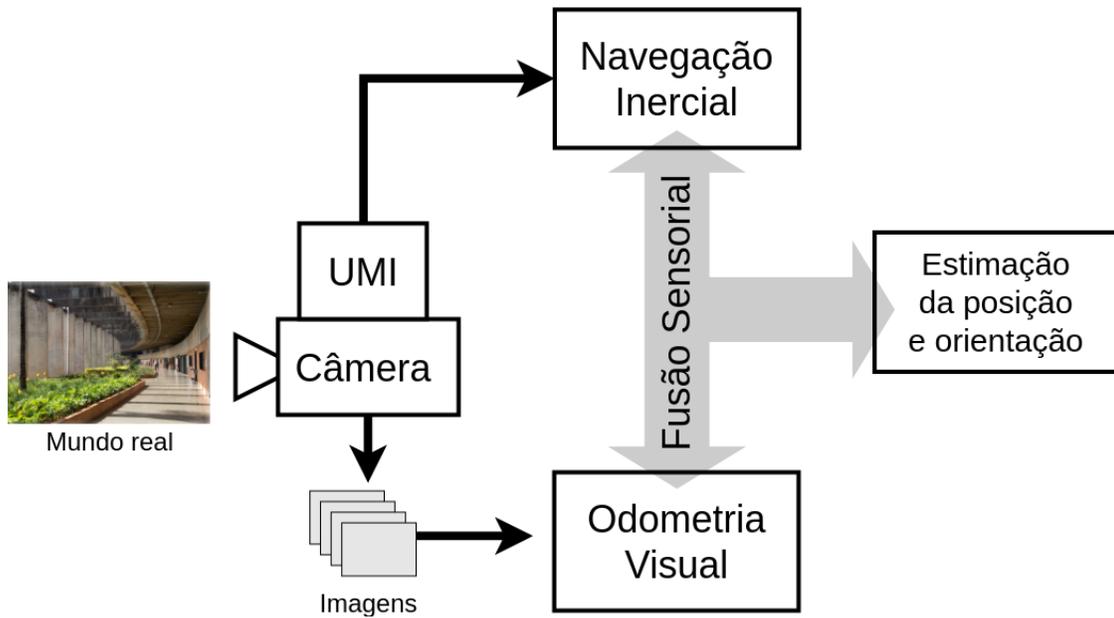


Figura 2.1: Esquemático da combinação do sensoriamento visual e inercial.

disponibilizar para o robô informações a respeito de sua posição (como será mostrado na seção 2.3, que trata a respeito da Odometria Visual). Os giroscópios de uma UMI, que são responsáveis por medir a velocidade angular, apresentam ordem derivativa 1 e os acelerômetros lineares de uma UMI são sensores de ordem derivativa 2, já que são capazes de medir a aceleração. Sendo assim, combinando as leituras dos sensores de uma UMI (giroscópios e acelerômetros) com a estimativa de movimentação de um sistema de Odometria Visual, tem-se um “sensor” robótico que apresenta informações de todas as três ordens derivativas espaciais.

A Figura 2.1 apresenta ao leitor um resumo visual de como funciona essa propriocepção robótica, ou Localização Visual Inercial. Tem-se representada uma câmera monocular fixamente acoplada a uma UMI no corpo de um robô. A câmera é utilizada para capturar uma sequência de *frames*, os quais são usados como entrada do sistema de Odometria Visual, que fornece como saída a estimativa da posição e orientação da câmera que capturou os *frames*. Já os dados lidos da UMI (velocidade angular e aceleração linear) são apresentados como entrada para o Sistema de Navegação Inercial, que fará integrações para estimar a localização e orientação da UMI no espaço. Uma vez que estão disponíveis os resultados da Navegação Inercial e da Odometria Visual, entra em ação a fusão sensorial para que se aproveite as melhores informações de cada uma das saídas a fim de se obter, com uma confiança mais elevada, uma estimativa da localização do robô.

Os preços de mercado de câmeras CMOS (*Complementary Metal-Oxide Semiconductor*), giroscópios e acelerômetros MEMS (*Micro-Electro-Mechanical Systems*) vem caindo cada vez mais nos últimos anos [28, 25, 22]. Isso fez com que esses tipos de sensores estivessem cada vez mais presentes em pesquisas relacionadas a robótica e, conseqüentemente, vêm se mostrando uma alternativa barata e viável para se projetar sistemas de navegação e localização para robôs móveis.

O restante deste capítulo está organizado da seguinte maneira: Na seção 2.2, será apresentado o funcionamento dos sensores de uma UMI, bem como a teoria por trás da Navegação Inercial, evidenciando seus pontos fracos e fortes. A seção 2.3 apresenta a definição de Odometria Visual e mostra como é possível extrair a estimativa da movimentação de um robô a partir, unicamente, de uma sequência de *frames* capturados com uma câmera monocular fixamente acoplada ao seu corpo. Em seguida, a seção 2.4 apresenta como melhorar a confi-

ança na estimativa da localização do robô móvel utilizando-se das saídas da Odometria Visual e da Navegação Inercial. Por fim, a seção 2.5 traz um panorama de como é feita a calibração de câmeras, que se mostrará um importante requisito de uma boa Odometria Visual.

## 2.2 NAVEGAÇÃO INERCIAL

Dá-se o nome de Navegação Inercial para o conjunto de técnicas que são empregadas para que, a partir de medidas de acelerômetros e giroscópios, sejam estimadas a posição e orientação de um agente (por exemplo um robô, ser humano, ou veículo) com um dado ponto, orientação e velocidade iniciais[29, 25, 28]. Interessante ressaltar também que a Navegação Inercial é auto-contida, isto é, ela independe de configuração prévia do ambiente ou de outros sensores externos.

Giroscópios e acelerômetros são conhecidos como sensores inerciais por conseguirem fazer medias explorando a propriedade da inércia, ou seja, realizam medições a cerca da resistência a mudança no momento[25]. No caso do giroscópio essa resistência faz com que seja possível medir os deslocamentos angulares e, no caso dos acelerômetros, com que seja possível perceber alterações na aceleração linear. Uma importante propriedade dos sensores inerciais é que eles independem de referências externas a menos, é claro, da própria gravidade. Apesar de existirem outros sensores inerciais, como os inclinômetros que conseguem fazer suas medidas a partir do vetor da aceleração da gravidade por ele percebido, os giroscópios e acelerômetros receberão uma atenção especial neste trabalho por serem os sensores inerciais mais comuns nas UMIs.

UMIs são normalmente compostas por três giroscópios ortogonais entre si e por três acelerômetros, também ortogonais entre si. Os giroscópios tem a função de medir as componentes da velocidade angular e os acelerômetros são empregados para detectar o vetor de aceleração linear. Processando os dados lidos por uma UMI, é possível rastrear a posição e orientação de um dispositivo, como será melhor detalhado ao longo da presente seção.

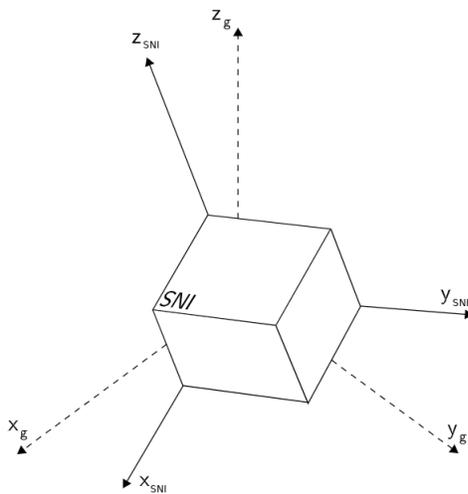


Figura 2.2: Distinção de {G} e de {SNI}.

Existem duas classificações que englobam praticamente todas as UMIs. Para entender essas duas classificações, é necessário distinguir o sistema de coordenadas globais, que daqui em diante será referido como  $\{G\}$ , do sistema de coordenadas do sistema de navegação, doravante denominado de  $\{SNI\}$ . A Figura 2.2 pode ajudar no melhor entendimento dos dois sistemas de coordenadas. Enquanto que o  $\{G\}$  é sempre fixo, o  $\{SNI\}$  varia de orientação a medida que o corpo do objeto rastreado se movimenta.

As duas classificações de UMIs são: UMIs de Plataforma Fixa e as chamadas UMIs *Strapdown*[28]. No caso das UMIs de plataforma fixa, os sensores são posicionados em uma plataforma que, com o auxílio de motores e de *gimbals*, os mantém sempre na mesma orientação do  $\{G\}$ , daí o nome da classificação. A Figura 2.3 apresenta uma ilustração de uma UMI de plataforma Fixa. Já no caso das UMIs do tipo *Strapdown*, tem-se que os sensores da UMI ficam fixamente acoplados no corpo do robô ou de qualquer outro objeto que se deseja rastrear a posição. Sendo assim, é natural que a orientação dos sensores varie em conjunto com o  $\{SNI\}$ . Ou seja, a diferença entre as duas classificações de UMI está no sistema de coordenadas usado como referência pelos sensores: A UMI de Plataforma Fixa faz leituras em relação ao  $\{G\}$  e a UMI *Strapdown* faz medidas referentes ao  $\{SNI\}$ . Apresentada essa distinção, a partir deste ponto, sempre que o presente trabalho mencionar uma UMI, o leitor deverá assumir que se trata de uma UMI do tipo *Strapdown*, que é um dos objetos de estudo desta pesquisa.

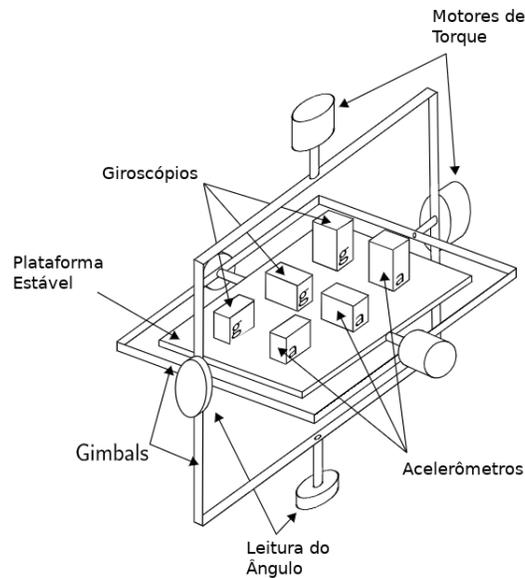


Figura 2.3: UMI de plataforma fixa.

As UMIs são amplamente utilizadas. Suas aplicações vão desde detecção de orientação de câmeras fotográficas e vão até a sistemas de orientação e navegação de aeronaves[30, 31, 28, 25].

O restante desta seção irá apresentar em detalhes os princípios de funcionamento dos giroscópios e acelerômetros normalmente presentes em UMIs, bem como detalhar a teoria por trás da Navegação Inercial, isto é, apresentar como pode-se estimar a posição e orientação de um dado objeto fazendo uso apenas das leituras de uma UMI acoplada ao corpo do objeto.

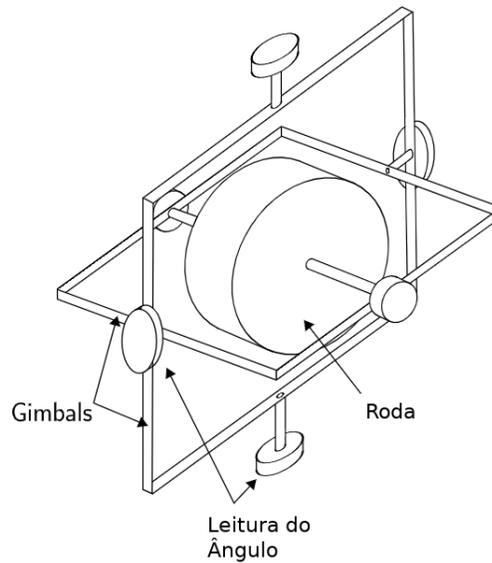


Figura 2.4: Um giroscópio mecânico convencional.

### 2.2.1 Giroscópio

Giroscópios são sensores de velocidade e posição angular bastante populares. O Giroscópio foi inventado e batizado em 1852 por Léon Foucault. Três das mais conhecidas classificações de giroscópios são: os mecânicos; os ópticos e os MEMS[25]. Note que os sensores que aqui estão descritos não compõem uma lista muito extensa. Se o leitor se interessar pelo tema, recomenda-se a leitura da referência [28].

Giroscópios Mecânicos são instrumentos compostos basicamente por uma roda giratória acoplada a dois gimbals que permitem que a roda possa rotacionar em todos os três eixos como apresentado na Figura 2.4. O seu funcionamento se baseia no princípio da conservação da quantidade de momento angular, que faz com que a roda giratória resista a variações em sua orientação. Portanto, quando um giroscópio mecânico é submetido a alguma movimentação angular, ou seja, a uma rotação, a roda giratória tenderá a manter a sua orientação global constante mediante alteração dos ângulos entre os gimbals que a sustentam. Dessa forma, pode-se medir a orientação do giroscópio através dos ângulos entre os gimbals. Importante observar que, no caso dos giroscópios mecânicos, o que se é medido é a orientação do dispositivo. O mesmo não acontece com outros tipos de giroscópios mais modernos, como será apresentado em seguida.

Giroscópios Ópticos como os Giroscópios de Fibra Óptica (FOG na sigla em inglês) fazem uso de propriedades da luz para medir a velocidade angular. Um FOG consiste em uma espiral de fibra óptica na qual são introduzidos dois feixes de luz, um em cada uma das direções possíveis. Caso o sensor esteja sendo submetido a uma movimentação angular, o feixe de luz que está percorrendo a fibra na direção da rotação irá percorrer um caminho mais longo até sair da fibra do que o feixe que está percorrendo a direção contrária a rotação como pode ser observado na Figura 2.5. A este fenômeno dá-se o nome de Efeito Sagnac.

Quando os dois feixes de luz saem da espiral de fibra óptica, eles são combinados. O desvio de fase provocado pelo Efeito Sagnac faz com que este novo feixe de luz originário da combinação dos dois feixes iniciais tenha a intensidade variando de acordo com a velocidade angular. Dessa forma, é possível medir, a

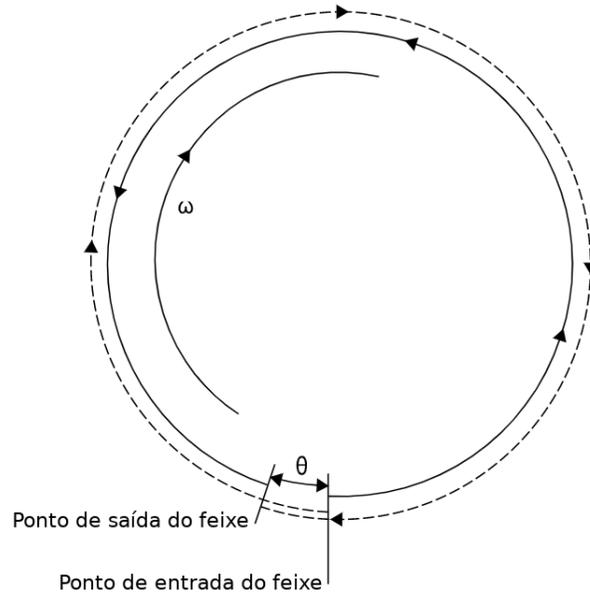


Figura 2.5: Efeito Sagnac. As linhas pontilhadas representam o caminho percorrido pelo feixe de luz viajando na direção da rotação enquanto que as linhas sólidas representam o caminho percorrido pelo feixe de luz que viaja na direção oposta a rotação.  $\theta$  é o ângulo que o giroscópio se movimentou enquanto os feixes de luz viajavam.

partir desse feixe combinado, a velocidade angular a que está sendo submetido o giroscópio. Ao contrário dos giroscópios mecânicos, os FOGs não apresentam nenhuma parte móvel. Sua precisão está atrelada ao comprimento da espiral de fibra óptica, quanto maior ela for, maior será a precisão.

Mesmo já estando disponíveis há muitos anos, os giroscópios ópticos e mecânicos ainda contam com um grande número de peças e dependem de uma precisão muito alta na sua fabricação. Sendo assim, os preços finais aos quais são submetidos no mercado ainda são elevados. Em contrapartida, os giroscópios MEMS podem ser fabricados em larga escala e com um número bem reduzido de peças. Isso confere ao sensor um nível de popularidade muito alto, pois, como é fácil de ser fabricado, seu custo final é bastante acessível[25]. Isso fez com que a maioria das UMIs adotassem este tipo de sensor para realizar as medições inerciais angulares, por essa razão, esse tipo de sensor será explorado em mais detalhes.

Seu funcionamento é relativamente simples e baseado no Efeito Coriolis[28]. Este efeito diz que, se uma massa  $m$  estiver com uma velocidade angular  $\omega$  e for submetido a uma velocidade linear  $v$ , será exercida sobre esta massa a força Coriolis:

$$F_C = -2m(\omega \times v). \quad (2.1)$$

Os Giroscópios MEMS apresentam em sua composição estrutural algum elemento vibratório para que seja possível medir o Efeito Coriolis. Este elemento vibratório pode apresentar diversos formatos, mas o mais simples de todos eles é apenas uma única massa que é submetida a uma vibração ao longo de um único eixo (eixo de acionamento), como ilustrado na Figura 2.6. Quando o giroscópio é submetido a uma rotação, outra vibração será induzida no eixo perpendicular (eixo sensitivo da Figura 2.6) em razão da Força Coriolis. A

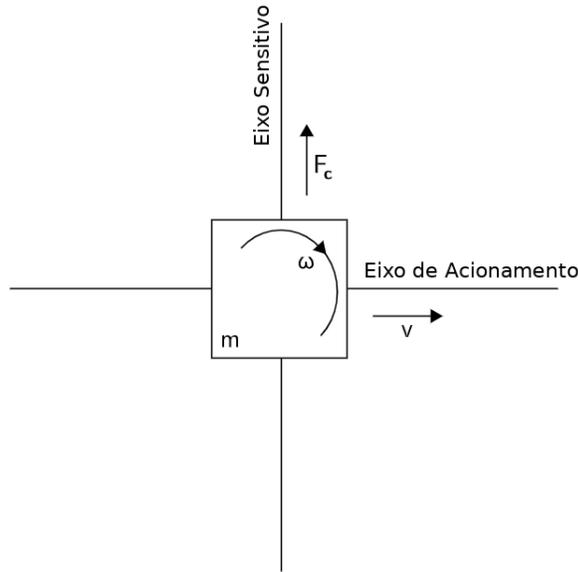


Figura 2.6: Giroscópio MEMS de massa vibratória.

velocidade angular a que o giroscópio está sendo submetido pode ser calculada a partir dos efeitos desta força.

Os giroscópios MEMS apresentam como desvantagem em relação aos seus equivalentes mecânicos e ópticos o fato de não apresentarem a mesma acurácia. Mas recentes avanços em sua fabricação vem reduzindo cada vez mais os seus erros de leitura o que os torna ainda mais atrativos do que já são pelo fato de serem muito mais baratos[29, 28].

Os erros que podem interferir em giroscópios MEMS são de diversas origens e possuem causas e efeitos para a integração (da velocidade para a posição angular) variados. Um dos tipos de erro mais comuns é o viés (mais comumente referido pelo termo em inglês “*bias*”) que perturba as leituras do sensor. Este viés pode ser entendido como sendo a média das leituras de velocidade angular do giroscópio quando o mesmo não está sendo submetido a rotação alguma[28].

Este tipo de erro é normalmente medido em  $^{\circ}/h$  (graus por hora). Assumindo um erro  $\epsilon$  de viés como o descrito, quando integrado, gera um erro na estimativa da posição angular que cresce linearmente com o tempo

$$\theta(t) = \epsilon \cdot t. \quad (2.2)$$

Este viés no sensor pode ser calculado fazendo a média das leituras do giroscópio por um longo período de tempo sem que o mesmo seja submetido a qualquer rotação. Uma vez que este viés for calculado, basta subtraí-lo de todas as medidas do sensor para obter leituras não enviesadas[28].

Outro tipo de erro que altera as medidas dos giroscópios MEMS é o causado por ruído termomecânico. Este ruído oscila em uma frequência maior que a taxa de leitura do sensor. Dessa forma, ocorre que as leituras do giroscópio são perturbadas por uma sequência de ruído branco, que nada mais é do que uma sequência de variáveis aleatórias descorrelacionadas e com média zero, onde cada variável é igualmente distribuída e apresenta variância  $\sigma^2$ .

Para descobrir o efeito que esse tipo de ruído causa no processo de integração dos dados coletados pelo sensor, faz-se a análise da integração deste erro isoladamente: Sendo  $N_i$  a  $i$  – ésima variável aleatória na sequência de ruído branco, tem-se que cada  $N_i$  é igualmente distribuída e apresenta média:

$$E(N_i) = E(N) = 0$$

e variância:

$$Var(N_i) = Var(N) = \sigma^2.$$

Pela definição de sequência de ruído branco, tem-se ainda que

$$Cov(N_i, N_j) = 0$$

para todo  $i \neq j$ .

O resultado da integração do ruído branco  $\epsilon(t)$  ao longo do período  $t = n \cdot \delta_t$  é:

$$\int_0^t \epsilon(\tau) d\tau = \delta_t \sum_{i=1}^n N_i \quad (2.3)$$

onde  $n$  é o número de leituras realizadas pelo sensor durante o período e  $\delta_t$  é o tempo que se passa entre duas leituras consecutivas. Utilizando-se da linearidade da operação de esperança, isto é, sabendo que  $E(aX + bY) = aE(X) + bE(Y)$  e que  $Var(aX + bY) = a^2Var(X) + b^2Var(Y) + 2abCov(X, Y)$  (onde  $a$  e  $b$  são constantes e  $X$  e  $Y$  são variáveis aleatórias), tem-se que:

$$E\left(\int_0^t \epsilon(\tau) d\tau\right) = \delta_t \cdot n \cdot E(N) = 0 \quad (2.4)$$

$$Var\left(\int_0^t \epsilon(\tau) d\tau\right) = \delta_t^2 \cdot n \cdot Var(N) \quad (2.5)$$

$$= \delta_t \cdot t \cdot \sigma^2 \quad (2.6)$$

Dessa forma, percebe-se que o ruído branco adicionado às leituras da velocidade angular do giroscópio causa um *Random Walk* no sinal integrado (posição angular) com um desvio padrão

$$\sigma_\theta(t) = \sigma \sqrt{\delta_t \cdot t} \quad (2.7)$$

que cresce proporcionalmente à raiz quadrada do tempo[29, 28]. Normalmente os fabricantes especificam este desvio padrão em  $^\circ/\sqrt{h}$ .

Ainda existem outros erros que podem interferir nas amostras obtidas por giroscópios MEMS. Mudanças na temperatura, tanto ambiente quanto do próprio sensor durante o funcionamento, podem alterar o viés (que é dado como constante) do sensor. Importante perceber que este tipo de alteração não é percebida no cálculo do viés, pois este é medido em situações controladas (e.g. temperatura ambiente constante).

Qualquer alteração no viés calculado do sensor vai culminar em um erro na posição angular que cresce linearmente com o tempo, como mostrado na equação 2.2. De maneira geral, a relação entre temperatura

Tabela 2.1: Resumo das fontes de Erro em Giroscópios MEMS

Tipo de Erro	Descrição	Resultado na Integração
Viés	Um viés constante $\epsilon$ nas leituras do sensor	Um erro linearmente crescente na estimação da posição angular: $\theta(t) = \epsilon \cdot t$
Ruído Branco	Ruído branco com desvio padrão $\sigma$	Um <i>Random Walk</i> no ângulo da posição angular, com desvio padrão crescendo proporcionalmente com a raiz quadrada do tempo: $\sigma_{\theta}(t) = \sigma \sqrt{\delta_t \cdot t}$
Variação na Temperatura	Um viés residual que é dependente da temperatura	Qualquer viés residual integrado causa um erro na medida da orientação que cresce linearmente com o tempo.

e variação do viés do sensor é não linear. Várias UMIs possuem um sensor de temperatura interno que é utilizado para tentar corrigir esta alteração no viés causada pela mudança de temperatura.

A tabela 2.1 apresenta um resumo dos erros aqui apresentados, que podem ser tidos como os mais influentes no caso dos giroscópios MEMS[29].

## 2.2.2 Acelerômetro

Acelerômetros são sensores utilizados para medir aceleração linear[25, 22]. Seu uso vem ficando cada vez mais frequente com a popularização de versões baratas e simples de fabricar. Juntamente com o giroscópio, ele forma a base da navegação inercial. Existem basicamente duas classificações para os acelerômetros: eles podem ser sensores mecânicos ou de estado sólido[29].

Os acelerômetros mecânicos apresentam tanto a montagem quanto o funcionamento bastante simples. Eles são compostos basicamente por uma massa, que tem sua posição rastreada, sustentada por molas (de coeficiente elástico conhecido). Quando uma aceleração incide sobre o sensor, a massa se desloca de maneira proporcional a força  $F$  que está atrelada a aceleração em questão. A segunda Lei de Newton é então empregada para que se possa calcular a aceleração que está agindo sobre o dispositivo. A Figura 2.7 apresenta um esquemático de um acelerômetro mecânico. Note que trata-se de uma ilustração bastante simplificada, mas que já fornece compreensão suficiente das estruturas de um acelerômetro desta classificação.

Já os acelerômetros de estado sólido apresentam diversos subgrupos, dentre os quais estão os chamados acelerômetros SAW (da sigla em inglês *para Surface Acoustic Wave*)[28], que apresentam um funcionamento interessante: São compostos por uma espécie de haste, ou viga, em balanço na qual ressoa uma onda acústica. Em uma das suas extremidades é acoplada uma massa que é livre para se movimentar, enquanto que a outra extremidade é fixamente presa a estrutura do acelerômetro. A Figura 2.8 apresenta um esquemático de um

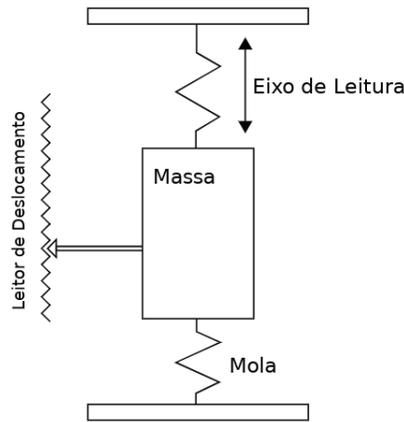


Figura 2.7: Esquemático simplificado de um acelerômetro mecânico.

sensor desta classificação. Se uma aceleração for imposta ao sensor, a massa fixada a viga fará com que a mesma envergue. Isso fará com que a frequência da onda acústica que ressoa na viga seja alterada de maneira proporcional a tensão aplicada à haste. A aceleração a que o sensor foi submetido pode ser reconstruída a partir desta mudança na frequência da onda.

Mas o que realmente tornou os acelerômetros tão populares foi o avanço nas técnicas de fabricação de Sistemas Microeletromecânicos (MEMS, da sigla em inglês), que permitiram que fossem projetados acelerômetros de baixo custo[25]. Os acelerômetros MEMS seguem os mesmos princípios dos acelerômetros já apresentados, isto é, podem ser mecânicos ou de estado sólido. Mas assim como os giroscópios MEMS, os acelerômetros microeletromecânicos apresentam uma série de fontes de erros em suas medias[28]. Esses erros serão integrados não uma (como no caso dos giroscópios que integram a velocidade uma vez para encontrar a posição angular), mas duas vezes durante o processo de cálculo da posição a partir da aceleração medida. A seguir será apresentada uma análise mais detalhada do efeito da integração desses erros, a exemplo de como foi feito na seção anterior com os ruídos presentes nas leituras dos giroscópios.

Assim como apresentado para os giroscópios, acelerômetros MEMS fazem leituras enviesadas, isto é, as amostras da aceleração coletadas pelo sensor apresentam um viés constante  $\epsilon$  somado a verdadeira aceleração que deveria ser lida. Ao se integrar duas vezes a aceleração a fim de se obter a posição do dispositivo, acaba por se integrar também  $\epsilon$ , que provoca um erro na estimativa da posição que cresce quadraticamente com o tempo. O erro acumulado na posição provocado por este viés é:

$$s(t) = \epsilon \cdot \frac{t^2}{2} \quad (2.8)$$

onde  $t$  representa o período de integração.

Assim como no caso do giroscópio, para se calcular o viés de leitura do acelerômetro, pode-se calcular a média das medidas de aceleração obtidas pelo sensor quando o mesmo não está sendo submetido a qualquer aceleração. O grande problema desta tarefa está na existência da gravidade, pois esta será lida como viés por este procedimento. Sendo assim, é necessário conhecer a orientação exata do sensor em relação à Terra para que seja possível distinguir a aceleração da gravidade do viés do acelerômetro[29].

As medidas do acelerômetro também sofrem perturbações causadas por uma sequência de ruído branco,

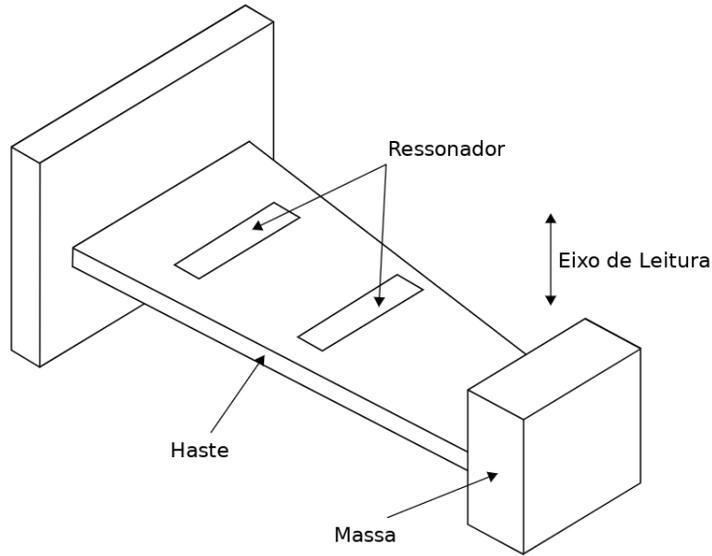


Figura 2.8: Esquemático de um acelerômetro do tipo SAW.

cuja origem está nas oscilações termomecânicas (a exemplo dos giroscópios). Na seção anterior, foi apresentado que, quando se integra esse tipo de ruído uma vez, o resultado é um *Random Walk* na saída do sinal integrado com desvio padrão crescente e proporcional a  $\sqrt{t}$ . Mas como o interesse é em descobrir a posição a partir da aceleração linear lida pelo sensor, é necessário integrar o ruído duas vezes para se conhecer os erros por ele causados nesta estimativa de posição. Assim, sendo  $N_i$  a  $i$ -ésima variável aleatória na sequência de ruído branco adicionada pelo ruído, tem-se que cada  $N_i$  é igualmente distribuída e apresenta média:

$$E(N_i) = E(N) = 0$$

e variância:

$$Var(N_i) = Var(N) = \sigma^2.$$

O resultado da dupla integração do ruído branco  $\epsilon(t)$  ao longo do período  $t = n \cdot \delta_t$  é:

$$\int_0^t \int_0^t \epsilon(\tau) d\tau d\tau = \delta_t \sum_{i=1}^n \delta_t \sum_{j=1}^i N_j \quad (2.9)$$

$$= \delta_t^2 \sum_{i=1}^n (n - i + 1) N_i \quad (2.10)$$

onde  $n$  é o número de amostras coletadas pelo acelerômetro durante o período de integração  $t$  e  $\delta_t$  é o tempo decorrido entre duas amostras consecutivas do sensor. Desta forma, a expectativa do erro na posição causado por este ruído é:

$$E\left(\int_0^t \int_0^t \epsilon(\tau) d\tau d\tau\right) = \delta_t^2 \sum_{i=1}^n (n-i+1) E(N_i) \quad (2.11)$$

$$= 0 \quad (2.12)$$

e a variância deste erro é:

$$Var\left(\int_0^t \int_0^t \epsilon(\tau) d\tau d\tau\right) = \delta_t^4 \sum_{i=1}^n (n-i+1)^2 Var(N_i) \quad (2.13)$$

$$= \frac{\delta_t^4 n(n+1)(2n+1)}{6} Var(N) \quad (2.14)$$

$$\approx \frac{1}{3} \delta_t \cdot t^3 \cdot \sigma^2 \quad (2.15)$$

onde a aproximação assume que  $\delta_t$  é pequeno (o que é bastante provável na realidade, uma vez que a frequência de amostragem dos sensores MEMS costuma ser bastante elevada). Este resultado mostra que a posição linear calculada a partir das leituras de um acelerômetro é sujeito a um erro *Random Walk* de segunda ordem com média zero e desvio padrão

$$\sigma_s(t) \approx \sigma \cdot \sqrt{t^3} \cdot \sqrt{\frac{\delta_t}{3}} \quad (2.16)$$

que cresce proporcional a  $\sqrt{t^3}$ .

Alterações na temperatura ambiente ou na temperatura de operação do acelerômetro MEMS também acarretam em flutuações no viés do sensor. Como mostrado na equação 2.8, qualquer viés não corrigido que interfira nas medidas da aceleração irão gerar um erro na estimativa da posição do dispositivo que cresce quadraticamente com o tempo.

Como pode-se perceber, os tipos de erros presentes nos acelerômetros foram os mesmos dos apresentados para o caso dos giroscópios, e assim como dito na seção anterior, estes podem ser tidos como os principais erros que interferem na estimação da posição a partir dos ruídos presentes nas leituras da aceleração linear capturadas[28]. Os tipos de erros e suas consequências aqui mostradas estão condensadas na Tabela 2.2.

## Paralelo com os sensores presentes na natureza:

Na natureza, os seres vivos desenvolveram sensores, de certa, forma análogos ao giroscópio e ao acelerômetro para conseguirem perceber a velocidade angular e a aceleração linear a que estão submetidos[25, 24]. No caso dos vertebrados, esta percepção se dá no sistema vestibular, que está abrigado no ouvido interno dos indivíduos. Este sistema é composto basicamente por três estruturas principais: a cóclea, o vestíbulo e os canais semicirculares.

O vestíbulo abriga dois órgãos sensoriais, o utrículo e o sáculo, que conseguem perceber forças inerciais e gravitacionais fornecendo ao indivíduo informações a respeito da posição angular e movimentação linear da sua cabeça. Já os canais semicirculares são estruturas que detectam a velocidade angular da cabeça dos vertebrados. Devido a complexa estrutura tubular destas estruturas, o conjunto delas é normalmente conhecido como labirinto[25]. A Figura 2.9 apresenta o desenho anatômico do ouvido interno humano, onde pode-se observar todas as estruturas antes mencionadas.

Tabela 2.2: Resumo das fontes de Erro em acelerômetros MEMS

Tipo de Erro	Descrição	Resultado na Integração
Viés	Um viés constante $\epsilon$ nas leituras do sensor	Um erro quadraticamente crescente na estimativa da posição linear: $s(t) = \epsilon \cdot \frac{t^2}{2}$
Ruído Branco	Ruído branco com desvio padrão $\sigma$	Um <i>Random Walk</i> de segunda ordem na posição linear, com desvio padrão crescendo proporcionalmente com a raiz quadrada do cubo do tempo: $\sigma_s(t) \approx \sigma \cdot \sqrt{t^3} \cdot \sqrt{\frac{\delta t}{3}}$
Varição na Temperatura	Um viés residual que é dependente da temperatura	Qualquer viés residual duplamente integrado causa um erro na medida da posição que cresce quadraticamente com o tempo.

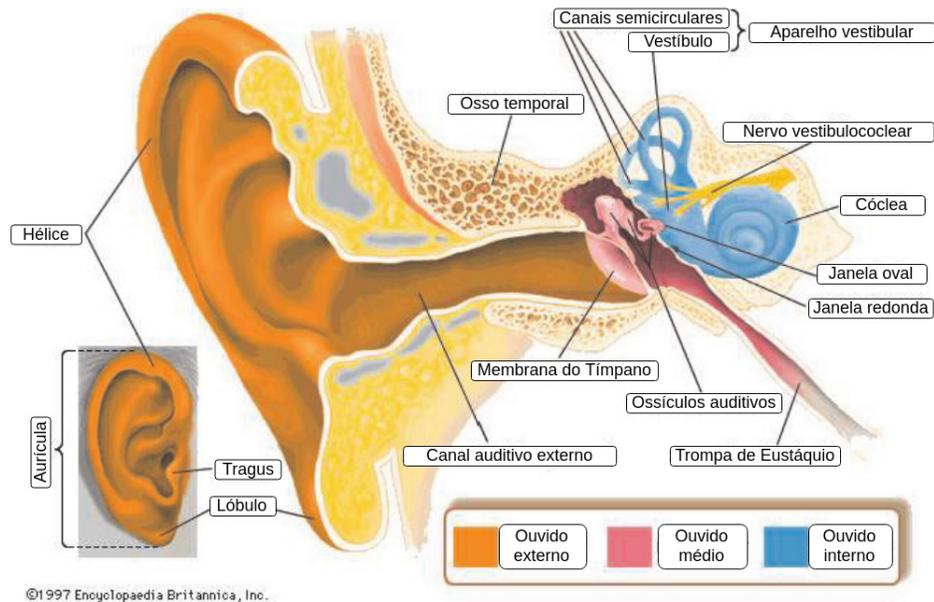


Figura 2.9: Ouvido de um ser humano.

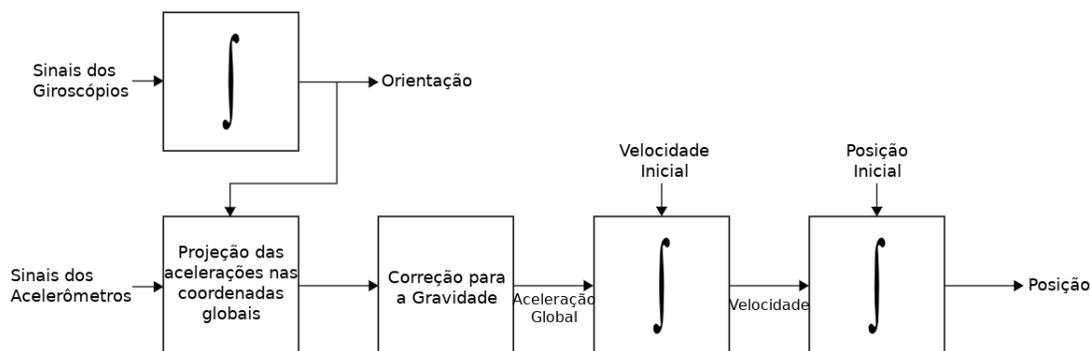


Figura 2.10: O algoritmo da Navegação Inercial.

## 2.2.3 Algoritmo da Navegação Inercial

Até agora foram apresentados os princípios por trás do funcionamento dos sensores presentes em uma UMI, que é o dispositivo utilizado para se realizar a Navegação Inercial. A partir de agora será apresentado o algoritmo utilizado para, a partir das leituras do acelerômetro e do giroscópio de uma UMI, rastrear a posição e orientação de um dispositivo ao qual a unidade de medição inercial está fixamente acoplada. Ao longo desta seção, serão utilizados os subscritos  $s$ , para se referir a medidas no {SNI}, e  $g$ , para indicar que a grandeza se refere ao {G}.

A Figura 2.10 apresenta o esquema do algoritmo da Navegação Inercial. No esquemático é possível perceber todas as etapas do processamento que o sinal da UMI passa para que seja possível estimar a posição e orientação do objeto sendo rastreado. Em resumo, tem-se que a velocidade angular lida pelo giroscópio é integrada para que seja possível estimar a orientação. Esta orientação é usada para projetar as acelerações no {G} (já que o sensor mede as acelerações em relação ao {SNI}). Em seguida, é desconsiderada a aceleração da gravidade, para que então o sinal corrigido seja integrado duas vezes a fim de que se possa obter a posição do objeto.

### 2.2.3.1 Rastreado a Orientação

#### Teoria

A orientação, ou atitude, de um Sistema de Navegação Inercial relativa ao sistema de referências globais {G} é obtido a partir do sinal de velocidade angular

$$\omega_s(t) = \begin{bmatrix} \omega_{sx}(t) \\ \omega_{sy}(t) \\ \omega_{sz}(t) \end{bmatrix}$$

lido pelo giroscópio. Existem várias formas de se representar a orientação do SNI, algumas das maneiras mais comuns de se realizar esta representação são: ângulos de Euler, quaternions unitários e matrizes de rotação[32, 33, 34]. Esta última foi a escolhida para apresentar a derivação do algoritmo para se rastrear a atitude do SNI.

Mas derivações utilizando de outras representações podem ser encontradas em [29].

A matriz de rotação  $\mathbf{R}$  é uma matriz  $3 \times 3$  que pode ser utilizada para “converter” um vetor  $\mathbf{v}_s$ , representado no {SNI} para seu equivalente  $\mathbf{v}_g$  no {G}. Matematicamente:

$$\mathbf{v}_g = \mathbf{R}\mathbf{v}_s \quad (2.17)$$

a transformação inversa é dada por

$$\mathbf{v}_s = \mathbf{R}^T \mathbf{v}_g \quad (2.18)$$

uma vez que a matriz inversa de uma matriz de rotação é dada pela sua transposta[32].

Para que seja alcançado o objetivo de se rastrear a orientação do SNI, é necessário conseguir rastrear  $\mathbf{R}$  ao longo do tempo. Se a atitude no instante de tempo  $t$  é dada por  $\mathbf{R}(t)$ , então a taxa de mudança de  $\mathbf{R}$  no instante  $t$  é dada por:

$$\dot{\mathbf{R}}(t) = \lim_{\delta_t \rightarrow 0} \frac{\mathbf{R}(t + \delta_t) - \mathbf{R}(t)}{\delta_t} \quad (2.19)$$

onde pode  $\mathbf{R}(t + \delta_t)$  pode ser reescrito como sendo o produto de duas matrizes

$$\mathbf{R}(t + \delta_t) = \mathbf{R}(t)\mathbf{A}(t) \quad (2.20)$$

onde  $\mathbf{A}(t)$  é a matriz de rotação que relaciona o {SNI} no instante  $t$  com o {SNI} do instante  $t + \delta_t$ . Se  $\delta_{\theta_x}$ ,  $\delta_{\theta_y}$  e  $\delta_{\theta_z}$  são as pequenas rotações que acometeram o {SNI} entre os instantes de tempo  $t$  e  $t + \delta_t$  sobre seus eixos x, y e z, respectivamente, então, pela aproximação de pequenos ângulos, pode-se reescrever  $\mathbf{A}(t)$  como:

$$\mathbf{A}(t) = \mathbf{I} + \delta\mathbf{\Psi} \quad (2.21)$$

onde

$$\delta\mathbf{\Psi} = \begin{pmatrix} 0 & -\delta_{\theta_z} & \delta_{\theta_y} \\ \delta_{\theta_z} & 0 & -\delta_{\theta_x} \\ -\delta_{\theta_y} & \delta_{\theta_x} & 0 \end{pmatrix}. \quad (2.22)$$

Assim, por substituição:

$$\dot{\mathbf{R}}(t) = \lim_{\delta_t \rightarrow 0} \frac{\mathbf{R}(t + \delta_t) - \mathbf{R}(t)}{\delta_t} \quad (2.23)$$

$$= \lim_{\delta_t \rightarrow 0} \frac{\mathbf{R}(t)\mathbf{A}(t) - \mathbf{R}(t)}{\delta_t} \quad (2.24)$$

$$= \lim_{\delta_t \rightarrow 0} \frac{\mathbf{R}(t)(\mathbf{I} + \delta\mathbf{\Psi}) - \mathbf{R}(t)}{\delta_t} \quad (2.25)$$

$$= \mathbf{R}(t) \lim_{\delta_t \rightarrow 0} \frac{\delta\mathbf{\Psi}}{\delta_t} \quad (2.26)$$

No limite  $\delta_t \rightarrow 0$ , a aproximação de pequenos ângulos é válida, e

$$\lim_{\delta_t \rightarrow 0} \frac{\delta \Psi}{\delta_t} = \Omega(t) \quad (2.27)$$

onde

$$\Omega(t) = \begin{pmatrix} 0 & -\omega_{sz}(t) & \omega_{sy}(t) \\ \omega_{sz}(t) & 0 & -\omega_{sx}(t) \\ -\omega_{sy}(t) & \omega_{sx}(t) & 0 \end{pmatrix} \quad (2.28)$$

é a forma matricial antissimétrica do vetor de velocidade angular  $\omega_s(t)$ . Portanto, para que se rastreie a orientação, deve-se resolver a equação diferencial:

$$\dot{\mathbf{R}}(t) = \mathbf{R}(t)\Omega(t) \quad (2.29)$$

que tem a solução

$$\mathbf{R}(t) = \mathbf{R}(0) \cdot \exp\left(\int_0^t \Omega(t) dt\right) \quad (2.30)$$

onde  $\mathbf{R}(0)$  é a atitude inicial do SNI[29].

## Implementação

O giroscópio presente na UMI não fornece uma leitura contínua de  $\omega_s(t)$ , na verdade o sensor apresenta amostras obtidas a uma certa frequência. Dessa forma, é necessário utilizar um esquema numérico para integrar o sinal amostrado. O esquema a ser escolhido depende da aplicação. Para curtos intervalos de tempo ou para aplicações que não necessitam de tanta precisão assim do sensor, utilizar a regra dos retângulos já é suficiente[29, 28].

Sendo o tempo entre amostras sucessivas da velocidade angular denotado por  $\delta_t$ , para um único período  $[t, t + \delta_t]$ , a solução para a equação 2.29 pode ser escrita como:

$$\mathbf{R}(t + \delta_t) = \mathbf{R}(t) \cdot \exp\left(\int_t^{t+\delta_t} \Omega(t) dt\right) \quad (2.31)$$

usando a regra dos retângulos, pode-se escrever:

$$\int_t^{t+\delta_t} \Omega(t) dt = \mathbf{B} \quad (2.32)$$

onde

$$\mathbf{B} = \begin{pmatrix} 0 & -\omega_{sz}\delta_t & \omega_{sy}\delta_t \\ \omega_{sz}\delta_t & 0 & -\omega_{sx}\delta_t \\ -\omega_{sy}\delta_t & \omega_{sx}\delta_t & 0 \end{pmatrix} \quad (2.33)$$

e o vetor

$$\boldsymbol{\omega}_s = \begin{bmatrix} \omega_{sx} \\ \omega_{sy} \\ \omega_{sz} \end{bmatrix}$$

é a amostra da velocidade angular correspondente ao período analisado. Sendo  $\rho = |\boldsymbol{\omega}_s \delta_t|$ , substituindo a Equação 2.32 na Equação 2.31 e em seguida realizando a expansão de Taylor da operação exponencial, tem-se:

$$\mathbf{R}(t + \delta_t) = \mathbf{R}(t) \cdot \left( \mathbf{I} + \mathbf{B} + \frac{\mathbf{B}^2}{2!} + \frac{\mathbf{B}^3}{3!} + \frac{\mathbf{B}^4}{4!} + \dots \right) \quad (2.34)$$

$$= \mathbf{R}(t) \cdot \left( \mathbf{I} + \mathbf{B} + \frac{\mathbf{B}^2}{2!} - \frac{\rho^2 \mathbf{B}}{3!} - \frac{\rho^2 \mathbf{B}^2}{4!} + \dots \right) \quad (2.35)$$

$$= \mathbf{R}(t) \cdot \left( \mathbf{I} + \left( 1 - \frac{\rho^2}{3!} + \frac{\rho^4}{5!} + \dots \right) \mathbf{B} + \left( \frac{1}{2!} - \frac{\rho^2}{4!} + \frac{\rho^4}{6!} + \dots \right) \mathbf{B}^2 \right) \quad (2.36)$$

$$= \mathbf{R}(t) \cdot \left( \mathbf{I} + \frac{\sin \rho}{\rho} \mathbf{B} + \frac{1 - \cos \rho}{\rho} \mathbf{B}^2 \right) \quad (2.37)$$

que é a fórmula para se atualizar a estimativa da orientação  $\mathbf{R}$  sempre que uma nova medida do giroscópio estiver disponível.

### 2.2.3.2 Rastreamento a posição

#### Teoria

Seguindo a lógica apresentada no esquemático da Figura 2.10, para que seja possível rastrear a posição de um SNI, o vetor de aceleração

$$\mathbf{a}_s(t) = \begin{bmatrix} a_{sx}(t) \\ a_{sy}(t) \\ a_{sz}(t) \end{bmatrix}$$

obtido pelo acelerômetro deve ser projetado no {G}:

$$\mathbf{a}_g = \mathbf{R}(t) \mathbf{a}_s(t). \quad (2.38)$$

A aceleração da gravidade deve então ser subtraída e a aceleração restante deve ser integrada uma vez para que se obtenha a velocidade linear e mais uma vez para que se obtenha o deslocamento:

$$\mathbf{v}_g(t) = \mathbf{v}_g(0) + \int_0^t \mathbf{a}_g(t) - \mathbf{g}_g dt \quad (2.39)$$

$$\mathbf{s}_g(t) = \mathbf{s}_g(0) + \int_0^t \mathbf{v}_g(t) dt \quad (2.40)$$

onde  $\mathbf{v}_g(t)$  é a velocidade inicial do dispositivo,  $\mathbf{s}_g(0)$  é a posição inicial do mesmo e  $\mathbf{g}_g$  é a aceleração da gravidade em relação ao {G}.

## Implementação

Da mesma forma que mostrado no caso do uso do giroscópio para medir a velocidade angular, o acelerômetro não fornece uma medida do sinal constante de  $\mathbf{a}_s(t)$ , sendo assim, é necessário utilizar de técnicas de integração numérica para que seja possível recuperar o deslocamento do dispositivo no período em questão.

Utilizando a regra dos retângulos para se realizar as integrações, a fórmula de atualização da posição pode ser escrita como:

$$\mathbf{v}_g(t + \delta_t) = \mathbf{v}_g(t) + \delta_t \cdot (\mathbf{a}_g(t + \delta_t) - \mathbf{g}_g) \quad (2.41)$$

$$\mathbf{s}_g(t + \delta_t) = \mathbf{s}_g(t) + \delta_t \cdot \mathbf{v}_g(t + \delta_t) \quad (2.42)$$

### 2.2.3.3 Propagação de erros

Os erros que surgem das leituras realizadas pelo acelerômetro são propagadas pelas duas integrações e resultam em um erro direto na posição estimada, como já mostrado na seção 2.2.2. O que agora também se torna evidente é que os erros que surgem na leitura da velocidade angular também resultam em erros que incidem diretamente sobre a posição, uma vez que a matriz de rotação  $\mathbf{R}$ , que é calculada a partir das amostras do giroscópio, é utilizada para realizar a projeção das leituras do acelerômetro no  $\{G\}$ [29]. Um erro na estimação da orientação fatalmente desencadeará em um erro na projeção da aceleração e, conseqüentemente, em um erro na estimação da posição, seja porque as integrações irão levar em conta acelerações nas direções erradas ou seja porque se torna impossível de remover a ação da gravidade com a precisão adequada.

Conforme evidenciado nas Tabelas 2.1 e 2.2, os erros que surgem nos sensores da UMI crescem conforme o tempo de operação do dispositivo avança[28]. E como é muito difícil de se trabalhar em uma situação em que a UMI não é sujeita a erros, é praticamente impossível contar apenas com a Navegação Inercial para se estimar a posição de um robô móvel.

## 2.3 ODOMETRIA VISUAL

Odometria visual é o nome que se dá a técnica de estimação da trajetória percorrida por um agente (veículo, pessoa, robô móvel...) a partir da sequência de imagens capturadas por uma ou mais câmeras fixamente acopladas ao seu corpo[15]. As aplicações para esse tipo de técnica vão desde robótica móvel até aplicações de realidade aumentada. O termo Odometria Visual foi primeiramente empregado em [35], e foi escolhido devido a sua semelhança com a odometria que, a partir de integrações das voltas das rodas, consegue estimar qual foi a trajetória percorrida por um veículo[15]. De maneira semelhante, a Odometria Visual também estima a trajetória do agente de maneira incremental, a partir de mudanças nas imagens que são geradas devido a movimentação da câmera embarcada.

Para que a Odometria Visual funcione corretamente, é necessário que a câmera esteja em um ambiente com boa iluminação e com boa textura, para que seja possível extrair a movimentação entre as imagens[15, 16, 35]. Além disso, a câmera deve capturar imagens com uma frequência suficiente para que exista superposição de cenas em imagens consecutivas.

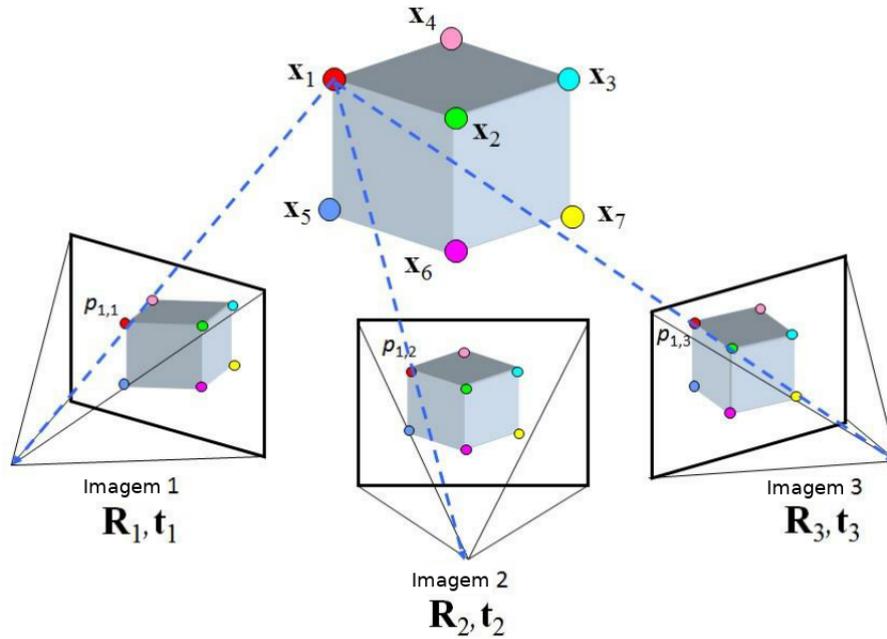


Figura 2.11: Representação de uma cena de múltiplas vistas, com câmeras em posições e orientações diferentes. A presença de pontos do objeto em imagens diferentes é utilizada na SFM para se estimar a posição 3D do ponto, bem como as posições e orientações das câmeras que capturaram as imagens.

A Odometria Visual pode ser entendida como um caso particular da área de estudo da visão computacional chamada de SFM (do termo em inglês *Structure From Motion*)[18], onde, a partir de múltiplas imagens de uma mesma cena, capturadas com uma ou mais câmeras de pontos de vista diferentes, estima-se a posição no espaço onde as câmeras estavam no momento que as imagens foram capturadas bem como as posições dos pontos tridimensionais que são projetados em pixels nas imagens. Técnicas de SFM são muito utilizadas para se realizar o mapeamento tridimensional de ambientes, ou mesmo de objetos mas, de maneira geral, pouco importa a ordem ou instante de tempo que as imagens foram obtidas. O que difere a Odometria Visual das demais técnicas de SFM é justamente o fato de se estar interessado em estimar as posições da câmera ao longo do tempo, estimando assim, sua trajetória a medida que ela captura imagens[18, 35, 15]. A seção 2.3.1 apresenta em mais detalhes a definição do problema da Odometria Visual.

### 2.3.1 Formulação do Problema

Um agente se movimenta em um ambiente enquanto uma (ou mais) câmeras, fixamente acopladas ao seu corpo, capturam imagens a intervalos discretos de tempo  $k$ . Essas imagens são denotadas por

$$I_{0:n} = \{I_0, I_1, \dots, I_n\}$$

no caso de sistemas que possuem apenas uma única câmera (monoculares). Para sistemas stereo (com duas câmeras) adiciona-se um subscrito para indicar de qual câmera veio a imagem capturada:  $I_{e,0:n} = \{I_{e,0}, I_{e,1}, \dots, I_{e,n}\}$  para as imagens vindas da câmera da esquerda e  $I_{d,0:n} = \{I_{d,0}, I_{d,1}, \dots, I_{d,n}\}$  para as imagens vindas da câmera da direita. Para casos com mais câmeras, aplica-se lógica semelhante.

Buscando simplificar, adota-se que o sistema de coordenadas da câmera, que também é representado por {CAM}, é o mesmo do agente em movimento. No caso de um sistema stereo, o sistema de coordenadas do agente é escolhido como o mesmo do sistema de coordenadas da câmera da esquerda.

A posição e orientação no espaço da câmera em dois instantes adjacentes de tempo  $k - 1$  e  $k$ , são relacionadas pela transformação de corpo rígido  $T_{k,k-1} \in \mathbb{R}^{4 \times 4}$  de modo que:

$$T_{k,k-1} = \begin{bmatrix} \mathbf{R}_{k,k-1} & \mathbf{t}_{k,k-1} \\ 0 & 1 \end{bmatrix} \quad (2.43)$$

onde  $\mathbf{R}_{k,k-1} \in SO(3)$  é a matriz de rotação que representa a rotação ocorrida entre os instantes de tempo em questão, da mesma forma,  $\mathbf{t}_{k,k-1} \in \mathbb{R}^{3 \times 1}$  é o vetor de translação ocorrido entre os mesmos instantes de tempo. O conjunto

$$T_{1:n} = \{T_{1,0}, \dots, T_{k,k-1}\}$$

contém todas as movimentações consecutivas realizadas pela câmera entre cada dois instantes de tempo. O conjunto de poses da câmera:

$$C_{0:n} = \{C_0, C_1, \dots, C_n\}$$

contém as transformações aplicadas à câmera a partir da pose inicial no instante de tempo  $k = 0$ , ou seja, apresenta a trajetória percorrida pela câmera. Dessa forma, a pose atual da câmera pode ser calculada concatenando todas as transformações  $T_{k,k-1}$  ( $k = 1, \dots, n$ ) de modo que:

$$C_n = C_{n-1}T_{n,n-1} \quad (2.44)$$

com  $C_0$  sendo a pose inicial da câmera no instante de tempo  $k = 0$ .

O objetivo da Odometria Visual é descobrir as transformações  $T_{k,k-1}$  a partir, unicamente, das imagens  $I_{k-1}$  e  $I_k$  e, com este resultado, concatenar as transformações encontradas para que se consiga recuperar a trajetória  $C_{0:n}$  da câmera. Ou seja, fica evidente que a Odometria Visual consegue estimar a trajetória incrementalmente, pose depois de pose [35, 15, 36, 4, 17, 16].

## Categorias de Odometria Visual

Como dito anteriormente, as técnicas de Odometria Visual podem ser classificadas quanto ao número de câmeras que são utilizadas para se reconstruir a trajetória percorrida pelo agente. Quando se utiliza apenas uma única câmera, tem-se a chamada Odometria Visual Monocular, e Odometria Visual Stereo é o nome que se dá para quando a mesma é realizada com duas câmeras. No que diz respeito a quantidade de câmeras utilizadas, essas são as duas principais classificações presentes na literatura [15].

Mas ainda é possível realizar outras classificações em relação às técnicas de Odometria Visual que vão além da quantidade de sensores visuais utilizados [15, 16, 35]. Por exemplo, para que o sistema seja capaz de estimar a movimentação ocorrida entre um *frame* e outro, ele precisa realizar uma especie de triangulação entre pixels (ou grupo de pixels), que estão presentes em duas imagens consecutivas, e seus pontos reais que geram as projeções nas imagens. Essa triangulação pode ser realizada a partir de diferentes representações dos pontos

em questão: No caso de sistemas stereo, tem-se a possibilidade de, a partir da disparidade e de informações a respeito da montagem das câmeras, recuperar a profundidade dos pontos projetados nas imagens, isto é, existe como se encontrar todas as coordenadas tridimensionais dos pontos projetados. Sendo assim, existem sistemas de Odometria Visual Stereo que realizam essas triangulações tratando os pontos projetados como pontos no espaço 3D e recuperando a movimentação das câmeras como a solução de um problema de alinhamento de pontos 3D para 3D. Ainda no caso de sistemas Stereo, a partir de [35], foram apresentadas novas técnicas que não mais necessitavam da representação tridimensional dos pontos projetados nas imagens para que fosse possível recuperar a movimentação das câmeras, isto é, a estimação do deslocamento das câmeras também pode ser a solução de um problema de alinhamento de pontos 3D para 2D. Mesmo com a possibilidade de se obter a informação 3D a respeito dos pontos que são projetados nas imagens Stereo, ainda existem técnicas que, fazendo uso do tensor quadrifocal para, a partir de uma solução de alinhamento de pontos 2D para 2D, recuperar a movimentação realizada pelas câmeras[36, 4, 17, 16, 37, 38, 39, 15].

Já no caso da Odometria Visual Monocular, não se consegue recuperar informações tridimensionais dos pontos projetados na imagem capturada pela câmera[18, 19, 15]. Isso porque é impossível descobrir a escala a partir de imagens obtidas com câmeras monoculares[40, 41]. Sendo assim, a solução encontrada para a estimação da movimentação realizada entre duas capturas por uma única câmera se dá a partir de uma solução de alinhamento de pontos 2D para 2D e com o agravante de se desconhecer a escala em que se deu a movimentação estimada. Sistemas monoculares podem ainda ser classificados com base no tipo de câmera que utilizam, sendo dois dos mais comuns os de câmera perspectiva e os de câmera omnidirecional[15].

Por fim, os sistemas de Odometria Visual podem ser categorizados a partir do tipo de informação que buscam nas imagens capturadas. Existem os sistemas baseados em *features* [35, 15], isto é, sistemas que procuram pontos salientes nas imagens que podem ser encontrados em outras imagens subsequentes, existem também as técnicas que se utilizam da própria intensidade dos pixels das imagens capturadas para se extrair informações de movimentação [4, 17], chamadas de técnicas baseadas na aparência (*appearance-based*) e também existem os sistemas que fazem uso das duas informações, tanto pontos salientes quanto intensidade dos pixels das imagens e são conhecidas como técnicas híbridas[36, 15].

No presente trabalho, daqui em diante, sempre que se falar de Odometria Visual, deve-se pensar no caso monocular, de câmera perspectiva e baseado em *features*. As seções seguintes entrarão em mais detalhes a cerca do funcionamento desta categoria de Odometria Visual.

### 2.3.2 Teoria

Para se entender o funcionamento da Odometria Visual, é necessário conhecer o modelo matemático da câmera. O modelo para câmera perspectiva mais comumente utilizado na literatura é baseado na projeção *pinhole* [19, 18, 22], onde a imagem é formada pela interseção dos raios de luz vindos da cena que passam pelo centro de projeção da câmera com o plano focal.

Um modelo de câmera serve basicamente para relacionar um ponto tridimensional  $\mathbf{P} = \begin{bmatrix} X & Y & Z \end{bmatrix}^T$  representado no {CAM} com seu correspondente projetado no plano da imagem  $\mathbf{p} = \begin{bmatrix} u & v \end{bmatrix}^T$ .

Uma formulação bastante simplificada deste modelo pode ser dada por:

$$u = f \frac{X}{Z} + u_0 \quad (2.45)$$

$$v = f \frac{Y}{Z} + v_0 \quad (2.46)$$

onde  $u$  e  $v$  são as coordenadas do pixel na imagem resultado da projeção do ponto  $\mathbf{P}$ ,  $(u_0, v_0)$  corresponde às coordenadas do centro da projeção na imagem e  $f$  é a distância focal da câmera. Esse modelo pode ser expresso, em coordenadas homogêneas e a menos de um fator de escala  $w$ , em notação matricial na forma:

$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 & 0 \\ 0 & f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \quad (2.47)$$

Um maior detalhamento a respeito do modelo de câmera utilizado no desenvolvimento deste trabalho será apresentado na seção 2.5. Mas este modelo descrito pela Equação 2.47, apesar de ser bastante simplificado, é suficiente para mostrar as consequências de uma movimentação da câmera nos pixels projetados na imagem como será feito a seguir.

Suponha que a câmera esteja se movendo com velocidade angular

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

e com velocidade linear

$$\mathbf{V} = \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix}$$

ambos em relação ao  $\{G\}$ . Sendo  $\mathbf{P}$  um ponto tridimensional representado no  $\{CAM\}$ , sua velocidade representada no mesmo sistema de coordenadas pode ser expressa por:

$$\dot{\mathbf{P}} = \boldsymbol{\omega} \times \mathbf{P} + \mathbf{V}. \quad (2.48)$$

Substituindo as Equações 2.45 e 2.46 em 2.48, pode-se expressar as derivadas das coordenadas de  $\mathbf{P}$  como funções das coordenadas  $u$  e  $v$  de sua projeção no plano da imagem como:

$$\dot{X} = Z\omega_y - \frac{vZ}{f}\omega_z + V_x \quad (2.49)$$

$$\dot{Y} = \frac{uZ}{f}\omega_z - Z\omega_x + V_y \quad (2.50)$$

$$\dot{Z} = \frac{Z}{f}(v\omega_x - u\omega_y) + V_z. \quad (2.51)$$

Fazendo uso da regra do quociente para se calcular as derivadas de  $u$  e  $v$ , tem-se que:

$$\dot{u} = f \frac{Z\dot{X} - X\dot{Z}}{Z^2} \quad (2.52)$$

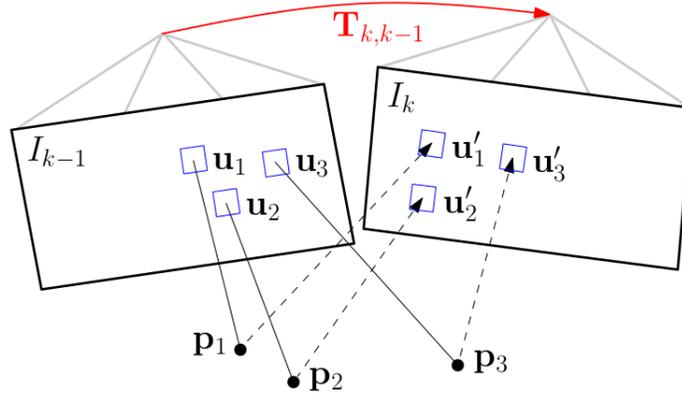


Figura 2.12: Deslocamento de pixels correspondentes em dois *frames* consecutivos devido a movimentação da câmera.

$$= \frac{f}{Z^2} \left\{ Z \left[ Z\omega_y - \frac{vZ}{f}\omega_z + V_x \right] - \frac{uZ}{f} \left[ \frac{Z}{f} (v\omega_x - u\omega_y) + V_z \right] \right\} \quad (2.53)$$

$$= \frac{f}{Z} V_x - \frac{u}{Z} V_z - \frac{uv}{f} \omega_x + \frac{f^2 + u^2}{f} \omega_y - v\omega_z \quad (2.54)$$

e de maneira análoga, tem-se que:

$$\dot{v} = \frac{f}{Z} V_x - \frac{v}{Z} V_z - \frac{f^2 + v^2}{f} \omega_x + \frac{uv}{f} \omega_y + u\omega_z. \quad (2.55)$$

Reescrevendo as equações 2.54 e 2.55 em formato matricial, obtém-se:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} \frac{f}{Z} & 0 & -\frac{u}{Z} & -\frac{uv}{f} & \frac{f^2 + u^2}{f} & -v \\ 0 & \frac{f}{Z} & -\frac{v}{Z} & -\frac{f^2 + v^2}{f} & \frac{uv}{f} & u \end{bmatrix} \times \begin{bmatrix} V_x \\ V_y \\ V_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (2.56)$$

onde fica claro que uma movimentação na câmera entre capturas de imagens acarreta em um deslocamento de pixels[25]. A Figura 2.12 mostra de maneira visual este deslocamento. Esta relação entre deslocamento dos pixels entre dois *frames* consecutivos e a movimentação espacial da câmera é o que torna a Odometria Visual possível.

Como pode-se perceber na Equação 2.56, ter o conhecimento dos elementos do modelo matemático da câmera é crucial (no caso do modelo simplificado apresentado na Equação 2.47, é necessário se conhecer a distância focal  $f$  e o centro de projeção da câmera  $(u_0, v_0)$ ). O processo no qual se busca conhecer, ou estimar, os valores destes elementos é conhecido como Calibração da Câmera[20, 18, 19] e será apresentado na seção 2.5. A seguir, será detalhada a implementação da Odometria Visual e ficará mais evidente essa dependência destes parâmetros.

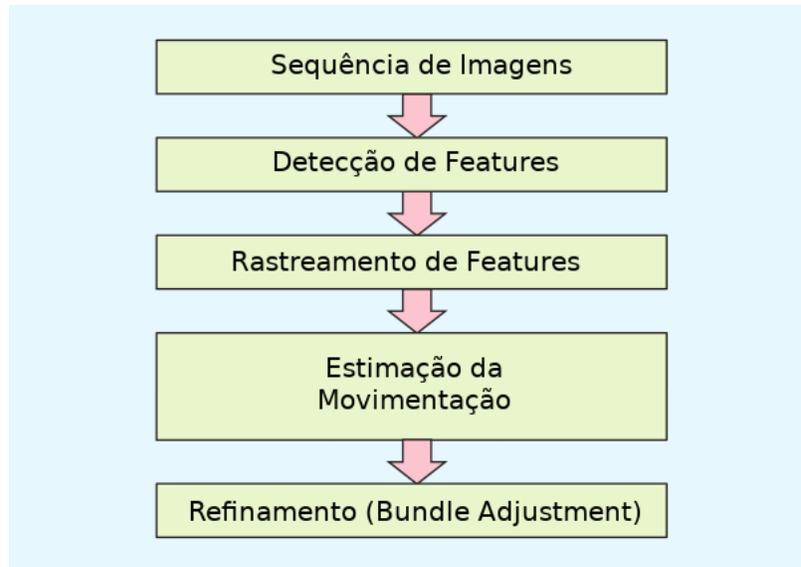


Figura 2.13: Diagrama de blocos mostrando os principais componentes de um sistema de Odometria Visual.

### 2.3.3 Implementação

Na Figura 2.13 é apresentado um diagrama de blocos no qual são mostradas as principais etapas da Odometria Visual. Como pode-se perceber, o diagrama começa pela sequência de imagens que abastece o sistema. O objetivo principal é que a cada duas imagens consecutivas seja estimada a movimentação a qual a câmera foi submetida entre as capturas das imagens.

Retomando ao diagrama, depois de ser abastecido com a sequência de dois *frames* consecutivos, o sistema procura por pontos salientes no primeiro destes *frames*. Isso se dá com o auxílio de técnicas de detecção de pontos que sejam fáceis de serem identificados e reconhecidos (como cantos, bordas e quinas presentes nas imagens). Algumas das técnicas de detecção de *features* mais utilizadas são: FAST, SIFT, SURF e ORB [42, 43, 44, 45]. Uma lista das principais características e detalhamento do funcionamento de cada uma destas e de outras técnicas pode ser encontrada em [15] e [16]. Mas o importante é ressaltar que, ao final desta etapa, o sistema de Odometria Visual identifica pontos de interesse no primeiro *frame*.

Na próxima etapa, ocorre a procura das *features* encontradas no primeiro *frame* na segunda imagem. Isto é feito por técnicas de rastreamento, como o rastreador Lucas Kanade[46], só que, mais uma vez, não é de interesse do presente trabalho apresentar esta ou outras técnicas de rastreamento, recomenda-se que o leitor busque mais informações em [47]. Ao final desta etapa o sistema possui pontos no segundo *frame*, representados por  $p'_i$ , que correspondem aos mesmos pontos salientes selecionados no primeiro *frame*, estes, representados por  $p_i$ .

A partir deste conjunto de pares de pontos correspondentes é possível calcular a chamada Matriz Essencial[18, 15]. Esta matriz nada mais é que uma especialização da Matriz Fundamental no caso de a câmera usada ser calibrada (ou seja, se conhecer os parâmetros do modelo da câmera). Essas matrizes são elementos da Geometria Epipolar [18] que relacionam, geometricamente, pontos de imagens de uma mesma cena de vistas diferentes. A matriz Essencial  $E$  está presente na constante epipolar:

$$\hat{p}'^T E \hat{p} = 0 \quad (2.57)$$

onde  $\hat{p}$  e  $\hat{p}'$  são as coordenadas normalizadas [18] dos pontos encontrados na primeira e na segunda imagem, respectivamente. A matriz  $\mathbf{E}$  carrega as informações a respeito da movimentação da câmera entre as capturas a menos de uma escala na translação. Isso fica evidente ao se constatar que  $\mathbf{E}$  pode ser formulada como:

$$\mathbf{E} = [\mathbf{t}]_{\times} \mathbf{R} \quad (2.58)$$

onde  $\mathbf{R}$  representa a matriz de rotação que corresponde a orientação da câmera ao capturar a segunda vista da cena em relação a orientação da primeira captura e  $\mathbf{t} = [t_x \ t_y \ t_z]^T$  corresponde ao vetor de translação, a menos de uma escala, que representa o deslocamento espacial da câmera entre a primeira e a segunda captura. Na Equação 2.58 o vetor de deslocamento está apresentado na forma:

$$[\mathbf{t}]_{\times} = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}. \quad (2.59)$$

É possível aproveitar a formulação de  $\mathbf{E}$  presente na Equação 2.58 para recuperar qual foi a movimentação desempenhada pela câmera entre as duas capturas. Isso é feito a partir de pelo menos cinco pontos comuns as duas imagens, essa solução foi apresentada em [48], mas também pode ser feito de maneira direta a partir de oito pontos comuns as duas imagens como descrito a seguir [21].

A partir da Equação 2.57, percebe-se que, para cada *feature* presente em ambas as imagens, pode-se escrever a constante epipolar na forma:

$$\begin{bmatrix} \hat{u}\hat{u}' & \hat{u}'\hat{v} & \hat{u}' & \hat{u}\hat{v}' & \hat{v}\hat{v}' & \hat{v}' & \hat{u} & \hat{v} & 1 \end{bmatrix} \mathbf{E} = 0 \quad (2.60)$$

onde  $\mathbf{E} = [e_1 \ e_2 \ e_3 \ e_4 \ e_5 \ e_6 \ e_7 \ e_8 \ e_9]^T$  é a representação vetorial da matriz Essencial. Empilhando oito destas constantes, pode-se escrever o sistema de equações lineares na forma  $\mathbf{A}\mathbf{E} = 0$ , e resolvendo-o, é possível encontrar os elementos da matriz  $\mathbf{E}$ . A solução para este sistema de equações homogêneas pode ser facilmente encontrada utilizando a Decomposição de Valores Singulares (DVS). A DVS da matriz  $\mathbf{A}$  é da forma  $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ . Com base neste resultado, e de posse das propriedades que definem a matriz Essencial [18], pode-se recuperar quatro possíveis soluções para  $\mathbf{R}$  e  $\mathbf{t}$  que formam a mesma  $\mathbf{E}$ . Essas quatro possíveis soluções são da forma:

$$\mathbf{R} = \mathbf{U} (\pm \mathbf{W}^T) \mathbf{V}^T \quad (2.61)$$

$$[\mathbf{t}]_{\times} = \mathbf{U} (\pm \mathbf{W}) \mathbf{S} \mathbf{U}^T \quad (2.62)$$

onde

$$\mathbf{W} = \begin{bmatrix} 0 & \pm 1 & 0 \\ \mp 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.63)$$

Apesar de surgirem quatro possíveis soluções, apenas uma delas pode satisfazer as propriedades físicas do movimento e da captura das imagens, pois em três das soluções percebe-se que, em ao menos uma imagem,

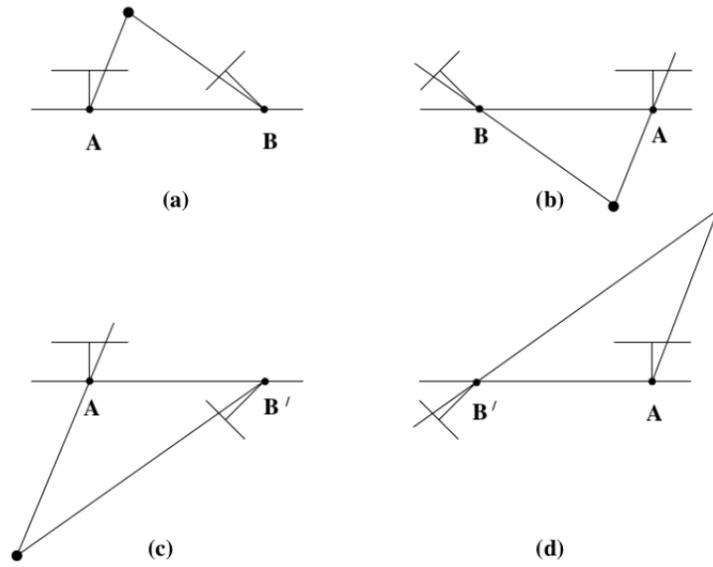


Figura 2.14: As quatro possíveis soluções para a reconstrução de  $\mathbf{R}$  e  $\mathbf{t}$  a partir de  $\mathbf{E}$ .

a cena se encontra atrás da câmera (o que impossibilita a captura da cena pela câmera). Isso pode ser bem observado na Figura 2.14, que apresenta de maneira visual as quatro possíveis soluções.

Assim sendo, finalmente tem-se recuperada a matriz de rotação  $\mathbf{R}_{k,k-1}$  e o vetor de translação  $\mathbf{t}_{k,k-1}$  que caracterizam a transformação de corpo rígido sofrida pela câmera no intervalo entre as capturas das imagens  $I_k$  e  $I_{k-1}$ . A partir deste ponto, seguem as demais operações para a recuperação da trajetória da câmera apresentadas na seção 2.3.1.

A implementação de um sistema de Odometria Visual, como ficou evidente, depende da correta localização dos pontos  $p$  e  $p'$  correspondentes em ambas as imagens, isto é, é crucial que o ponto  $p_i$  encontrado na imagem  $I_{k-1}$  corresponda ao ponto  $p'_i$  presente na imagem  $I_k$ . Caso contrário, ou seja, caso esta correspondência tenha sido mal executada (falsa correspondência) a rotação e translação recuperadas da matriz  $\mathbf{E}$  calculada serão diferentes das reais translação e rotação. Para se evitar esse tipo de erro, é comum que técnicas como o RANSAC sejam implementadas a fim de se remover *outliers*, recomenda-se a leitura de [16] e [49] para se ter uma boa noção das possíveis formas de se aplicar o RANSAC no problema da Odometria Visual.

Ainda podem ser realizadas algumas etapas de refinamento da estimação da trajetória da câmera, como uma varredura de uma janela temporal dos últimos  $n$  frames a fim de se corrigir alguma eventual má estimação de movimentação. A esse tipo de técnica dá-se o nome de *Bundle Adjustment* [50]. O leitor pode encontrar uma boa referência sobre as formas de se implementar essa correção em [15] e [16].

Para finalizar, é importante que o leitor termine esta seção com o entendimento de que a Odometria Visual depende diretamente da correta calibração da câmera. Sem o conhecimento dos parâmetros que descrevem o modelo matemático da câmera, ou mesmo com valores errados destes parâmetros, não é possível chegar ao cálculo da matriz Essencial verdadeira e, conseqüentemente, não se consegue recuperar  $\mathbf{R}$  e  $\mathbf{t}$ . Outra importante observação, é que a Odometria Visual Monocular fornece estimação da trajetória a menos de uma escala que deve ser descoberta a partir de observações de outros sensores[15, 40, 41, 31].

## 2.4 FUSÃO SENSORIAL

Fusão sensorial é o nome que se dá ao processo de combinação ou integração da informação obtida, ou medida, por diferentes sensores a fim de se produzir um conjunto de dados mais específico, abrangente e unificado a respeito de algum evento ou estado que foi observado pelos sensores[27, 6]. De maneira geral, se bem sucedida, a fusão sensorial é capaz de atingir uma acurácia muito maior, isto é, reduz a incerteza a respeito de um estado predito, e realizar inferências mais específicas do que as que se poderiam ser feitas utilizando apenas um único sensor.

Em outras palavras, a utilização de mais de um sensor observando um certo agente ou grandeza (ou grandezas relacionadas) faz com que se tenha uma maior confiança a respeito da medida, além disso, pode-se fazer inferências a respeito de grandezas não observadas diretamente[40, 41].

### 2.4.1 Classificações

A fusão sensorial pode ser classificada quanto a topologia da configuração dos sensores utilizados[6]. Dá-se o nome de Complementar para a configuração na qual os sensores utilizados não dependem uns dos outros diretamente para realizar suas medidas. Neste caso, um sensor monitora uma região ou um estado enquanto outro faz medidas a respeito de outra região ou outro estado e, de posse da combinação das duas medidas, pode-se ter um conhecimento mais completo a respeito do fenômeno observado. Um exemplo bastante simples deste tipo de topologia é o uso de múltiplas câmeras para se observar diferentes partes de um ambiente. Juntando as imagens de todas as câmeras tem-se uma visão completa do ambiente.

Outro tipo de configuração de sensores é a Competitiva[27]. Neste tipo de topologia, tem-se que múltiplos sensores são utilizados para fazer medidas ou observações independentes de um mesmo atributo. Esta configuração fornece leituras com maior robustez uma vez que as múltiplas leituras podem ser utilizadas para detectar falhas ou inconsistências em dados de algum sensor.

Uma terceira forma de se dispor dos sensores é pela chamada topologia Cooperativa[6, 27]. Neste tipo de configuração, informações obtidas de sensores independentes entre si são utilizadas para se derivar uma informação que não seria possível observar com o uso de um único sensor isolado. Um exemplo clássico deste tipo de topologia é o caso da visão stereo, onde a partir da combinação de duas imagens 2D, fornecidas por duas câmeras em diferentes vistas, consegue se estimar a profundidade da cena. É bastante difícil de se projetar uma fusão sensorial cooperativa, uma vez que os dados resultantes desta fusão são muito sensíveis a imprecisões presentes nos sensores individuais.

Importante ressaltar que a fusão sensorial não é restrita a utilização de apenas uma topologia. Isto é, estas configurações apresentadas não são mutualmente excludentes. Configurações híbridas utilizam as vantagens de todos os tipos de configurações e podem ser utilizadas na maioria dos casos em que se deseja realizar a fusão sensorial[6].

Outra maneira de se classificar as técnicas de fusão sensorial é pelo modelo da arquitetura da fusão propriamente dita. Existem basicamente três maneiras de se realizar todos os processos que são executados pela fusão. São elas: A arquitetura Centralizada, a Descentralizada e a Hierárquica. Essas três arquiteturas se diferenciam basicamente pela maneira que se processam os dados obtidos pelos sensores.

Métodos de se realizar fusão sensorial podem ser baseados em: (1) modelos probabilísticos e estatísticos

como inferência Bayesiana, teoria da evidência e estatística robusta; (2) mínimos quadrados, como Filtros de Kalman e otimização; ou mesmo (3) outros métodos heurísticos como Redes Neurais Artificiais e Lógica Fuzzy[6].

A próxima seção irá mostrar em mais detalhes o uso do Filtro de Kalman aplicado a fusão sensorial, uma vez que esta ferramenta será de grande valia para o restante do presente trabalho. Mais informações a cerca das classificações de técnicas de fusão sensorial podem ser encontradas em [6].

## 2.4.2 Filtro de Kalman como Técnica de Fusão Sensorial

O Filtro de Kalman [51] é uma ferramenta muito utilizada como método de estimação de estados de sistemas dinâmicos. Pode-se atribuir o seu sucesso a sua formulação intuitiva no espaço de estados, a sua estrutura de filtragem recursiva ou mesmo de sua estimação baseada na predição e correção[7]. Além disso, é uma técnica de fácil implementação em computadores e com bom repertório na execução de tarefas em tempo real. A seguir será apresentada a formulação discreta do Filtro de Kalman e como ele pode ser interpretado como uma ferramenta de fusão sensorial.

Um sistema dinâmico pode ser apresentado de maneira genérica como:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{w}(k) \quad (2.64)$$

$$\mathbf{z}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{v}(k) \quad (2.65)$$

onde  $\mathbf{x}$  é o vetor de estados a serem estimados de tamanho  $n \times 1$ ;  $\mathbf{u}$  é o vetor de de entradas com dimensão  $p \times 1$ ;  $\mathbf{z}$  é o vetor de dimensão  $m \times 1$  carregando as medições realizadas;  $\mathbf{w}$  é o ruído gaussiano branco do processo, com média nula e matriz de covariância  $\mathbf{Q}$ ;  $\mathbf{v}$  é o ruído gaussiano branco da medição, com média nula e matriz de covariância  $\mathbf{R}$ ;  $\mathbf{A}$  é a matriz de transição de estados com dimensão  $n \times n$ , e é responsável por propagar o estado  $\mathbf{x}$  do instante  $k$  para o instante  $k+1$ ;  $\mathbf{B}$  é o vetor, ou matriz, de magnitude de entrada;  $\mathbf{C}$  é a matriz de dimensão  $m \times n$  que modela a medição do sistema.

O problema de estimação de estados utilizando Filtro de Kalman pode ser entendido como: Dados os modelos dinâmicos do sistema, as características estatísticas do ruído que perturba o processo e as medições, o vetor de medições e o vetor de entrada, estimar, de maneira ótima, o estado do sistema[51, 7].

O processo de estimação de estados com o Filtro de Kalman pode ser dividido em dois momentos: (1) etapa de predição, ou de propagação de estados, quando se utiliza o modelo de evolução do processo do modelo dinâmico para fazer uma predição do estado no instante  $k+1$  a partir do estado no instante  $k$ ; e a (2) etapa de medição, ou correção, na qual, com auxílio do vetor de medições, faz-se a correção do estado predito para que se encontre uma estimação ótima do estado  $\mathbf{x}$ .

Na etapa de predição, tem-se que:

$$\hat{\mathbf{x}}'(k+1) = \mathbf{A}\hat{\mathbf{x}}(k) \quad (2.66)$$

$$\hat{\mathbf{P}}'(k+1) = \mathbf{A}\hat{\mathbf{P}}(k)\mathbf{A}^T + \mathbf{Q} \quad (2.67)$$

onde  $\hat{\mathbf{x}}(k)$  é a estimação do estado  $\mathbf{x}$  no instante  $k$ ,  $\hat{\mathbf{x}}'(k+1)$  é a estimação *a priori* do vetor de estados  $\mathbf{x}$ , obtida ao final da etapa de predição,  $\hat{\mathbf{P}}(k)$  é a matriz de covariância do vetor de estados  $\mathbf{x}$  no instante  $k$  e

$\hat{P}'(k+1)$  é a matriz de covariância a da estimação *a priori* do vetor de estados (também obtida ao final da etapa de predição).

Já na etapa de correção, tem-se que:

$$\hat{x}(k) = \hat{x}'(k) + K(z(k) - C\hat{x}'(k)) \quad (2.68)$$

$$\hat{P}(k) = (I - KC)\hat{P}'(k) \quad (2.69)$$

onde  $K$  é o chamado ganho de Kalman dado por:

$$K = \hat{P}'C^T (C\hat{P}'C^T + R)^{-1}. \quad (2.70)$$

Como pode-se perceber, a estimação final do vetor de estados  $\hat{x}$ , também chamada de estimação *a posteriori*, apresentada pela Equação 2.68, ao final de cada instante de tempo  $k$  é basicamente uma junção, ou mesmo uma fusão, da estimação *a priori* do estado com as medições presentes no vetor  $z(k)$ .

Pode-se fazer uma interpretação do modelo matemático do filtro de Kalman para que ele execute a fusão sensorial de dados obtidos por sensores diferentes. Uma possibilidade é modelar o sistema de forma que a matriz de transição de estados seja nula e o vetor de entradas  $u$  seja composto por medições de um dado sensor, ou grupo de sensores. Desta forma, a matriz  $B$  seria responsável por transformar as leituras dos sensores deste grupo no vetor de estados  $x$ . Sendo assim, quando for calculada a estimação *a posteriori*, o que se terá será a fusão sensorial dos sensores que foram modelados para compor o vetor  $u$  com os sensores que foram modelados para compor o vetor  $z$ . Outras formulações de fusão sensorial baseados no Filtro de Kalman podem ser encontradas em [31].

Importante ressaltar que o Filtro de Kalman deve ser aplicado a sistemas lineares [51, 7]. Quando não se está lidando com sistemas deste tipo, é necessário fazer uso de uma das variações do filtro que tratam de sistemas não lineares, como o Filtro de Kalman Estendido ou o Filtro de Kalman Unscented [8].

### 2.4.3 Vantagens da Fusão Sensorial Visual e Inercial

Combinar as informações obtidas através da Odometria Visual e da Navegação Inercial faz com que cada um dos sistemas aproveite as vantagens do outro para que seja realizada uma estimação muito mais confiável a respeito da posição e orientação do agente rastreado [25].

Como evidenciado na seção 2.2, os sensores utilizados na navegação inercial (acelerômetros e giroscópios) apresentam ruídos em suas leituras mesmo quando o agente por eles rastreado não apresenta movimento, o chamado viés do sensor. Este tipo de erro, quando o agente está parado ou em baixa velocidade, acarretará em uma grande incerteza associada a estimação da posição e orientação, porém, quando o agente se locomove com uma alta velocidade, apresenta incerteza relativamente baixa associada a estimação da posição e orientação[25].

No caso da Odometria Visual, como verificado na seção 2.3, é necessário encontrar *features*, ou pontos de interesse, em pares de imagens consecutivas. Esta tarefa é relativamente simples no caso de o agente rastreado estar se movendo com baixa velocidade, porém, fica complicada quando o mesmo se locomove muito rápido, pois as imagens capturadas vão apresentar borrões e distorções devido a velocidade da câmera durante a captura [25, 15].

Outras duas vantagens de se fundir os dados da Odometria Visual com a Navegação Inercial são: (1) a partir das leituras dos sensores inerciais é possível para o sistema visual reconstruir a ordem de grandeza, ou a escala, em que se deram os movimentos [41, 40]; e (2) com base nas estimações de deslocamento da câmera entre duas capturas, os sensores inerciais podem realizar correções em suas estimações para minimizar os erros provenientes das múltiplas integrações dos vieses dos sensores (que inclusive tornavam as estimações de posição e orientação puramente inerciais praticamente inviáveis, uma vez que os erros crescem com o tempo)[13, 12, 11, 10, 9].

Sendo assim, é possível perceber que a Odometria Visual e a Navegação Inercial são complementares e uma pode se utilizar da outra quando se encontra com dificuldades de estimação da posição e orientação do agente. Diversos trabalhos foram realizados para se propor formas de se fundir esses dois tipos de sensores a fim de se obter o melhor de cada um [30, 52, 9, 53, 31, 54, 55, 10, 11, 12, 37, 56, 26, 57, 58, 13, 59, 60, 40, 14, 61, 62, 63].

## 2.5 CALIBRAÇÃO DE CÂMERA

Até aqui a calibração de câmera foi apresentada como sendo a determinação dos valores dos elementos que compõem a matriz que representa o modelo matemático da câmera. A presente seção irá apresentar com mais detalhes a formulação do modelo de câmera utilizado neste trabalho, bem como irá descrever como se é realizada, normalmente, a calibração de câmeras.

### 2.5.1 Modelo da Câmera

Como já mostrado na seção 2.3, o modelo de câmera perspectiva mais presente na literatura é o baseado no modelo *pinhole* [19]. O modelo *pinhole* é uma das representações matemáticas mais simples para câmeras. Ele é formulado a partir do princípio da colinearidade. Neste modelo, todos os pontos tridimensionais do espaço são projetados na imagem através de um segmento de reta. Este segmento de reta começa no ponto 3D no espaço, passa pelo centro de projeção (o *pinhole*) e atinge o plano da imagem.

Pode-se traduzir este enunciado em termos matemáticos dizendo que as coordenadas do ponto projetado na imagem são dadas por:

$$\begin{cases} u = \frac{X_{cam}}{Z_{cam}} \\ v = \frac{Y_{cam}}{Z_{cam}} \end{cases} \quad (2.71)$$

ou então pela equação matricial em coordenadas homogêneas:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{Z_{cam}} \begin{pmatrix} \mathbf{I} & \mathbf{0} \end{pmatrix} \begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \\ 1 \end{bmatrix} \quad (2.72)$$

onde  $\mathbf{I}$  é a matriz identidade, tem-se que o ponto tridimensional  $\mathbf{P} = [X_{cam} \ Y_{cam} \ Z_{cam}]^T$  está representado no sistema de coordenadas da câmera e as coordenadas  $u$  e  $v$  do ponto projetado estão no sistema de coordenadas do plano da imagem, também referido como {IMG}. A Figura 2.15 ilustra como se dá a projeção

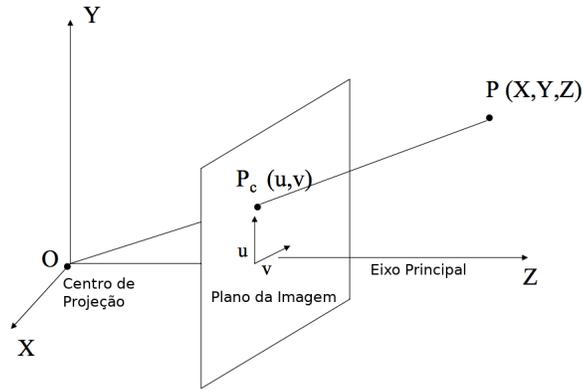


Figura 2.15: Esquemático da projeção de um ponto 3D em uma câmera *pinhole*.

de um ponto  $P$  representado no  $\{CAM\}$  no plano da imagem. Importante notar como são dispostos os eixos do  $\{CAM\}$ , isto é, o eixo de coordenadas  $Z$ , também conhecido como eixo principal da câmera, representa a profundidade a que se encontram os pontos no espaço em relação ao centro de projeção da câmera.

Apesar desta representação simples, o modelo *pinhole* é praticamente teórico, pois ele parte de vários pressupostos que são inválidos em quase todos os casos reais [18]. Por exemplo, a distância  $f$  entre o plano da imagem e o centro de projeção (o *pinhole*) normalmente é diferente de 1. Além disso, os sensores que formam os pixels das imagens digitais é normalmente retangular e não perfeitamente quadrado, sendo assim, um ajuste de escala entre a projeção das coordenadas  $u$  e  $v$  se faz necessário. Levando essas considerações em conta, pode-se modificar o modelo inicialmente apresentado na Equação 2.71 para:

$$\begin{cases} u = \frac{kf}{Z_{cam}} X_{cam} \\ v = \frac{lf}{Z_{cam}} Y_{cam} \end{cases} \quad (2.73)$$

onde foram adicionadas a distância entre a imagem e o centro de projeção da câmera, ou simplesmente, distância focal  $f$ , que é medida em metros, e as constantes  $k$  e  $l$ , ambas mensuradas em  $pixel \times m^{-1}$ , que são responsáveis pela escala da projeção nas direções  $u$  e  $v$ , respectivamente.

Quando se trabalha em unidades de pixel, é esperado que a origem do plano da imagem esteja localizado no canto superior esquerdo da imagem, e não no centro da mesma como a equação 2.73 sugere. Para que isso seja corrigido, é necessário adicionar, para cada pixel projetado na imagem, as constantes  $u_0$  e  $v_0$ . Essas constantes correspondem às coordenadas do centro da imagem, em pixels. Além disso, pode-se reduzir o número de constantes presentes nas equações com o auxílio de  $\alpha = kf$  e  $\beta = lf$ . Dessa forma, a equação de projeção da câmera pode ser reescrita na forma:

$$\begin{cases} u = \frac{\alpha}{Z_{cam}} X_{cam} + u_0 \\ v = \frac{\beta}{Z_{cam}} Y_{cam} + v_0 \end{cases} \quad (2.74)$$

Outra imperfeição do modelo *pinhole* quando comparado com câmeras reais é o fato de que o processo de fabricação da câmera pode tê-la feito de modo que o sistema de coordenadas do plano da imagem fosse ligeiramente distorcido [19]. Em outras palavras, os eixos das coordenadas da imagem formam um ângulo  $\phi$

entre eles que não necessariamente é de  $90^\circ$ . Pode-se modelar este tipo de imperfeição da seguinte forma:

$$\begin{cases} u = \frac{\alpha}{Z_{cam}} X_{cam} - \alpha \cot(\phi) \frac{Y_{cam}}{Z_{cam}} + u_0 \\ v = \frac{\beta}{\sin(\phi)} \frac{Y_{cam}}{Z_{cam}} + v_0 \end{cases} \quad (2.75)$$

Esta nova formulação pode ser representada em forma matricial utilizando coordenadas homogêneas de modo que:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{Z_{cam}} \begin{bmatrix} \alpha & -\alpha \cot(\phi) & u_0 & 0 \\ 0 & \frac{\beta}{\sin(\phi)} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \\ 1 \end{bmatrix}. \quad (2.76)$$

Até agora, assumiu-se que o ponto tridimensional sendo projetado no plano da imagem é representado no sistema de coordenadas da câmera. Só que nem sempre se dispõe desta informação. Na verdade, o mais comum é que se tenha as coordenadas de pontos 3D em relação ao sistema de coordenadas globais [19, 22]. Mas é relativamente fácil obter a conversão de representação de pontos de um sistema de coordenadas para o outro, bastando ter conhecimento da rotação e translação existente entre eles [32].

No caso da formulação do modelo de câmera, se se dispuser apenas da representação do ponto 3D em relação ao {G}, pode-se converter esta representação para {CAM} a partir da matriz de rotação  $\mathbf{R}_g^{cam}$  e do vetor de translação  $\mathbf{t}_g^{cam}$  que representam a rotação e a translação entre o sistema de coordenadas globais e da câmera, respectivamente. Matematicamente:

$$\begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \end{bmatrix} = \mathbf{R}_g^{cam} \begin{bmatrix} X_g \\ Y_g \\ Z_g \end{bmatrix} + \mathbf{t}_g^{cam} \quad (2.77)$$

onde  $X_g, Y_g$  e  $Z_g$  são as coordenadas de um ponto 3D representadas no {G}.

Combinando as Equações 2.76 e 2.77, tem-se, em coordenadas homogêneas:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{Z_{cam}} \mathbf{K} \left( \begin{matrix} \mathbf{R}_g^{cam} & \mathbf{t}_g^{cam} \end{matrix} \right) \begin{bmatrix} X_g \\ Y_g \\ Z_g \\ 1 \end{bmatrix} \quad (2.78)$$

onde  $\mathbf{K}$  é a matriz da câmera, dada por:

$$\mathbf{K} = \begin{bmatrix} \alpha & -\alpha \cot(\phi) & u_0 \\ 0 & \frac{\beta}{\sin(\phi)} & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.79)$$

A Equação 2.78 deixa uma divisão bem clara entre dois tipos de componentes, ou parâmetros que compõem o modelo da câmera. Os valores que compõem a matriz  $\mathbf{K}$  são conhecidos como os parâmetros internos, ou intrínsecos, da câmera. Já os valores que compõem a matriz  $\mathbf{R}_g^{cam}$  e o vetor  $\mathbf{t}_g^{cam}$  são conhecidos como parâmetros externos, ou extrínsecos, da câmera [19]. Essa nomenclatura é bastante direta. Todos aqueles valores que forem relacionados a estrutura interna da câmera, como seu formato ou disposição de componentes,

serão tratados como parâmetros internos e estão explicitados na matriz  $K$ . Já os parâmetros externos são aqueles que relacionam, ou melhor, referem-se a posição e orientação da câmera em relação às coordenadas globais.

Os parâmetros internos da câmera podem ainda contar com alguns elementos que serão responsáveis por modelar as distorções provocadas pela lente da câmera [18, 19]. Os dois modelos de distorção mais utilizados são o de distorção tangencial e o de distorção radial.

## 2.5.2 Implementação da Calibração

O objetivo do processo de calibração de câmera, é obter os parâmetros que modelam o funcionamento do sensor visual [64]. De maneira geral, este processo visa obter tanto os parâmetros intrínsecos quanto os parâmetros extrínsecos da câmera. Entretanto, é válido ressaltar que cada captura de uma câmera está sujeita a alterações de seus parâmetros externos, já que a mesma não necessariamente estará na mesma posição e orientação no espaço [19].

Um dos métodos mais comuns utilizados para realizar a calibração de câmera faz uso da Equação 2.78 para relacionar pontos 3D conhecidos na cena capturada, com os seus respectivos pontos projetados na imagem obtida pela câmera.

Este método faz uso de um tabuleiro de xadrez com dimensões dos lados das casas conhecidas [64, 20]. Este tabuleiro é capturado na imagem da câmera de diversos ângulos, de modo que se tenha imagens dele em todas as regiões do plano de projeção, como mostra a Figura 2.16. Os cantos dos quadrados do tabuleiro de xadrez são muito fáceis de serem reconhecidos na imagem com a utilização de técnicas de detecção de cantos, como o detector Harris [65]. Sendo assim, o algoritmo de calibração é capaz de relacionar os pontos 3D correspondentes aos cantos do tabuleiro com os seus pontos equivalentes na projeção da imagem e, a partir de uma abordagem de mínimos quadrados, realizar a estimação dos parâmetros internos e externos da câmera.

Várias são as implementações de calibradores de câmera, que recebem como entrada fotos de tabuleiro de xadrez em várias posições em conjunto com suas dimensões e apresentam como saída os valores dos parâmetros internos e externos da câmera. Essas ferramentas são muito comuns para serem utilizadas em MATLAB [66], mas também tem sua implementação para OpenCV [67] muito difundida.

### Desvantagens deste método:

Apesar de ser um método amplamente utilizado, a técnica de calibração de câmera que faz uso do tabuleiro de xadrez apresenta alguns problemas. O primeiro deles é que, como descrito, o método necessita de um conhecimento a respeito do ambiente, já que é necessário implantar nas fotos o tabuleiro de dimensões conhecidas. Sendo assim, sempre que se for realizar a calibração de câmera por este método, é necessária uma preparação do cenário.

Outro problema que pode ser identificado é que este método é processado de maneira *offline*. Isto faz com que ele seja executado sozinho, sem qualquer outro procedimento que necessite das imagens rodando em paralelo, até porque, uma outra tarefa que precise dos parâmetros internos para funcionar, como a Odometria Visual, terá de esperar o processo de calibração terminar para dispor dos parâmetros. Isso faz com que o resultado da estimação dos parâmetros internos sejam tomados como constantes, o que pode ser entendido como um outro problema a parte.

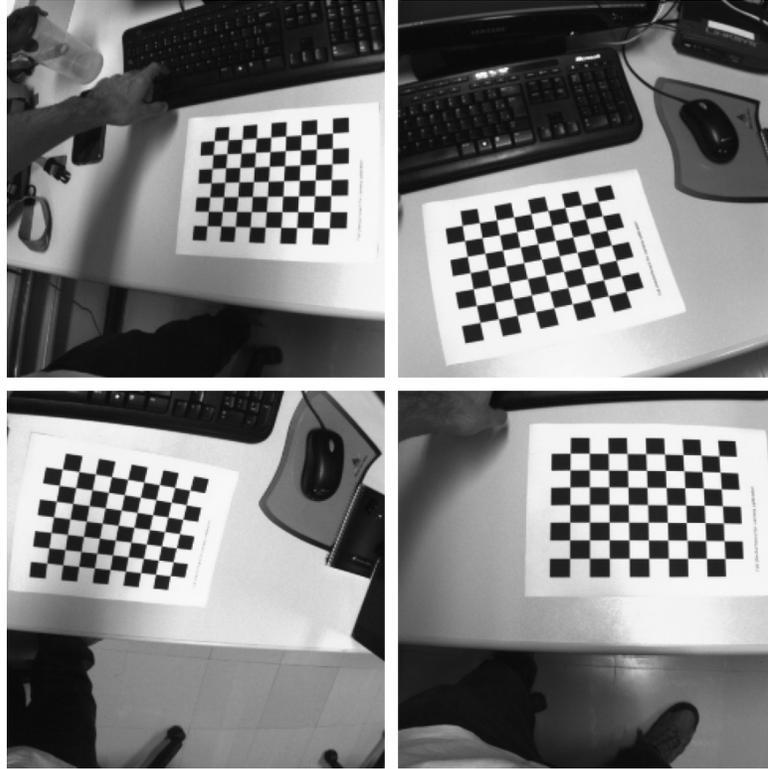


Figura 2.16: Imagens capturadas para a calibração de uma câmera.

Apesar de os parâmetros intrínsecos serem grandezas físicas das quais não se espera alteração, como distância focal e ângulo entre eixos dos sensores, alguns choques mecânicos, trepidações e processos de dilatação térmica sofridos pela câmera podem alterar os valores destas supostas constantes [22]. Como algoritmos que precisam destes parâmetros requerem uma alta exatidão, uma mesma câmera pode precisar ser calibrada mais de uma vez após algum tempo passado de sua calibração anterior, pois, a depender do uso, seus valores podem ter sofrido alguma alteração. Isso faz com que seja necessário submeter a câmera a todo o processo de estimação de seus parâmetros novamente, isto é, capturar outras fotos do tabuleiro de xadrez e executar novamente o calibrador. Além disso, tomar os valores dos parâmetros como constantes é ignorar o fato de que eles foram encontrados a partir de um processo de estimação e, portanto, apresentam uma medida de incerteza associada ao valor estimado.

# 3

## ABORDAGENS DESENVOLVIDAS

---

### 3.1 INTRODUÇÃO

O Capítulo 2 apresentou, de maneira detalhada, boa parte da base teórica por trás da Localização Visual Inercial. Foi apresentada a definição da problemática, apresentou-se o funcionamento físico dos sensores envolvidos, discutiu-se classificações de metodologias de implementação e, claro, os problemas existentes e as dificuldades que tanto a Localização Inercial ou a Odometria Visual apresentam.

Foi ainda evidenciada a característica complementar que a Odometria Visual e a Localização Inercial possuem, tornando-as um excelente par de sensores a serem combinados com técnicas de Fusão Sensorial [25]. Enquanto os sensores inerciais são muito bons para medir a movimentação do agente com altas velocidades, pois nesse caso o ruído provocado pelo viés do sensor fica irrelevante frente as leituras medidas devido ao real movimento, as câmeras apresentam seu pior desempenho em capturar imagens com a nitidez necessária para se realizarem os procedimentos da Odometria Visual, uma vez que as altas velocidades provocam efeito de borramento nas imagens capturadas. No caso de o agente estar se movendo com baixas velocidades, tem-se que as câmeras conseguem capturar as imagens com a nitidez necessária para realizar a estimação de movimento do agente com boa precisão, porém, o SNI realiza estimativas muito enviesadas devido aos ruídos em suas leituras.

Além disso, foi apresentado o fato de que, no caso de um sistema monocular, a Odometria Visual é incapaz de recuperar a escala em que se deram suas estimações de movimentação [15]. Porém, é possível fazer o uso de outros sensores dos quais pode-se extrair a ordem de grandeza em que se deram os deslocamentos do agente, e o SNI executa muito bem esta tarefa. Mais uma das vantagens de se utilizar os dois sistemas conjuntamente.

Por fim, foi mostrado que o sistema de Odometria Visual precisa dos parâmetros internos da câmera. Ou seja, é necessário que a câmera utilizada seja calibrada, e para isto, é comum que a câmera passe por um procedimento de calibração *offline* e que utiliza um ambiente controlado [64]. Esta necessidade de contar com os parâmetros internos da câmera, que ainda podem ter seus valores alterados por situações como choques mecânicos e trepidações sofridos pela câmera (muito comuns na robótica móvel), foi a grande motivação por trás dos algoritmos desenvolvidos.

No presente capítulo serão apresentadas formalmente as duas abordagens propostas para que se implemente um sistema de Localização Visual Inercial com Auto-Calibração da Câmera (LOVIACC). Este tipo de sistema é capaz de estimar os parâmetros internos enquanto realiza as medidas de deslocamento do agente rastreado. Essa calibração é realizada de maneira *online* e sem qualquer necessidade de conhecimento prévio do ambiente. Com a auto-calibração, os valores calibrados não precisam ser tomados como constantes, pois serão estimados continuamente, e trarão consigo uma medida de sua incerteza.

A seção 3.2 descreverá os detalhes da primeira tentativa desta pesquisa de se projetar um sistema de auto-calibração de câmera com o auxílio de uma UMI, que ainda se mostrou um possível método de LOVIACC. Por fim, a seção 3.3 apresentará o LOVIACC propriamente desenvolvido.

## 3.2 AUTO-CALIBRAÇÃO DE CÂMERA COM AUXÍLIO DE UMA UMI

### 3.2.1 Definição do problema

Um agente robótico móvel (mas que também pode ser um veículo qualquer ou mesmo uma pessoa) está em um ambiente genérico carregando uma câmera monocular e uma UMI fixamente acopladas. A medida da rotação  $\mathbf{R}_{cam}^{umi}$  existente entre a câmera e a UMI, bem como a translação  $\mathbf{t}_{cam}^{umi}$  entre os sensores são medidas e conhecidas.

O robô começa a se movimentar em qualquer direção do espaço ao mesmo tempo que a câmera vai realizando capturas de imagens sequenciais da cena ao redor. Essas imagens são denotadas por  $I_{1:n} = \{I_1, I_2, \dots, I_n\}$ . Enquanto a câmera realiza suas capturas, os giroscópios da Unidade de Medição Inercial fazem leituras a cerca da orientação da UMI, que, com o auxílio da rotação conhecida entre os sensores, podem também representar a orientação da câmera no espaço. As leituras da UMI são sincronizadas com a câmera, de forma que sempre que uma nova imagem é obtida pela câmera, é também realizada a leitura dos giroscópios.

A equação matricial em coordenadas homogêneas

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{Z_{cam}} \begin{bmatrix} \alpha & -\alpha \cot(\phi) & u_0 & 0 \\ 0 & \frac{\beta}{\sin(\phi)} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \\ 1 \end{bmatrix} \quad (3.1)$$

já apresentada no capítulo 2, relaciona todos os pontos tridimensionais, representados no {CAM}, com suas respectivas projeções no plano da imagem.

Nota-se no entanto que este modelo de projeção não leva em conta qualquer distorção provocada por aberrações na lente da câmera. Uma possível distorção provocada por lentes é a chamada Distorção Radial [19]. Este tipo de problema pode ser modelado alterando um pouco a matriz da câmera  $\mathbf{K}$ , de modo que

$$\mathbf{K}' = \begin{bmatrix} \frac{1}{\xi} & 0 & 0 \\ 0 & \frac{1}{\xi} & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{K} \quad (3.2)$$

onde  $\mathbf{K}'$  representa a matriz da câmera que leva em conta a distorção radial da lente.  $\xi$  é uma função polinomial da forma

$$\xi = 1 + k_1 r^2 + k_2 r^4 + \dots \quad (3.3)$$

onde  $r := \sqrt{u^2 + v^2}$ . Importante ressaltar que o número de constantes  $k_n$  utilizadas para se modelar a distorção varia de acordo com a precisão que se deseja obter do modelo da distorção.

De maneira análoga ao que acontece na Odometria Visual, a movimentação da câmera no espaço entre capturas de imagens acarreta em um deslocamento da posição de pixels que correspondem a um mesmo ponto tridimensional. Esse deslocamento de pixels, em conjunto com as leituras dos giroscópios da UMI, são utilizados para se realizar a estimação dos parâmetros internos da câmera

$$\begin{bmatrix} \alpha \\ \beta \\ \phi \\ u_0 \\ v_0 \end{bmatrix} \quad (3.4)$$

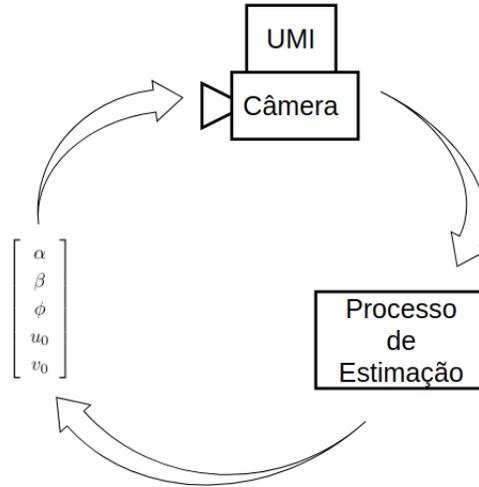


Figura 3.1: Esquemático da estimação dos parâmetros internos da câmera.

bem como a constante  $k_1$  que modela a distorção radial presente na câmera. Esse processo de estimação acontece a cada nova captura de imagem. A Figura 3.1 ilustra, de maneira simples, a dinâmica desse processo de auto-calibração.

### 3.2.2 Algoritmo

O algoritmo que é apresentado nesta seção é fruto de uma das linhas de pesquisa deste trabalho e resultou na publicação do artigo “*Online Inertial-Aided Monocular Camera Self-Calibration*” [23].

Os sensores envolvidos na estimação dos parâmetros internos da câmera são basicamente os giroscópios da UMI e a própria câmera monocular a ser calibrada. Importante ressaltar novamente que as capturas de ambos os sensores são sincronizadas, de forma que sempre que uma nova imagem é capturada pela câmera também é feita uma leitura da orientação da UMI. No instante em que o sistema é iniciado ocorre a primeira aquisição de dados dos sensores, isto é, obtém-se a primeira imagem da câmera, bem como a primeira leitura de orientação da UMI.

Neste instante da primeira leitura dos sensores, é realizado o posicionamento da origem do sistema de coordenadas global, ou simplesmente  $\{G\}$ . Ela é posicionada de forma a coincidir com o centro óptico da câmera. Desta forma, no momento inicial de execução deste algoritmo, os principais sistemas referenciais são coincidentes, o  $\{CAM\}$  e o  $\{G\}$ . A partir das próximas leituras dos sensores poderá ter sido realizado movimento do  $\{CAM\}$  em relação a  $\{G\}$ , que será percebido pelo sistema como será descrito ao longo deste texto.

A primeira imagem capturada pela câmera,  $I_1$ , ainda exerce outro papel fundamental além de ser utilizada como marco temporal para o posicionamento de  $\{G\}$ . É nesta imagem que se aplica o detector de *features* SURF [43] para que sejam encontrados na imagem os pontos mais proeminentes e com boas características para serem rastreados em imagens futuras. É válido apontar que não existe nenhuma obrigação em se utilizar esta técnica, o importante para o algoritmo é que dada a imagem inicial  $I_1$ , que se obtenha pontos nesta imagem a serem encontrados em imagens futuras.

Deste conjunto de pontos proeminentes, é escolhido um de maneira que se espere encontrá-lo nas imagens subsequentes. Este ponto selecionado será rastreado nas próximas imagens e sua localização, em pixels, será crucial para a estimação dos parâmetros internos da câmera.

O ponto escolhido para ser rastreado apresenta coordenadas, em pixels,  $(u_1, v_1)^T$  dentro da imagem. A partir deste ponto no texto,  $u_n$  e  $v_n$  vão ser as coordenadas do ponto rastreado na imagem  $n = \{1, 2, \dots\}$ . Importante lembrar que  $u_0$  e  $v_0$  são variáveis a parte, e representam o ponto principal da imagem, mas tal confusão é evitada uma vez que não se identifica, neste trabalho, imagem com o índice menor que 1.

O ponto  $(u_1, v_1)$  presente na imagem  $I_1$  é uma projeção do ponto tridimensional  $(X_g, Y_g, Z_g)^T$ . Essa projeção é modelada, como já apresentado antes, através da equação:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{Z_{cam}} \mathbf{K} \begin{pmatrix} \mathbf{R}_g^{cam} & \mathbf{t}_g^{cam} \end{pmatrix} \begin{bmatrix} X_g \\ Y_g \\ Z_g \\ 1 \end{bmatrix} \quad (3.5)$$

que será, como mostrado ao longo do restante desta seção, a base de todo o processo de formulação da estimação dos parâmetros internos da câmera.

### 3.2.2.1 Estimador

Devido a não linearidade do processo de projeção na imagem de um ponto tridimensional, escolheu-se realizar o estimador baseado em um Filtro de Kalman Estendido (FKE) discreto [7]. Este tipo de estimador é relativamente simples e pode ser resumido da seguinte forma:

Sendo

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) + \varepsilon_{x,t-1} \quad (3.6)$$

a função de propagação de estados do modelo em questão, onde  $\mathbf{x}_t$  representa o estado do sistema no instante de tempo discreto  $t$  e  $\varepsilon_{x,t}$  corresponde ao vetor de ruído gaussiano de média nula e com matriz de covariância  $Q$  que afeta o vetor de estados. E também sendo

$$\mathbf{y}_t = h(\mathbf{x}_t) + \varepsilon_{y,t} \quad (3.7)$$

o modelo de medições existente no sistema em questão, onde  $\mathbf{y}_t$  representa o vetor de medições no instante de tempo discreto  $t$  e  $\varepsilon_{y,t}$  corresponde ao vetor de ruído gaussiano de média nula e com matriz de covariância  $R$  que afeta o vetor de medições, pode-se definir:

$$\hat{\mathbf{x}}'_t = E(\mathbf{x}_t | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{t-1}), \quad (3.8)$$

$$\hat{\mathbf{x}}_t = E(\mathbf{x}_t | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_t), \quad (3.9)$$

$$\mathbf{P}'_t = E(\mathbf{x}_t \mathbf{x}_t^T | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{t-1}) - \hat{\mathbf{x}}'_t \hat{\mathbf{x}}'^T_t, \quad (3.10)$$

$$\mathbf{P}_t = E(\mathbf{x}_t \mathbf{x}_t^T | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_t) - \hat{\mathbf{x}}_t \hat{\mathbf{x}}_t^T \quad (3.11)$$

onde  $E(\cdot)$  corresponde ao operador de esperança. Estes elementos são denominados, respectivamente, de estimativa a priori, estimativa a posteriori, matriz de covariância a priori e matriz de covariância a posteriori do vetor de estados do sistema em questão.

O FKE visa obter, a cada novo instante de tempo  $t$ , uma atualização a respeito do valor estimado para as variáveis presentes no vetor de estados. Sendo assim, a cada novo instante de tempo, tem-se:

$$\hat{\mathbf{x}}'_t = f(\mathbf{x}_{t-1}) \quad (3.12)$$

como a forma de se calcular a estimativa a priori. A matriz de covariância a priori correspondente a estimativa é calculada como

$$\mathbf{P}'_t = F_t \mathbf{P}_{t-1} F_t^T + Q \quad (3.13)$$

onde o Jacobiano  $F_t$  é dado por:

$$F_t = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{t-1}}. \quad (3.14)$$

Para o cálculo da estimativa a posteriori, é necessário primeiro encontrar o chamado ganho de Kalman  $\mathbf{K}_t$  e a inovação  $\mathbf{z}_t$ . Estes são encontrados por suas formulações:

$$\mathbf{K}_t = \mathbf{P}'_t H_t^T S_t^{-1} \quad (3.15)$$

$$\mathbf{z}_t = \mathbf{y}_t - h(\hat{\mathbf{x}}'_t) \quad (3.16)$$

onde tem-se que a matriz auxiliar  $S_t$  é dada por:

$$S_t = H_t \mathbf{P}'_t H_t^T + R \quad (3.17)$$

e a matriz  $H_t$  é calculado como sendo o Jacobiano:

$$H_t = \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}'_t} \quad (3.18)$$

De posse destes novos elementos, pode-se retomar o cálculo da estimativa posteriori como sendo:

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}'_t + \mathbf{K}_t \mathbf{z}_t \quad (3.19)$$

e a respectiva matriz de covariância correspondente como:

$$\mathbf{P}_t = \mathbf{P}'_t - \mathbf{K}_t H_t \mathbf{P}'_t. \quad (3.20)$$

Está claro, contudo, que o FKE ainda depende de uma estimativa inicial  $\hat{\mathbf{x}}_0$  atrelada a uma matriz de covariância  $\mathbf{P}_0$  para servir de pontapé inicial do estimador. Tal estimativa pode uma medida prévia de valores que se deseja estimar, ou mesmo um valor completamente arbitrário. Importante deixar claro que esta estimativa inicial impacta diretamente a convergência do filtro.

Essa é apenas uma possível maneira de se resumir o FKE. Para resolver qualquer dúvida ou caso haja necessidade de maior aprofundamento a respeito do assunto, recomenda-se a leitura de [7]. O caso discreto

foi escolhido pois as imagens, e consequentemente as leituras dos giroscópios da UMI, chegam de maneira discreta para serem processados. Nos tópicos seguintes serão apresentados, de maneira mais detalhada, o vetor de estados a ser estimado, a função de propagação de estados e o modelo de medição projetado para o estimador dos parâmetros internos da câmera.

### 3.2.2.2 Vetor de Estados

Como mostrado na seção anterior, o estimador baseado no FKE entrega, a cada novo instante de tempo  $t$ , uma nova estimativa dos elementos do vetor de estados a partir do modelo do sistema em questão em conjunto com o vetor de medições obtido.

No caso do FKE projetado, vetor de estados do sistema é dado por:

$$\mathbf{x}_t = \begin{bmatrix} X \\ Y \\ Z \\ t_{1,t} \\ t_{2,t} \\ t_{3,t} \\ \alpha \\ \beta \\ \phi \\ k_1 \\ u_0 \\ v_0 \end{bmatrix}. \quad (3.21)$$

Cada uma das variáveis presentes no vetor de estados será devidamente detalhada a seguir.

As primeiras três variáveis,  $\begin{pmatrix} X & Y & Z \end{pmatrix}^T$ , correspondem às coordenadas globais do ponto tridimensional projetado no pixel  $(u_1, v_1)^T$  que foi encontrado como sendo o mais proeminente na imagem  $I_1$ . Como já mencionado anteriormente, este pixel será rastreado nas imagens subsequentes, mas é importante reparar que, independente do *frame* ou das coordenadas em que este pixel for encontrado, ele sempre corresponderá ao mesmo ponto tridimensional do espaço.

Retornando ao vetor de estados, as variáveis  $\begin{pmatrix} t_{1,t} & t_{2,t} & t_{3,t} \end{pmatrix}^T$  correspondem ao vetor de translação da câmera em relação a origem do  $\{G\}$  no instante de tempo  $t$ .

As variáveis seguintes,  $\begin{pmatrix} \alpha & \beta & \phi & k_1 & u_0 & v_0 \end{pmatrix}^T$ , são os parâmetros internos da câmera, que modelam a matriz

$$\mathbf{K} = \begin{bmatrix} \frac{1}{\xi} & 0 & 0 \\ 0 & \frac{1}{\xi} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha & -\alpha \cot(\phi) & u_0 \\ 0 & \frac{\beta}{\sin(\phi)} & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.22)$$

Com isso todos os elementos do vetor de estados utilizado na estimativa já estão detalhados. Os próximos tópicos vão tratar a respeito da função de propagação de estados e do modelo de medição projetado.

### 3.2.2.3 Função de propagação de estados

A função de propagação de estados é o que modela o estado atual a partir do estado anterior. Como evidenciado na seção 3.2.2.1, ela tem impacto direto, juntamente com o modelo de medição, na estimação dos parâmetros a partir do FKE.

Na proposta de sistema de estimação que está sendo apresentada, optou-se por deixar a cargo apenas do modelo de medição a tarefa de se realizar as correções das estimativas. Tal escolha ficará mais compreensível na próxima seção, porém, pode ser justificada pela ausência de uma fórmula que possibilitasse a previsão da evolução dos elementos do vetor de estados, sendo possível apenas atribuir a estes elementos os seus valores previamente encontrados.

Traduzindo isto para termos matemáticos, tem-se que a função de propagação de estados  $f$  é dada simplesmente por:

$$x_t = x_{t-1} + \varepsilon_{x,t-1}. \quad (3.23)$$

### 3.2.2.4 Modelo de medição

Com isso, chega-se ao tópico mais importante desta seção, onde de fato é apresentada a contribuição deste trabalho. Será mostrada agora a formulação da modelagem da medição do FKE proposto. Este modelo de medição é o grande responsável pela estimação dos parâmetros.

A primeira captura de imagem com a câmera em questão é importante para a marcação da origem do  $\{G\}$ , bem como para a determinação do ponto mais proeminente da imagem que será rastreado. A partir da captura da segunda imagem, ou seja, de  $I_2$  em diante, tem-se que a nova imagem obtida é submetida a um procedimento de rastreamento, onde o objetivo é encontrar, na nova imagem, um ponto que corresponda ao pixel mais proeminente de  $I_1$ . Quando este ponto é encontrado, são realizadas algumas medições e comparações entre a sua posição e a posição do ponto equivalente em  $I_1$  para formar o vetor de medições do FKE.

O vetor de medições é dado por:

$$y_t = \begin{bmatrix} \Delta u_t \\ \Delta v_t \\ e_{u,t} \\ e_{v,t} \end{bmatrix} \quad (3.24)$$

onde os dois primeiros elementos,  $\Delta u$  e  $\Delta v$ , correspondem ao deslocamento da posição do pixel que está sendo rastreado entre a primeira imagem,  $I_1$ , e a imagem atual,  $I_t$ . Já os elementos seguintes do vetor,  $e_{u,t}$  e  $e_{v,t}$ , são as coordenadas, em pixels, da epípola na imagem atual. Todos estes elementos serão melhor detalhados e terão sua formulação apresentada no restante deste tópico.

Analisando a Equação 3.5, observa-se que o deslocamento do pixel rastreado se dá em virtude da transformação de corpo rígido sofrida pela câmera em relação ao sistema de coordenadas globais. Em outras palavras, o pixel rastreado se movimenta dentro da imagem graças a movimentação da câmera no ambiente.

Dito isto, percebe-se ser possível modelar este deslocamento de pixel entre imagens a partir da Equação 3.5 e de leituras dos giroscópios da UMI, como será mostrado a seguir.

No instante da captura de  $I_1$ , tem-se que os dois sistemas de referencias, {CAM} e {G}, são coincidentes. Dessa forma, como espécie de condição inicial, pode-se dizer que:

$$\mathbf{R}_{g,1}^{cam} = \mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.25)$$

bem como

$$\mathbf{t}_{g,1}^{cam} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad (3.26)$$

uma vez que não há nem translação nem rotação entre a câmera e a origem do sistema de coordenadas global.

Porém, essa condição só é válida para  $I_1$ , e a partir das imagens subsequentes, a movimentação que a câmera vier a sofrer será traduzida em uma transformação de corpo rígido descrita por  $\mathbf{R}_{g,t}^{cam}$  e  $\mathbf{t}_{g,t}^{cam}$  diferentes dos apresentados como condição inicial.

O vetor de translação  $\mathbf{t}_{g,t}^{cam}$  é uma das grandezas a ser estimado com o FKE, como mostrado no tópico 3.2.2.2. Já a matriz de rotação  $\mathbf{R}_{g,t}^{cam}$  pode ser encontrada a partir das leituras da orientação da UMI realizadas pelos seus giroscópios como já foi muito bem detalhado no Capítulo 2, mais especificamente no tópico 2.2.1. Como a UMI está fixamente acoplada ao corpo da câmera, sempre que houver uma movimentação da mesma, a UMI também se deslocará, porém, a transformação de corpo rígido, descrita por  $\mathbf{R}_{cam}^{umi}$  e  $\mathbf{t}_{cam}^{umi}$ , entre a câmera e a UMI permanecerá constante e, como dito anteriormente, conhecida. Sendo assim, é possível encontrar a orientação da câmera em relação ao {G}.

O primeiro passo para atingir este objetivo é calibrar as leituras dos giroscópios para que as mesmas correspondam a rotações em relação a orientação inicial da UMI, isto é feito pré multiplicando todas as medidas realizadas pelo giroscópio pelo inverso da primeira leitura. Matematicamente:

$$[\mathbf{R}_g^{umi}]_t = [\mathbf{R}_g^{umi-desescalibrada}]_1^{-1} [\mathbf{R}_g^{umi-desescalibrada}]_t. \quad (3.27)$$

E, finalmente, para encontrar a orientação da câmera, basta fazer uso da rotação conhecida entre os dois sensores, de modo que a orientação da câmera em relação ao {G} é dada matematicamente por:

$$[\mathbf{R}_g^{cam}]_t = [\mathbf{R}_g^{umi}]_t [\mathbf{R}_{cam}^{umi}]^{-1}. \quad (3.28)$$

É importante ressaltar que o processo de obtenção da orientação da UMI a partir das leituras de seus giroscópios não está inserido no FKE que está sendo apresentado. As leituras da orientação da UMI chegam já filtradas de modo que o presente algoritmo as utiliza como constantes.

Tem-se então que a cada captura de uma imagem, as leituras da UMI fornecem indiretamente a orientação da câmera em relação ao {G} no momento em que a imagem foi obtida. Esta orientação é representada pela matriz de rotação  $\mathbf{R}_{g,t}^{cam}$  e é expressa na forma:

$$\mathbf{R}_{g,t}^{cam} = \begin{bmatrix} R_{1,t} & R_{2,t} & R_{3,t} \\ R_{4,t} & R_{5,t} & R_{6,t} \\ R_{7,t} & R_{8,t} & R_{9,t} \end{bmatrix}. \quad (3.29)$$

Esta matriz de rotação em conjunto com o vetor de translação  $\mathbf{t}_{g,t}^{cam}$ , presente no vetor de estados, podem ser utilizados para representar as coordenadas de um ponto tridimensional, com suas componentes no {G} conhecidas, no sistema de coordenadas da câmara após ela ter realizado algum movimento em relação à sua posição inicial. Matematicamente, isso se dá na forma:

$$\begin{bmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \end{bmatrix} = \mathbf{R}_g^{cam} \begin{bmatrix} x_g \\ y_g \\ z_g \end{bmatrix} + \mathbf{t}_g^{cam}. \quad (3.30)$$

No caso da formulação do modelo de medição em questão, é importante conseguir representar o ponto tridimensional, que corresponde ao pixel que é rastreado entre as imagens, no {CAM} independentemente de onde este sistema de coordenadas se encontre em relação ao {G}. Isso é necessário para que se possa utilizar a fórmula de projeção do ponto 3D para as coordenadas do pixel, apresentada na Equação 3.5, para todas as imagens capturadas. Para tal feito, sempre que uma nova imagem for capturada, cria-se as variáveis auxiliares:

$$\Lambda_t = R_{1,t}X + R_{2,t}Y + R_{3,t}Z + t_{1,t} \quad (3.31)$$

$$\mu_t = R_{4,t}X + R_{5,t}Y + R_{6,t}Z + t_{2,t} \quad (3.32)$$

$$\Gamma_t = R_{7,t}X + R_{8,t}Y + R_{9,t}Z + t_{3,t} \quad (3.33)$$

que nada mais são do que a representação das coordenadas do ponto tridimensional  $\begin{pmatrix} X & Y & Z \end{pmatrix}^T$ , presente no vetor de estados, representadas no {CAM} no instante de tempo  $t$ .

Terminada a apresentação destas três variáveis auxiliares, pode-se seguir adiante e apresentar a formulação da expressão das variáveis de medição  $\Delta u_t$  e  $\Delta v_t$ . Como descrito anteriormente, essas variáveis correspondem ao deslocamento, em pixels, do ponto rastreado entre a primeira imagem capturada e sua posição na imagem atual. Ou seja:

$$\Delta u_t = u_t - u_1 \quad (3.34)$$

e

$$\Delta v_t = v_t - v_1. \quad (3.35)$$

No instante da primeira captura, ou seja, quando se encontra  $u_1$  e  $v_1$ , tem-se que as condições iniciais apresentadas nas Equações 3.25 e 3.26 são verdadeiras, logo, a formulação matemática para as coordenadas do pixel encontrado é dada por:

$$\begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = \frac{1}{Z} \mathbf{K} \begin{pmatrix} \mathbf{R}_{g,1}^{cam} & \mathbf{t}_{g,1}^{cam} \end{pmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (3.36)$$

$$\begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = \frac{1}{Z} \mathbf{K} \begin{pmatrix} \mathbf{I} & \mathbf{0} \end{pmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (3.37)$$

Substituindo  $\mathbf{K}$  pela Equação 3.22, tem-se:

$$\begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} \frac{1}{\xi_1} & 0 & 0 \\ 0 & \frac{1}{\xi_1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha & -\alpha \cot(\phi) & u_0 & 0 \\ 0 & \frac{\beta}{\sin(\phi)} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \quad (3.38)$$

Percebe-se então que pode-se expressar  $u_1$  de maneira algébrica como:

$$u_1 = \frac{1}{\xi_1} \frac{1}{Z} (\alpha X - \alpha \cot(\phi) Y + u_0 Z) \quad (3.39)$$

$$u_1 = \frac{1}{\xi_1} \frac{1}{Z} \alpha (X - \cot(\phi) Y) + u_0. \quad (3.40)$$

E pode-se realizar a mesma expansão algébrica para  $v_1$ , que fica expresso como:

$$v_1 = \frac{1}{\xi_1} \frac{1}{Z} \left( \frac{\beta}{\sin(\phi)} Y + v_0 Z \right) \quad (3.41)$$

$$v_1 = \frac{1}{\xi_1} \frac{1}{Z} \frac{\beta}{\sin(\phi)} Y + v_0. \quad (3.42)$$

Para finalizar então as formulações de  $\Delta u_t$  e  $\Delta v_t$ , é necessário elaborar expressões algébricas para as coordenadas genéricas  $u_t$  e  $v_t$  do pixel rastreado nas imagens subsequentes a  $I_1$ . De maneira análoga a realizada para encontrar as expressões de  $u_1$  e  $v_1$ , inicia-se com:

$$\begin{bmatrix} u_t \\ v_t \\ 1 \end{bmatrix} = \frac{1}{Z} \mathbf{K} \begin{pmatrix} \mathbf{R}_{g,t}^{cam} & \mathbf{t}_{g,t}^{cam} \end{pmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (3.43)$$

só que desta vez as condições iniciais não são válidas, pois a câmera pode ter se deslocado em relação à sua posição inicial. Desta forma, pode-se fazer uso das variáveis  $\Lambda_t$ ,  $\mu_t$  e  $\Gamma_t$  de modo que:

$$\begin{bmatrix} u_t \\ v_t \\ 1 \end{bmatrix} = \frac{1}{\Gamma_t} \mathbf{K} \begin{pmatrix} \mathbf{I} & \mathbf{0} \end{pmatrix} \begin{bmatrix} \Lambda_t \\ \mu_t \\ \Gamma_t \\ 1 \end{bmatrix} \quad (3.44)$$

Substituindo  $\mathbf{K}$  pela Equação 3.22, tem-se:

$$\begin{bmatrix} u_t \\ v_t \\ 1 \end{bmatrix} = \frac{1}{\Gamma_t} \begin{bmatrix} \frac{1}{\xi_t} & 0 & 0 \\ 0 & \frac{1}{\xi_t} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha & -\alpha \cot(\phi) & u_0 & 0 \\ 0 & \frac{\beta}{\sin(\phi)} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Lambda_t \\ \mu_t \\ \Gamma_t \\ 1 \end{bmatrix} \quad (3.45)$$

Dessa maneira, pode-se expressar  $u_t$  algebricamente como:

$$u_t = \frac{1}{\xi_t} \frac{1}{\Gamma_t} (\alpha \Lambda_t - \alpha \cot(\phi) \mu_t + u_0 \Gamma_t) \quad (3.46)$$

$$u_t = \frac{1}{\xi_1} \frac{1}{\Gamma_t} \alpha (\Lambda_t - \cot(\phi) \mu_t) + u_0. \quad (3.47)$$

enquanto que a expressão de  $v_t$  é dada por:

$$v_t = \frac{1}{\xi_t} \frac{1}{\Gamma_t} \left( \frac{\beta}{\sin(\phi)} \mu_t + v_0 \Gamma_t \right) \quad (3.48)$$

$$v_t = \frac{1}{\xi_t} \frac{1}{\Gamma_t} \frac{\beta}{\sin(\phi)} \mu_t + v_0. \quad (3.49)$$

Sendo assim, basta substituir as Equações 3.40 e 3.47 em 3.34 para obter a expressão final de  $\Delta u_t$ :

$$\Delta u_t = \left( \frac{1}{\xi_1} \frac{1}{\Gamma_t} \alpha (\Lambda_t - \cot(\phi) \mu_t) + u_0 \right) - \left( \frac{1}{\xi_1} \frac{1}{Z} \alpha (X - \cot(\phi) Y) + u_0 \right) \quad (3.50)$$

$$\Delta u_t = \alpha \left( \frac{\Lambda_t}{\xi_t \Gamma_t} - \frac{X}{\xi_1 Z} \right) - \alpha \cot(\phi) \left( \frac{\mu_t}{\xi_t \Gamma_t} + \frac{Y}{\xi_1 Z} \right) \quad (3.51)$$

De maneira semelhante, substituindo as Equações 3.42 e 3.49 em 3.35, encontra-se a expressão final de  $\Delta v_t$  como sendo:

$$\Delta v_t = \left( \frac{1}{\xi_t} \frac{1}{\Gamma_t} \frac{\beta}{\sin(\phi)} \mu_t + v_0 \right) - \left( \frac{1}{\xi_1} \frac{1}{Z} \frac{\beta}{\sin(\phi)} Y + v_0 \right) \quad (3.52)$$

$$\Delta v_t = \frac{\beta}{\sin(\phi)} \left( \frac{\mu_t}{\xi_t \Gamma_t} - \frac{Y}{\xi_1 Z} \right) \quad (3.53)$$

O que conclui a elaboração destes dois elementos do vetor de medição apresentado na Equação 3.24. Os outros dois elementos,  $e_{u,t}$  e  $e_{v,t}$ , são, como dito anteriormente, as coordenadas da epípola na imagem  $I_t$ .

A epípola é um importante elemento da Geometria Epipolar [18], e basicamente pode ser entendido como sendo o ponto de interseção de todas as linhas epipolares ou, de maneira equivalente, como sendo a imagem em uma vista do centro da câmera em outra vista. Sendo mais claro, a epípola na imagem  $I_t$  é o ponto onde seria projetado o centro da câmera no instante em que foi capturada a imagem  $I_1$ .

Pode-se colocar essa definição em termos matemáticos, fazendo uso de coordenadas homogêneas, da seguinte forma:

$$\mathbf{e}_t = \mathbf{K} \mathbf{t}_{g,t}^{cam}, \quad (3.54)$$

pois o vetor  $t_{g,t}^{cam}$  apresenta a translação da câmera no instante de tempo  $t$  em relação ao  $\{G\}$  que, dadas as condições iniciais, também é onde se encontrou a origem da câmera quando foi capturada a primeira imagem  $I_1$ . Dando sequência na expansão da formulação matemática da epípola:

$$\mathbf{e}_t = \begin{bmatrix} e_{1,t} \\ e_{2,t} \\ e_{3,t} \end{bmatrix} = \begin{bmatrix} \alpha & -\alpha \cot(\phi) & u_0 \\ 0 & \frac{\beta}{\sin(\phi)} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} t_{1,t} \\ t_{2,t} \\ t_{3,t} \end{bmatrix} \quad (3.55)$$

o que leva às expressões algébricas:

$$e_{1,t} = \alpha t_{1,t} - \alpha \cot(\phi) t_{2,t} + u_0 t_{3,t}, \quad (3.56)$$

$$e_{2,t} = \beta \frac{t_{2,t}}{\sin(\phi)} + v_0 t_{3,t}, \quad (3.57)$$

$$e_{3,t} = t_{3,t}. \quad (3.58)$$

Convertendo então o ponto  $e_t$ , dado em coordenadas homogêneas, para coordenadas em pixel na imagem  $I_t$ , tem-se que  $e_{u,t}$  é dado por:

$$e_{u,t} = \frac{e_{1,t}}{e_{3,t}} \quad (3.59)$$

$$e_{u,t} = \frac{\alpha t_{1,t} - \alpha \cot(\phi) t_{2,t} + u_0 t_{3,t}}{t_{3,t}} \quad (3.60)$$

$$e_{u,t} = \frac{\alpha}{t_{3,t}} (t_{1,t} - \cot(\phi) t_{2,t}) + u_0 \quad (3.61)$$

e que  $e_{v,t}$  é dado por:

$$e_{v,t} = \frac{e_{2,t}}{e_{3,t}} \quad (3.62)$$

$$e_{v,t} = \frac{\beta \frac{t_{2,t}}{\sin(\phi)} + v_0 t_{3,t}}{t_{3,t}} \quad (3.63)$$

$$e_{v,t} = \frac{\beta t_{2,t}}{\sin(\phi) t_{3,t}} + v_0. \quad (3.64)$$

Com isso, tem-se então que todos os elementos do vetor de medições apresentam uma formulação algébrica em função das variáveis de estado, como propõe a abordagem do estimador baseado no FKE.

### 3.2.2.5 Jacobianos

Como apresentado no tópico 3.2.2.1, o FKE necessita, para sua correta implementação, do cálculo dos jacobianos  $F_t$  e  $H_t$ . No caso do modelo projetado, a função de propagação de estados  $f$  apresentada na Equação 3.23, fornece o jacobiano:

$$F_t = \frac{\partial f}{\partial x} \Big|_{x=\hat{x}_{t-1}} \quad (3.65)$$

$$= \mathbf{I}. \quad (3.66)$$

Já a função que modela as variáveis de medição  $h$ , fornece um jacobiano um pouco mais complexo:

$$H_t = \frac{\partial h}{\partial x} \Big|_{x=\hat{x}'_{t-1}} \quad (3.67)$$

$$= \begin{bmatrix} \frac{\partial h_{\Delta u}}{\partial X} & \frac{\partial h_{\Delta u}}{\partial Y} & \frac{\partial h_{\Delta u}}{\partial Z} & \frac{\partial h_{\Delta u}}{\partial t_1} & \frac{\partial h_{\Delta u}}{\partial t_2} & \frac{\partial h_{\Delta u}}{\partial t_2} & \frac{\partial h_{\Delta u}}{\partial t_2} & \frac{\partial h_{\Delta u}}{\partial t_3} & \frac{\partial h_{\Delta u}}{\partial \alpha} & \frac{\partial h_{\Delta u}}{\partial \beta} & \frac{\partial h_{\Delta u}}{\partial \phi} & \frac{\partial h_{\Delta u}}{\partial k_1} & \frac{\partial h_{\Delta u}}{\partial u_0} & \frac{\partial h_{\Delta u}}{\partial v_0} \\ \frac{\partial h_{\Delta v}}{\partial X} & \frac{\partial h_{\Delta v}}{\partial Y} & \frac{\partial h_{\Delta v}}{\partial Z} & \frac{\partial h_{\Delta v}}{\partial t_1} & \frac{\partial h_{\Delta v}}{\partial t_2} & \frac{\partial h_{\Delta v}}{\partial t_2} & \frac{\partial h_{\Delta v}}{\partial t_2} & \frac{\partial h_{\Delta v}}{\partial t_3} & \frac{\partial h_{\Delta v}}{\partial \alpha} & \frac{\partial h_{\Delta v}}{\partial \beta} & \frac{\partial h_{\Delta v}}{\partial \phi} & \frac{\partial h_{\Delta v}}{\partial k_1} & \frac{\partial h_{\Delta v}}{\partial u_0} & \frac{\partial h_{\Delta v}}{\partial v_0} \\ \frac{\partial h_{e u}}{\partial X} & \frac{\partial h_{e u}}{\partial Y} & \frac{\partial h_{e u}}{\partial Z} & \frac{\partial h_{e u}}{\partial t_1} & \frac{\partial h_{e u}}{\partial t_2} & \frac{\partial h_{e u}}{\partial t_2} & \frac{\partial h_{e u}}{\partial t_2} & \frac{\partial h_{e u}}{\partial t_3} & \frac{\partial h_{e u}}{\partial \alpha} & \frac{\partial h_{e u}}{\partial \beta} & \frac{\partial h_{e u}}{\partial \phi} & \frac{\partial h_{e u}}{\partial k_1} & \frac{\partial h_{e u}}{\partial u_0} & \frac{\partial h_{e u}}{\partial v_0} \\ \frac{\partial h_{e v}}{\partial X} & \frac{\partial h_{e v}}{\partial Y} & \frac{\partial h_{e v}}{\partial Z} & \frac{\partial h_{e v}}{\partial t_1} & \frac{\partial h_{e v}}{\partial t_2} & \frac{\partial h_{e v}}{\partial t_2} & \frac{\partial h_{e v}}{\partial t_2} & \frac{\partial h_{e v}}{\partial t_3} & \frac{\partial h_{e v}}{\partial \alpha} & \frac{\partial h_{e v}}{\partial \beta} & \frac{\partial h_{e v}}{\partial \phi} & \frac{\partial h_{e v}}{\partial k_1} & \frac{\partial h_{e v}}{\partial u_0} & \frac{\partial h_{e v}}{\partial v_0} \end{bmatrix} \quad (3.68)$$

onde as componentes do jacobiano que são calculadas baseadas na formulação de  $\Delta u_t$ , apresentada na Equação 3.51, são dadas por:

$$\frac{\partial h_{\Delta u}}{\partial X} = \alpha \left( \frac{R_1}{\xi_t \Gamma_t} - \frac{R_7 \Lambda_t}{\xi_t \Gamma_t^2} - \frac{1}{\xi_1 Z} \right) - \alpha \cot(\phi) \left( \frac{R_4}{\xi_t \Gamma_t} - \frac{R_7}{\xi_t \Gamma_t^2} \right) \quad (3.69)$$

$$\frac{\partial h_{\Delta u}}{\partial Y} = \alpha \left( \frac{R_2}{\xi_t \Gamma_t} - \frac{R_8}{\xi_t \Gamma_t^2} \right) - \alpha \cot(\phi) \left( -\frac{R_8 \mu_t}{\xi_t \Gamma_t^2} + \frac{R_5}{\xi_t \Gamma_t} + \frac{1}{\xi_1 Z} \right) \quad (3.70)$$

$$\frac{\partial h_{\Delta u}}{\partial Z} = \alpha \left( \frac{R_3}{\xi_t \Gamma_t} - \frac{R_9 \Lambda_t}{\xi_t \Gamma_t^2} + \frac{X}{\xi_1 Z^2} \right) - \alpha \cot(\phi) \left( -\frac{R_9 \mu_t}{\xi_t \Gamma_t^2} + \frac{R_6}{\xi_t \Gamma_t} - \frac{Y}{\xi_1 Z^2} \right) \quad (3.71)$$

$$\frac{\partial h_{\Delta u}}{\partial t_1} = \frac{\alpha}{\xi_t \Gamma_t} \quad (3.72)$$

$$\frac{\partial h_{\Delta u}}{\partial t_2} = \frac{-\alpha \cot(\phi)}{\xi_t \Gamma_t} \quad (3.73)$$

$$\frac{\partial h_{\Delta u}}{\partial t_3} = \frac{-\alpha \Lambda_t}{\xi_t \Gamma_t} + \frac{\alpha \cot(\phi) \mu_t}{\xi_t \Gamma_t} \quad (3.74)$$

$$\frac{\partial h_{\Delta u}}{\partial \alpha} = \left( \frac{\Lambda_t}{\xi_t \Gamma_t} - \frac{X}{\xi_1 Z} \right) - \cot(\phi) \left( \frac{\mu_t}{\xi_t \Gamma_t} + \frac{Y}{\xi_1 Z} \right) \quad (3.75)$$

$$\frac{\partial h_{\Delta u}}{\partial \beta} = 0 \quad (3.76)$$

$$\frac{\partial h_{\Delta u}}{\partial \phi} = \alpha \csc^2(\phi) \left( \frac{\mu_t}{\xi_t \Gamma_t} + \frac{Y}{\xi_1 Z} \right) \quad (3.77)$$

$$\frac{\partial h_{\Delta u}}{\partial k_1} = \alpha \left( \frac{r_1^2 X}{\xi_1^2 Z} - \frac{r_t^2 \Lambda_t}{\xi_t^2 \Gamma_t} \right) - \alpha \cot(\phi) \left( \frac{r_t^2 \mu_t}{\xi_t^2 \Gamma_t} + \frac{r_1^2 Y}{\xi_1^2 Z} \right) \quad (3.78)$$

$$\frac{\partial h_{\Delta u}}{\partial u_0} = 0 \quad (3.79)$$

$$\frac{\partial h_{\Delta u}}{\partial v_0} = 0 \quad (3.80)$$

Já as componentes de  $H_t$  que foram calculadas a partir da expressão algébrica de  $\Delta v_t$ , apresentada na Equação 3.53, são:

$$\frac{\partial h_{\Delta v}}{\partial X} = \beta \csc(\phi) \left( \frac{R_4}{\xi_t \Gamma_t} - \frac{R_7 \mu_t}{\xi_t \Gamma_t^2} \right) \quad (3.81)$$

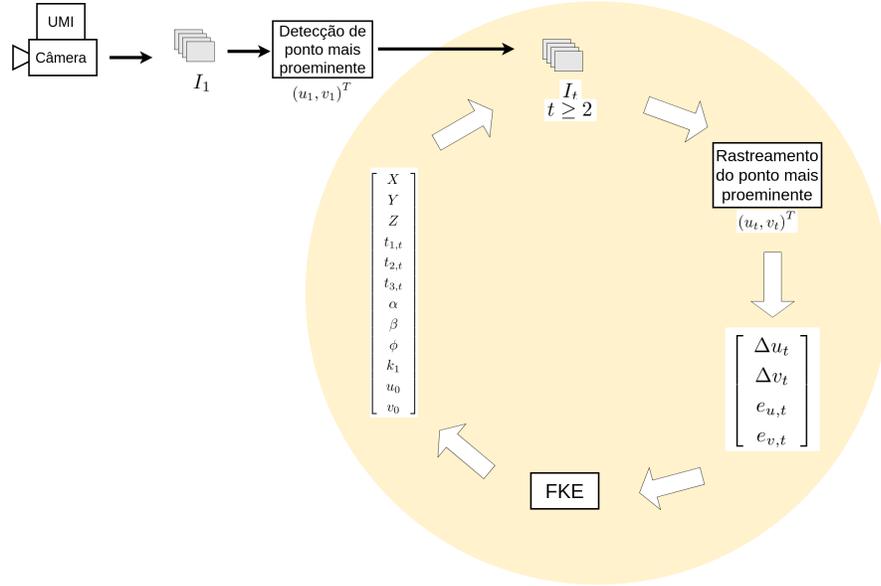


Figura 3.2: Esquemático do sistema de Auto-Calibração de Câmera com Auxílio de uma UMI.

$$\frac{\partial h_{\Delta v}}{\partial Y} = \beta \csc(\phi) \left( -\frac{R_8 \mu_t}{\xi_t \Gamma_t^2} + \frac{R_5}{\xi_t \Gamma_t} - \frac{1}{\xi_1 Z} \right) \quad (3.82)$$

$$\frac{\partial h_{\Delta v}}{\partial Z} = \beta \csc(\phi) \left( -\frac{R_9 \mu_t}{\xi_t \Gamma_t^2} + \frac{R_6}{\xi_t \Gamma_t} + \frac{Y}{\xi_1 Z^2} \right) \quad (3.83)$$

$$\frac{\partial h_{\Delta v}}{\partial t_1} = 0 \quad (3.84)$$

$$\frac{\partial h_{\Delta v}}{\partial t_2} = \beta \csc(\phi) \frac{1}{\xi_t \Gamma_t} \quad (3.85)$$

$$\frac{\partial h_{\Delta v}}{\partial t_3} = \beta \csc(\phi) \frac{-1 \mu_t}{\xi_t \Gamma_t^2} \quad (3.86)$$

$$\frac{\partial h_{\Delta v}}{\partial \alpha} = 0 \quad (3.87)$$

$$\frac{\partial h_{\Delta v}}{\partial \beta} = \csc(\phi) \left( \frac{\mu_t}{\xi_t \Gamma_t} - \frac{Y}{\xi_1 Z} \right) \quad (3.88)$$

$$\frac{\partial h_{\Delta v}}{\partial \phi} = \beta \cot(\phi) \csc(\phi) \left( \frac{\mu_t}{\xi_t \Gamma_t} - \frac{Y}{\xi_1 Z} \right) \quad (3.89)$$

$$\frac{\partial h_{\Delta v}}{\partial k_1} = \beta \csc(\phi) \left( \frac{r_1^2 Y}{\xi_1^2 Z} - \frac{r_t^2 \mu_t}{\xi_t^2 \Gamma_t} \right) \quad (3.90)$$

$$\frac{\partial h_{\Delta v}}{\partial u_0} = 0 \quad (3.91)$$

$$\frac{\partial h_{\Delta v}}{\partial v_0} = 0 \quad (3.92)$$

A expressão de  $e_{u,t}$ , apresentada na Equação 3.61, fornece os elementos:

$$\frac{\partial h_{e_u}}{\partial X} = 0 \quad (3.93)$$

$$\frac{\partial h_{e_u}}{\partial Y} = 0 \quad (3.94)$$

$$\frac{\partial h_{eu}}{\partial Z} = 0 \quad (3.95)$$

$$\frac{\partial h_{eu}}{\partial t_1} = \frac{\alpha}{t_{3,t}} \quad (3.96)$$

$$\frac{\partial h_{eu}}{\partial t_2} = -\frac{\alpha \cot(\phi)}{t_{3,t}} \quad (3.97)$$

$$\frac{\partial h_{eu}}{\partial t_3} = -\frac{\alpha (t_{1,t} - \cot(\phi)t_{2,t})}{t_{3,t}^2} \quad (3.98)$$

$$\frac{\partial h_{eu}}{\partial \alpha} = \frac{(t_{1,t} - \cot(\phi)t_{2,t})}{t_{3,t}} \quad (3.99)$$

$$\frac{\partial h_{eu}}{\partial \beta} = 0 \quad (3.100)$$

$$\frac{\partial h_{eu}}{\partial \phi} = \frac{\alpha}{t_{3,t}} t_{2,t} \csc^2(\phi) \quad (3.101)$$

$$\frac{\partial h_{eu}}{\partial k_1} = 0 \quad (3.102)$$

$$\frac{\partial h_{eu}}{\partial u_0} = 1 \quad (3.103)$$

$$\frac{\partial h_{eu}}{\partial v_0} = 0 \quad (3.104)$$

E para finalizar,  $e_{v,t}$ , da Equação 3.64, gera os elementos da última linha de  $H_t$ :

$$\frac{\partial h_{ev}}{\partial X} = 0 \quad (3.105)$$

$$\frac{\partial h_{ev}}{\partial Y} = 0 \quad (3.106)$$

$$\frac{\partial h_{ev}}{\partial Z} = 0 \quad (3.107)$$

$$\frac{\partial h_{ev}}{\partial t_1} = 0 \quad (3.108)$$

$$\frac{\partial h_{ev}}{\partial t_2} = \frac{\beta}{\sin(\phi)t_{3,t}} \quad (3.109)$$

$$\frac{\partial h_{ev}}{\partial t_3} = -\frac{\beta t_{2,t}}{\sin(\phi)t_{3,t}^2} \quad (3.110)$$

$$\frac{\partial h_{ev}}{\partial \alpha} = 0 \quad (3.111)$$

$$\frac{\partial h_{ev}}{\partial \beta} = \frac{t_{2,t}}{\sin(\phi)t_{3,t}} \quad (3.112)$$

$$\frac{\partial h_{ev}}{\partial \phi} = -\frac{\beta t_{2,t}}{t_{3,t}} \cot(\phi) \csc(\phi) \quad (3.113)$$

$$\frac{\partial h_{ev}}{\partial k_1} = 0 \quad (3.114)$$

$$\frac{\partial h_{ev}}{\partial u_0} = 0 \quad (3.115)$$

$$\frac{\partial h_{ev}}{\partial v_0} = 1 \quad (3.116)$$

### 3.2.2.6 Considerações finais

Com isso chega-se ao final da seção 3.2. Nela mostrou-se todas as etapas do desenvolvimento do projeto do FKE que realiza a estimação dos parâmetros internos da câmera monocular que é usada na captura das imagens. A Figura 3.2 apresenta um esquemático do algoritmo descrito nesta seção.

Análise em relação ao desempenho do algoritmo, bem como possíveis pontos a serem melhorados, serão apresentados no próximo capítulo.

## 3.3 LOCALIZAÇÃO VISUAL INERCIAL COM AUTO-CALIBRAÇÃO DE CÂMERA (LOVIACC)

A presente seção tem como objetivo apresentar o algoritmo de Localização Visual Inercial com Auto-Calibração de Câmera (LOVIACC) desenvolvido. Este algoritmo tem como base o estudo de auto-calibração de câmera a partir de técnica de filtragem estocástica mostrado na seção anterior. Mas apesar de tê-lo como base, o objetivo final deste algoritmo de LOVIACC não é mais a auto-calibração em si. A estimação dos parâmetros internos da câmera será apenas um dos propósitos do LOVIACC, pois, como o próprio nome diz, trata-se de um sistema de localização a partir dos sensores visual e inercial.

De certa forma, pode-se entender a presente seção como uma evolução ou melhoramento do estudo apresentado na seção 3.2. Esses melhoramentos vêm tanto no sentido de acrescentar novas grandezas a serem estimadas como no sentido de melhorar a estimação das grandezas já estimadas, tornando essas estimações mais confiáveis.

### 3.3.1 Definição do problema

Um agente robótico se locomove livremente no espaço carregando consigo uma câmera monocular e uma UMI. Ambos os sensores estão fixamente acoplados no corpo do agente e possuem uma rotação  $R_{cam}^{umi}$  e uma translação  $t_{cam}^{umi}$  entre si constantes e conhecidas. A câmera captura imagens do ambiente enquanto o robô realiza seus movimentos, a UMI, por sua vez, faz suas leituras de maneira independente da câmera, isto é, sem que haja qualquer sincronia com o momento ou frequência em que ocorrem as capturas de imagens.

A equação matricial em coordenadas homogêneas

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{Z_{cam}} \begin{bmatrix} \alpha & -\alpha \cot(\phi) & u_0 & 0 \\ 0 & \frac{\beta}{\sin(\phi)} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \\ 1 \end{bmatrix} \quad (3.117)$$

já apresentada no capítulo 2 e na seção anterior, relaciona todos os pontos tridimensionais, representados no {CAM}, com suas respectivas projeções no plano da imagem.

A cada nova imagem  $I_t$  capturada, tem-se a possibilidade (com um sistema de odometria visual) de se reconstruir a movimentação desempenhada pela câmera no intervalo de tempo entre as capturas a menos de uma escala  $\lambda$ .

A cada nova leitura dos acelerômetros e giroscópios da UMI tem-se a possibilidade de alimentar um sistema de localização inercial e, conseqüentemente, de se realizar a estimativa da orientação e posição da UMI em relação ao {G}.

Com estes sistemas de localização funcionando partir das leituras dos sensores mencionados, tem-se como objetivo a estimativa da localização do agente em relação a {G}, com grau de confiança maior do que o obtido com os sistemas isolados, isto é, deseja-se, com este algoritmo de LOVIACC, a estimativa de

$$\begin{bmatrix} s_t \\ v_t \\ q_t \\ \lambda \end{bmatrix} \quad (3.118)$$

onde  $s_t$  representa o vetor da posição  $\begin{pmatrix} s_x & s_y & s_z \end{pmatrix}^T$  na qual o agente se encontra no instante de tempo  $t$ ,  $v_t$  é o vetor de velocidade  $\begin{pmatrix} v_x & v_y & v_z \end{pmatrix}^T$  no instante de tempo  $t$ ,  $q_t$  é o quatérnio unitário [32] que corresponde a orientação do agente em relação ao {G}, também no instante de tempo  $t$  e  $\lambda$  é a escala de grandeza na qual foram feitas as estimativas de movimento com o sistema de Odometria Visual monocular utilizado.

Paralelamente a estas estimativas que dizem respeito a localização do robô no espaço, o sistema LOVIACC também deve ser capaz de estimar os parâmetros internos da câmera utilizada, isto é, deve conseguir estimar corretamente

$$\begin{bmatrix} \alpha \\ \beta \\ \phi \\ u_0 \\ v_0 \end{bmatrix} \quad (3.119)$$

que são as variáveis apresentadas na matriz da câmera presente na Equação 3.117.

### 3.3.2 Algoritmo

O algoritmo desenvolvido para solucionar o problema proposto pode ser explicado resumidamente da seguinte forma: Toda nova imagem capturada pela câmera é entregue a um sistema de Odometria Visual que, com todos os seus pré requisitos satisfeitos, consegue estimar a posição e orientação da câmera (e conseqüentemente do agente que a porta) em relação ao {G}.

Paralelamente a captura das imagens pela câmera, a Unidade de Medição Inercial está realizando leituras com seus acelerômetros e giroscópios. Estas leituras são utilizadas juntamente com as estimativas da Odometria Visual como entrada para uma Fusão Sensorial que pretende obter uma estimativa mais confiável da posição e orientação do robô que carrega os sensores em relação ao sistema de coordenadas globais.

Além de estimar as grandezas relativas à posição do robô, o processo de fusão sensorial consegue ainda recuperar a escala  $\lambda$  que falta na estimativa da Odometria Visual. Após ter sido realizada a fusão sensorial das leituras dos sensores, as estimativas mais confiáveis obtidas são utilizadas como um dos parâmetros de entrada do estimador dos parâmetros internos da câmera.

Os valores encontrados destes parâmetros são então entregues ao sistema de Odometria Visual para que o

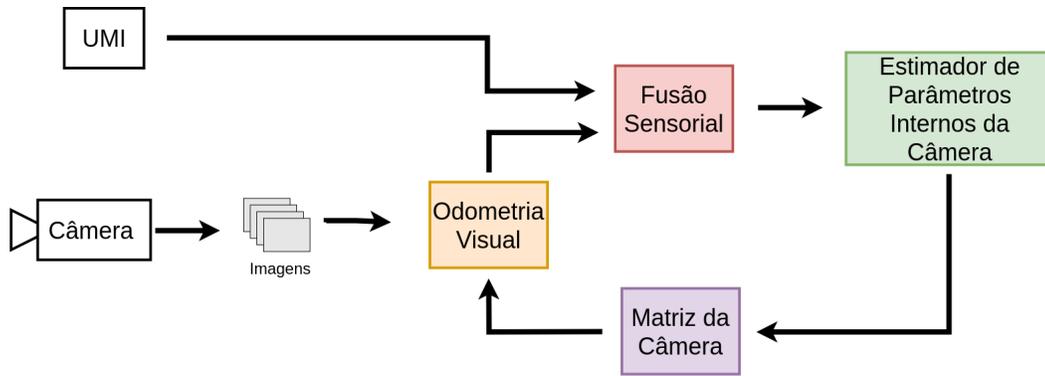


Figura 3.3: Esquemático do algoritmo de LOVIACC proposto.

mesmo consiga resultados ainda mais precisos na sua tarefa de reconstruir a trajetória da câmera.

A Figura 3.3 apresenta uma ilustração do esquema do algoritmo, onde é possível observar, de maneira global e superficial, o fluxo cíclico dos dados capturados pelos sensores, bem como dos dados estimados.

No restante da presente seção serão apresentados, em detalhes, as três principais partes do sistema LOVIACC desenvolvido:

- Odometria Visual;
- Fusão Sensorial;
- Estimador de Parâmetros.

### 3.3.2.1 Odometria Visual

Como apresentado no começo desta seção, o algoritmo de LOVIACC desenvolvido é composto, dentre outras partes, por um sistema independente de Odometria Visual (OV). O capítulo 2 apresentou na seção 2.3 uma possível implementação de OV, esta, por sua vez, foi a implementação escolhida para compor o algoritmo LOVIACC. A Figura 3.4 apresenta, de maneira simplificada, um esquema que ilustra a OV como uma “caixa preta” que recebe dados da câmera (imagens) conjuntamente com a informação de calibração do sensor visual (parâmetros internos que modelam a matriz da câmera) para estimar o vetor de posição  $s$  e o quaternião unitário  $q$ , que representa a orientação do agente robótico em relação ao  $\{G\}$ .

O restante deste tópico será dedicado a rerepresentar o funcionamento desta implementação de Odometria Visual, mas se abstendo de entrar em detalhes muito profundos, pois os mesmos já estão presentes na seção 2.3 do presente trabalho.

A OV implementada como parte do LOVIACC pode ser resumida basicamente da seguinte maneira:

No momento da captura da primeira imagem pela câmera, define-se que a origem do sistema de coordenadas globais está posicionada no meso ponto, e com a mesma orientação, do sistema de coordenadas da câmera. Tal medida faz com que seja menos custoso encontrar, a cada nova imagem capturada, a posição e orientação da câmera (e consequentemente do robô).

Voltando à primeira imagem capturada,  $I_1$ , nela se aplica um detector de pontos salientes, como o SURF[43].

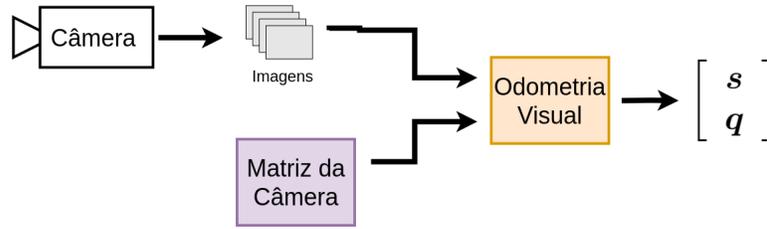


Figura 3.4: Esquemático de fluxo de dados do sistema de Odometria Visual utilizado no LOVIACC proposto.

Em seguida, a câmera (e o robô) estão livres para se movimentar livremente no espaço. Assim que a próxima imagem é adquirida, rastreia-se os pontos salientes encontrados na imagem anterior.

Com as coordenadas dos pontos presentes em ambas as imagens, é possível, com o auxílio da geometria epipolar, calcular a chamada Matriz Essencial. Esta matriz é um importante ente que traz em sua definição formal a constante epipolar, já mostrada no capítulo 2:

$$\hat{p}'^T \mathbf{E} \hat{p} = 0 \quad (3.120)$$

Esta definição matemática da matriz  $\mathbf{E}$  é a maneira pela qual o sistema de Odometria Visual a calcula numericamente, como já discutido na seção 2.3.

A Matriz Essencial  $\mathbf{E}$  pode ainda ser entendida pela sua definição matemática que a relaciona com a transformação de corpo rígido que caracteriza a movimentação desempenhada pela câmera entre as capturas dos *frames* em análise, isto é:

$$\mathbf{E} = [\mathbf{t}]_{\times} \mathbf{R} \quad (3.121)$$

Sendo assim, tem-se que a partir da Matriz Essencial encontrada é possível caracterizar a movimentação da câmera entre as capturas com um vetor de translação  $\mathbf{t}$  e com uma matriz de rotação  $\mathbf{R}$ .

Com a chegada de uma nova imagem capturada  $I_t$ , tem-se que o processo se repete, encontrando-se assim, uma sequência de translações e rotações desempenhadas pela câmera que podem ser concatenadas para se descobrir a posição e orientação do sensor visual (e do robô) em relação ao  $\{G\}$ .

É importante ressaltar, mais uma vez, que este processo é capaz de reconstruir a translação desempenhada pela câmera entre as capturas a menos de uma constante  $\lambda$ . Isso se dá pelo fato de se estar utilizando uma câmera monocular, o que faz com que se perca a noção de profundidade dos objetos com a projeção realizada pela câmera.

Esta escala  $\lambda$  pode ser encontrada se, em conjunto com a OV, existir outro sensor, ou conjunto de sensores, desempenhando papel equivalente de se estimar a movimentação do agente robótico. No caso do sistema LOVIACC que está sendo descrito, existe um sistema de Navegação Inercial em paralelo à OV. Este SNI é utilizado como um “lastro” para se recuperar a escala  $\lambda$ . O procedimento de recuperação desta grandeza se dará na etapa de fusão sensorial do algoritmo, e será melhor detalhado adiante.

### 3.3.2.2 Filtro de Kalman Unscented

O Filtro de Kalman (FK) [51] é uma importante ferramenta estocástica. É amplamente utilizada nas mais diversas áreas, inclusive em diversas abordagens de algoritmos da robótica. O único problema é que o filtro funciona somente com sistemas lineares, pois os mesmos realizam apenas transformações lineares sobre as variáveis de estado em suas funções de propagação de estados e medição, o que faz com que o resultado dessas transformações sejam também distribuições gaussianas.

Porém, existem diversos casos em que um sistema não consegue ser modelado linearmente. Nesse tipo de sistema, não é possível aplicar o Filtro de Kalman. Para sistemas não lineares, é necessário realizar uma espécie de linearização das transformações não lineares que são aplicadas às variáveis de estado para que o resultado das transformações seja também uma gaussiana. Nesse contexto, surgem algumas variantes do FK que servem justamente para expandir a gama de sistemas que podem usufruir das técnicas de filtragem que surgem com o FK.

Talvez a técnica mais famosa e implementada na literatura das pesquisas relacionadas a robótica móvel seja o Filtro de Kalman Estendido [7], que foi inclusive a técnica utilizada para se linearizar o sistema de Auto-Calibração de Câmera apresentado na seção anterior. O FKE, como já visto, realiza as linearizações necessárias para desempenhar a filtragem a partir de expansões de série de Taylor. Para que seja realizada esta expansão é necessário, também como já mostrado anteriormente, que se calcule uma série de Jacobianos, que podem se mostrar uma tarefa realmente trabalhosa e custosa, especialmente se o vetor de estados do sistema for muito grande.

Só que o FKE não é a única maneira de se realizar uma linearização das transformações de gaussianas. Como alternativa ao FKE, tem-se o chamado Filtro de Kalman Unscented (FKU) [8]. O FKU realiza a linearização estocástica a partir de uma regressão linear ponderada e apresenta resultados de linearização iguais ou superiores aos conseguidos com o FKE. Outra grande vantagem de se utilizar o FKU está no fato de não ser necessário calcular nenhum Jacobiano. Isso torna a sua implementação muito mais simples, especialmente para sistemas com um grande número de variáveis de estado. Claro que existe um lado negativo em se optar pelo FKU. Computacionalmente falando, o algoritmo é mais custoso do que o FKE.

O restante do presente tópico será dedicado à apresentação do algoritmo de implementação do FKU, que pode tanto ser utilizado nas mesmas situações de um FKE ou de um FK, em tópicos futuros serão apresentadas situações no algoritmo de LOVIACC em que o FKU é utilizado como uma ferramenta para se realizar a estimação de parâmetros, mas também serão apresentadas situações em que o FKU aparece como técnica de fusão sensorial.

Tem-se um sistema não linear:

$$x_t = f(x_{t-1}) + \varepsilon_{x,t-1} \quad (3.122)$$

$$y_t = h(x_t) + \varepsilon_{y,t} \quad (3.123)$$

Onde  $x_t$  é o vetor de estados do sistema,  $f$  é a função de propagação de estados,  $y_t$  é o vetor de medições e  $h$  é a função de medições do sistema. As variáveis  $\varepsilon_x$  e  $\varepsilon_y$  são os ruídos gaussianos brancos de média nula que atingem, respectivamente, a função de propagação de estados, com matriz de covariância  $R$ , e a função de medição, com matriz de covariância  $Q$ . Utiliza-se o FKU da seguinte maneira: Para cada instante de tempo  $t$ , deseja-se obter uma estimativa para o vetor de estados representada por uma gaussiana de mesma dimensão  $n$  que o vetor de estados com média  $\mu_t$  e matriz de covariância  $\Sigma_t$  a partir da estimativa anterior  $\mu_{t-1}$  e das

medições atuais contidas no vetor  $y_t$ . Essas estimativas são obtidas em etapas. Primeiro realiza-se a predição a priori e em seguida a estimativa a posteriori, como será detalhado a seguir.

O primeiro passo consiste em obter a linearização da aplicação da função  $f$  sobre o vetor de estados para que se obtenha uma gaussiana como estimativa a priori do FKU. Essa linearização se dá por meio da utilização de pontos sigma.

A maneira de se calcular os pontos sigma consiste em selecionar  $2n + 1$  pontos da gaussiana, onde  $n$  é a dimensão do vetor de estados (e conseqüentemente da gaussiana que se pretende obter ao final da estimação), de forma que:

$$\mathcal{X}_{t-1}^{[0]} = \mu_{t-1} \quad (3.124)$$

$$\mathcal{X}_{t-1}^{[i]} = \mu_{t-1} + \left( \sqrt{(n + \tau) \Sigma_{t-1}} \right)_i \quad \text{para } i = 1, \dots, n \quad (3.125)$$

$$\mathcal{X}_{t-1}^{[i]} = \mu_{t-1} - \left( \sqrt{(n + \tau) \Sigma_{t-1}} \right)_{i-n} \quad \text{para } i = n + 1, \dots, 2n \quad (3.126)$$

onde  $\tau = \rho^2 (n + \kappa) - n$ , com  $\rho$  e  $\kappa$  sendo parâmetros que determinam quão próximos ou distantes são os pontos sigma uns dos outros em relação à média  $\mu$  na distribuição gaussiana. Para cada ponto sigma  $\mathcal{X}_{t-1}^{[i]}$  encontrado existem dois pesos associados a ele. Um deles,  $\omega_m^{[i]}$ , é utilizado para se computar a influência do ponto sigma na média da estimativa, e o outro peso,  $\omega_c^{[i]}$ , é utilizado para se computar a influência do ponto sigma na matriz de covariância da estimativa. Os pesos são calculados da seguinte forma:

$$\omega_m^{[0]} = \frac{\tau}{n + \tau} \quad (3.127)$$

$$\omega_c^{[0]} = \frac{\tau}{n + \tau} + (1 - \rho^2 + 2) \quad (3.128)$$

$$\omega_m^{[i]} = \omega_c^{[i]} = \frac{1}{2(n + \tau)} \quad \text{para } i = 1, \dots, 2n. \quad (3.129)$$

Uma vez tendo sido encontrados todos os pontos sigma e seus respectivos pesos, os pontos sigma passam pela função  $f$  e, a partir dos resultados, calcula-se a média  $\bar{\mu}_t$  e a covariância  $\bar{\Sigma}_t$  ponderadas destes resultados como sendo a estimativa a priori. Matematicamente:

$$\bar{\mathcal{X}}_t^* = f(\mathcal{X}_{t-1}) \quad (3.130)$$

$$\bar{\mu}_t = \sum_{i=0}^{2n} \omega_m^{[i]} \bar{\mathcal{X}}_t^{[i]*} \quad (3.131)$$

$$\bar{\Sigma}_t = \sum_{i=0}^{2n} \omega_c^{[i]} \left( \bar{\mathcal{X}}_t^{[i]*} - \bar{\mu}_t \right) \left( \bar{\mathcal{X}}_t^{[i]*} - \bar{\mu}_t \right)^T + R. \quad (3.132)$$

Uma vez tendo sido encontrada a estimativa a priori, uma gaussiana de dimensão  $n$ , média  $\bar{\mu}_t$  e matriz de covariância  $\bar{\Sigma}_t$ , deseja-se encontrar a estimativa a posteriori. Para isso, é necessário encontrar outros  $2n + 1$  pontos sigma na distribuição a priori, de forma que:

$$\bar{\mathcal{X}}_t^{[0]} = \bar{\mu}_{t-1} \quad (3.133)$$

$$\bar{\mathcal{X}}_t^{[i]} = \bar{\mu}_{t-1} + \left( \sqrt{(n+\tau)\bar{\Sigma}_{t-1}} \right)_i \quad \text{para } i = 1, \dots, n \quad (3.134)$$

$$\bar{\mathcal{X}}_t^{[i]} = \bar{\mu}_{t-1} - \left( \sqrt{(n+\tau)\bar{\Sigma}_{t-1}} \right)_{i-n} \quad \text{para } i = n+1, \dots, 2n. \quad (3.135)$$

Estes pontos sigma  $\bar{\mathcal{X}}_t^{[i]}$  possuem os mesmos pesos  $\omega_m^{[i]}$  e  $\omega_c^{[i]}$  calculados anteriormente e são aplicadas a função de medição  $h$  a fim de se obter uma gaussiana com média  $\hat{z}_t$  e matriz de covariância  $S_t$ . Matematicamente:

$$\bar{\mathcal{Z}}_t = h(\bar{\mathcal{X}}_t) \quad (3.136)$$

$$\hat{z}_t = \sum_{i=0}^{2n} \omega_m^{[i]} \bar{\mathcal{Z}}_t^{[i]} \quad (3.137)$$

$$\bar{\Sigma}_t = \sum_{i=0}^{2n} \omega_c^{[i]} \left( \bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t \right) \left( \bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t \right)^T + Q. \quad (3.138)$$

Para finalizar o ciclo de estimação a posteriori, é necessário calcular a matriz de correlação cruzada entre o estado e as observações:

$$\bar{\Sigma}_t^{x,z} = \sum_{i=0}^{2n} \omega_c^{[i]} \left( \bar{\mathcal{X}}_t^{[i]} - \bar{\mu}_t \right) \left( \bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t \right)^T \quad (3.139)$$

A partir dessa matriz pode-se calcular o ganho de kalman  $K_t$ :

$$K_t = \bar{\Sigma}_t^{x,z} S_t^{-1}. \quad (3.140)$$

E finalmente, pode-se encontrar a distribuição gaussiana que representa a estimativa a posteriori do Filtro de Kalman Unscented. Trata-se de uma distribuição com média  $\mu_t$  e matriz de covariância  $\Sigma_t$  calculados como:

$$\mu_t = \bar{\mu}_t + K_t (z_t - \hat{z}_t) \quad (3.141)$$

$$\Sigma_t = \bar{\Sigma}_t - K_t S_t K_t^T. \quad (3.142)$$

Com isso conclui-se o ciclo de estimativas do FKU. Com a chegada de novas medições, pode-se efetuar todos os passos novamente para realizar uma nova filtragem. Mais uma vez é importante ressaltar que este FKU será utilizado no algoritmo de LOVIACC como técnica de fusão sensorial e como técnica de estimação de parâmetros, como será mostrado nos tópicos a seguir.

### 3.3.2.3 Fusão Sensorial da Odometria Visual com Navegação Inercial

O começo do presente capítulo trouxe mais uma vez as vantagens de se utilizar as informações da Odometria Visual em conjunto com as estimativas de um Sistema de Navegação Inercial. Ambos os sistemas apresentam seus pontos fortes e fracos, mas fundindo as estimativas de ambos, pode-se aproveitar o que cada um tem de melhor a oferecer.

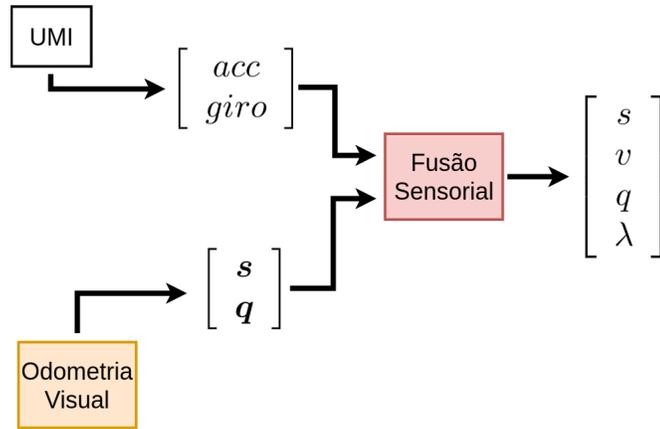


Figura 3.5: Esquemático da fusão sensorial entre a Odometria Visual e as leituras da UMI.

O resultado de uma fusão sensorial entre os dois sistemas é uma estimativa mais confiável a cerca da orientação e posição do agente robótico que porta os dois sensores. Como já mencionado anteriormente, o sistema LOVIACC desenvolvido apresenta como componente uma etapa de fusão sensorial entre a OV e as leituras da UMI.

A Figura 3.5 apresenta uma ilustração do fluxo de dados no que diz respeito à fusão sensorial do algoritmo de LOVIACC desenvolvido. Como pode-se perceber, o algoritmo da fusão pode ser resumindo como o processo que recebe da OV as estimativas do vetor de posição  $s$  e quatérnio unitário  $q$ , que representa a orientação do robô no espaço, e recebe também como entrada as leituras dos acelerômetros e giroscópios da UMI. A saída do processo de fusão consiste em uma estimativa mais confiável da pose do robô em relação ao  $\{G\}$ , bem como da escala  $\lambda$  que corrige as estimativas da Odometria Visual.

Esta fusão sensorial é realizada com a utilização de um FKU. O restante do presente tópico será dedicado a apresentar o filtro que foi projetado para a fusão.

## Vetor de Estados

Assim como no caso de um Filtro de Kalman Estendido, o FKU apresenta, a cada novo instante de tempo  $t$ , uma estimativa dos elementos do vetor de estados a partir do modelo de predição e em conjunto com as correções realizadas com o vetor de medições.

No caso do FKU projetado, o vetor de estados do sistema é dado por:

$$x_t = \begin{bmatrix} s_t \\ v_t \\ q_t \\ \lambda \end{bmatrix} \quad (3.143)$$

e cada uma das variáveis presentes no vetor serão melhor detalhadas a seguir.

O vetor  $s_t$  representa o ponto 3D,  $(s_x \ s_y \ s_z)^T$ , no espaço no qual se encontra o agente robótico rastreado em relação ao sistema de coordenadas globais no instante de tempo  $t$ . O vetor  $v_t$  corresponde ao vetor de velocidade,  $(v_x \ v_y \ v_z)^T$ , do robô também no instante de tempo  $t$ .

Já o quatérnio unitário  $q_t$  representa a orientação do robô em relação ao  $\{G\}$  no instante de tempo  $t$ .  $q_t$  é representado na forma  $\begin{bmatrix} \zeta & \varphi^T \end{bmatrix}^T$  e é chamado de quatérnio unitário pois  $\zeta^2 + \varphi \cdot \varphi = 1$  [31].

E para finalizar a apresentação das grandezas presentes no vetor de estados,  $\lambda$  representa a escala na qual se encontram as estimativas da Odometria Visual para a posição do robô. Como já discutido no capítulo 2, um sistema de OV com uma câmera monocular é incapaz de reconstruir, sozinho, a escala na qual são realizadas suas estimativas.  $\lambda$  pode ser facilmente entendido como uma espécie de “constante conversora” das unidades estimadas com a OV para os sistema métrico.

Com isso, todos os elementos do vetor de estados do FKU da fusão sensorial já estão devidamente descritos. O próximo tópico irá tratar a respeito das funções utilizadas para serem feitas as predições a priori do FKU projetado.

## Função de Propagação de Estados

Como já apresentado na seção 2.4 do capítulo 2, um filtro de Kalman pode ser utilizado para realizar uma fusão sensorial. Uma das abordagens possíveis é fazer com que um dos sensores seja responsável por atualizar a estimativa a priori e o outro seja responsável pela medição, sendo assim, quando se calcular a estimativa a posteriori, o ganho de Kalman será o responsável por fazer a ponderação adequada entre os dois sensores para que se obtenha a melhor estimativa da fusão.

A função de propagação de estados, ou simplesmente função de predição, é a responsável por realizar a estimativa a priori, isto é, ela encontra a estimativa do vetor de estados a partir do estado anterior,  $x_{t-1}$ , em conjunto com as variáveis de entrada do sistema,  $u_t$ . Em termos matemáticos, tem-se que a função  $f$  de propagação de estados apresenta a forma:

$$x_t = f(x_{t-1}, u_t) + \varepsilon_{x,t-1} \quad (3.144)$$

Isto é, para cada um dos elementos do vetor de estado, pode-se modelar uma função que prevê a evolução das variáveis. Como o FKU que está sendo apresentado é responsável por realizar fusão sensorial entre os dados lidos pela UMI e as estimativas de um sistema de Odometria Visual, optou-se por deixar a UMI como sendo responsável pela etapa de predição do FKU. Em outras palavras, sempre que forem lidos novos dados pela UMI, será executada a etapa de predição do FKU.

Como a Equação 3.143 apresenta, os elementos do vetor de estado são:  $s_t, v_t, q_t$  e  $\lambda$ . A partir de agora, cada um deles terá sua predição a partir das leituras da UMI detalhada.

O vetor de entradas, ou controle, do sistema,  $u_t$ , apresenta as leituras dos sensores inerciais no instante de tempo  $t$ . O vetor é descrito matematicamente como:

$$u_t = \begin{bmatrix} \gamma \\ \Omega \end{bmatrix} \quad (3.145)$$

onde  $\gamma$  está relacionado com as leituras dos acelerômetros e  $\Omega$  está relacionado às medidas dos giroscópios. Ambas as variáveis serão melhor detalhadas ao longo do presente tópico.

A predição da variável de estado  $s_t$  pode ser realizada a partir do estado anterior,  $x_{t-1}$ , da seguinte forma:

$$s_t = s_{t-1} + T v_{t-1} + \frac{1}{2} T^2 (R_{t-1} (\gamma + \varepsilon_\gamma) - g) + \frac{1}{2} T^2 \varepsilon_a. \quad (3.146)$$

Onde  $T$  é o tempo decorrido, em segundos, entre as duas últimas leituras da UMI.  $\gamma$  corresponde às medidas de aceleração  $\begin{pmatrix} a_x & a_y & a_z \end{pmatrix}^T$  adquiridas pelos acelerômetros da UMI em relação ao  $\{G\}$ .  $g$  é a aceleração gravitacional.  $\varepsilon_\gamma$  é o ruído gaussiano branco associado à leitura dos acelerômetros e  $\varepsilon_a$  é o ruído gaussiano branco de média nula associado à propagação de estados. Já a matriz  $R_{t-1}$  é a matriz de rotação (que é facilmente obtida a partir de um quatérnio unitário [32]) que corresponde a orientação do robô no instante  $t - 1$ .

Com isso, percebe-se que o estado de  $s_t$  consegue ser predito facilmente a cada nova leitura da UMI. De maneira análoga, a variável de estado  $v_t$  também pode ser obtida seguindo a mesma lógica. A função de predição para esta variável de estado é:

$$v_t = v_{t-1} + T (R_{t-1} (\gamma + \varepsilon_\gamma) - g) + T \varepsilon_a. \quad (3.147)$$

As Equações 3.146 e 3.147 mostram que é possível, a partir do estado anterior  $x_{t-1}$  e das leituras da UMI, realizar a estimação a priori das grandezas relacionadas a posição e velocidade do robô.

Para a variável de estado  $q_t$  a estimação a priori é realizada de uma maneira um pouco menos direta. Como a frequência de leitura da UMI é relativamente alta (cerca de 200Hz), é razoável se assumir que a movimentação angular do agente robótico percebida entre duas leituras seja pequena. Sendo assim, é aceitável realizar uma aproximação do quatérnio unitário[31] para que o mesmo seja denotado por suas duas componentes,  $\zeta$  e  $\varphi$ , como funções de  $\delta_{\theta,t}$ :

$$\zeta_t = \zeta_{t-1} - \frac{1}{2} \varphi_{t-1}^T \delta_{\theta,t}, \quad (3.148)$$

$$\varphi_t = \varphi_{t-1} + \frac{1}{2} \zeta_{t-1} \delta_{\theta,t} - \frac{1}{2} \varphi_{t-1} \times \delta_{\theta,t} \quad (3.149)$$

onde  $\delta_{\theta,t} = \begin{bmatrix} \delta_{\theta,x,t} & \delta_{\theta,y,t} & \delta_{\theta,z,t} \end{bmatrix}^T$  corresponde aos ângulos da rotação realizada em relação a cada um dos eixos  $x$ ,  $y$  e  $z$  do  $\{G\}$  percebida pelos giroscópios da UMI.

Sendo assim, só é necessário rastrear o comportamento de  $\delta_{\theta,t}$  para se realizar a predição do estado de  $q_t$ . Para isto, pode-se utilizar a leitura da velocidade angular  $\Omega = \begin{pmatrix} \omega_x & \omega_y & \omega_z \end{pmatrix}^T$  adquirida pela UMI e verificar que:

$$\delta_{\theta,t} = T (\Omega + \varepsilon_\Omega) + T \varepsilon_\omega. \quad (3.150)$$

Onde  $\varepsilon_\Omega$  é o ruído gaussiano branco associado as leituras dos giroscópios da UMI e  $\varepsilon_\omega$  é o ruído gaussiano branco que influencia da propagação do estado da função de predição.

Para finalizar todas as variáveis de estado, resta analisar o caso de  $\lambda$ . Tem-se nesta variável uma grandeza que não é observada diretamente por leitura alguma da UMI, restando a esta variável ser estimada a partir da medição (a exemplo do vetor de estados do FKE da seção 3.2). Desta forma, a estimativa a priori de  $\lambda$  será realizada da maneira mais simples possível: mantendo sua estimativa anterior. Matematicamente, tem-se que:

$$\lambda_t = \lambda_{t-1} + \varepsilon_\lambda \quad (3.151)$$

onde  $\varepsilon_\lambda$  é o ruído gaussiano branco que perturba a propagação de estado.

Com isso, conclui-se a apresentação de todas as funções de predição de todas as variáveis de estado. Como pode-se perceber, elas são atualizadas mediante leitura dos sensores da UMI. Ressalta-se assim que a predição do FKU projetado está atrelada às leituras dos sensores inerciais.

## Medição

Dessa forma, chega-se à etapa em que se descreve o processo de medição do FKU. Enquanto a etapa de predição foi destinada a descrever como os sensores inerciais fazem sua contribuição para a estimativa, a etapa de medição será aquela na qual o sensor visual, através dos resultados da Odometria Visual, faz sua contribuição.

A função de medição do Filtro de Kalman Unscented,  $h$ , é basicamente a função que consegue modelar a os elementos do vetor de medições,  $y_t$ , como função dos elementos do vetor de estados,  $x_t$ . Matematicamente:

$$y_t = h(x_t) + \varepsilon_y \quad (3.152)$$

onde  $\varepsilon_y$  corresponde ao ruído gaussiano branco que perturba o vetor de medições.

No caso do FKU projetado, o vetor de medições é composto pela estimativa de posição e da orientação obtidas pela câmera na Odometria Visual. Ou seja:

$$y_t = \begin{bmatrix} s_{ov,t} \\ q_{ov,t} \end{bmatrix}. \quad (3.153)$$

Talvez o mais importante neste momento seja perceber que a estimativa da pose do robô obtida pela Odometria Visual seja um pouco diferente da obtida pelos sensores inerciais da UMI. A OV faz suas estimativas em relação a posição inicial da câmera, que no caso foi estipulada para ser a origem do sistema de coordenadas globais, como mencionado no tópico 3.3.2.1. Já o sistema de navegação inercial que utiliza a UMI como entrada, embora faça suas estimativas também em relação ao  $\{G\}$ , faz suas estimativas de movimentação centradas no eixo de referências da UMI.

O FKU projetado selecionou a referência das estimativas de posição do SNI como sendo a referência da pose do robô representada no vetor de estados. Essa escolha se deu de maneira que viesse a minimizar o número de conversões de coordenadas realizadas ao longo da implementação do fusão. Dessa forma, a função de medição  $h$  tem como tarefa ser capaz de converter essas coordenadas (entre as que centram a UMI e as que centram a origem da câmera) bem como evidenciar a única variável de estado que necessita das estimativas da OV para poder ser estimada:  $\lambda$ . Logo, pode-se enunciar a função de medição de  $s_{ov,t}$  como:

$$s_{ov,t} = \lambda R_{umi}^{cam}(s_t) + t_{umi}^{cam} \quad (3.154)$$

onde a matriz  $R_{umi}^{cam}$  corresponde a rotação que existe entre a UMI e a câmera e o vetor  $t_{umi}^{cam}$  corresponde à translação existente entre os dois sensores. Válido lembrar que ambas essas medidas são fixas e conhecidas.

Já a medida de  $q_{ov,t}$  pode ser modelada como:



Figura 3.6: Representação gráfica da linha do tempo mostrando possíveis momentos em que ocorrem capturas de novas imagens pela Câmera ou leituras dos sensores da UMI.

$$\mathbf{q}_{ov,t} = \mathbf{q}_{umi}^{cam} \odot \mathbf{q}_t \quad (3.155)$$

onde o quatérnio  $\mathbf{q}_{umi}^{cam}$  representa a rotação que existe entre a UMI e a câmera e a operação  $\odot$  representa a multiplicação de quatérnios [32].

### Considerações finais

Assim chega-se ao ponto em que todos os elementos necessários para a implementação do Filtro de Kalman Unscented que vai realizar a fusão sensorial estão devidamente descritos. Basta seguir o algoritmo descrito no tópico 3.3.2.2 e será possível obter como resultado a estimativa da fusão entre a Odometria Visual e as leituras dos acelerômetros e giroscópios da UMI, bem como a estimativa da escala  $\lambda$  da estimativa da OV.

Existe ainda uma importante observação a ser feita. No tópico 3.3.1 foi dito que não existe qualquer necessidade de sincronia entre a captura de imagens por parte da câmera com a leitura dos sensores inerciais da UMI. Além disso, foi tomado o devido cuidado para que as etapas de predição e medição do FKU projetado não misturassem leituras da UMI com estimativas da OV, isto é, deixou-se a trabalho da etapa de predição lidar com as leituras da UMI enquanto a etapa de medição lida com a OV. Assim, tem-se uma situação na qual pode-se executar a etapa de predição toda vez que uma nova leitura da UMI estiver disponível, e executar a etapa de medição sempre que uma nova estimativa da Odometria Visual estiver disponível.

Como a OV depende de novas capturas de imagens por parte da câmera para poder realizar estimativas, tem-se que as mesmas possuem frequência de atualização limitada à taxa de captura da câmera. As taxas de leitura da UMI e da câmera são, na maioria das vezes, diferentes. A UMI apresenta uma frequência de atualização muito maior que a da câmera, desse modo, serão realizadas muito mais predições do FKU do que medições.

A Figura 3.6 apresenta uma ilustração que apresenta uma espécie de linha do tempo, na qual é mostrada possíveis momentos em que são realizadas novas leituras dos sensores inerciais da UMI e também os momentos em que chegam novas imagens vindas da câmera. Dessa forma é possível observar que ocorrem mais predições que medições do FKU.

Até aqui foi apresentado o algoritmo de fusão sensorial que é responsável pela estimativa da pose do agente robótico rastreado, ou seja, tem-se, até o presente momento, um sistema de Localização Visual Inercial. Para que o mesmo se torne um LOVIACC ainda falta a parte responsável pela estimativa dos parâmetros internos da câmera. O próximo tópico irá tratar justamente desta parte final do algoritmo desenvolvido.

#### 3.3.2.4 Estimador de Parâmetros Internos da Câmera

Na seção 3.2 foi apresentada uma primeira tentativa de se projetar um sistema de auto-calibração de câmeras monoculares a partir de informações contidas nas imagens capturadas e com o auxílio das leituras dos giroscó-

pios de uma UMI. Ou seja, um sistema que não precisasse de qualquer preparação ou conhecimento prévio do ambiente na qual a câmera está inserida e que fosse executado de maneira *online*, permitindo ao sistema ter conhecimento em tempo real a cerca dos valores dos parâmetros da câmera, sem precisar tomá-los como constantes.

O presente tópico irá apresentar a o que seria uma evolução da técnica desenvolvida naquele momento. Como já mencionado anteriormente, ela não tem mais o objetivo da auto-calibração por si só, mas sim como parte de um sistema mais complexo, um sistema de LOVIACC.

Sendo assim, é válido dizer que o estimador de parâmetros internos da câmera que será descrito apresenta muitas semelhanças com o descrito na seção anterior, porém, sempre que houver diferença, ela será ressaltada.

De início, uma diferença já fica evidente com o que já foi apresentado até o momento. Não mais será utilizado um FKE para realizar a estimação, mas sim um FKU, isso para se obter uma estimativa mais confiável da linearização das não linearidades envolvidas no processo de calibração, mas também para evitar a necessidade de se calcular os Jacobianos que o FKE exige.

Nos próximos tópicos serão apresentados todos os componentes necessários para se implementar o FKU do estimador, ou seja, a definição das variáveis de estado, as funções de predição, o vetor e as funções de medição do sistema.

## Vetor de Estados

Assim como no caso do algoritmo apresentado na seção 3.2, o estimador de parâmetros internos da câmera do sistema de LOVIACC tem como o objetivo a estimação dos valores que compõem a matriz da câmera

$$\mathbf{K} = \begin{bmatrix} \alpha & -\alpha \cot(\phi) & u_0 \\ 0 & \frac{\beta}{\sin(\phi)} & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.156)$$

E como o FKU consegue a cada ciclo de etapa de predição e medição a atualização dos elementos do vetor de estados, optou-se por projetar este FKU com um vetor de estados com a seguinte composição:

$$x_t = \begin{bmatrix} \alpha \\ \beta \\ \phi \\ u_0 \\ v_0 \end{bmatrix}. \quad (3.157)$$

Logo se observa que todos estes elementos estão inclusos também no vetor de estados do FKE da seção anterior apresentado na Equação 3.21. Mas também nota-se que vários elementos presentes no estimador anterior não estão mais listados como variáveis de estado deste novo estimador. Isso se dá principalmente porque as variáveis que diziam respeito a posição do agente robótico não precisam mais ser estimadas, uma vez que elas são medidas e obtidas através da fusão sensorial descrita no tópico 3.3.2.3, que é executado paralelamente ao FKU estimador dos parâmetros. Para se ter uma ideia mais ampla a cerca da temporalidade da execução do estimador com a fusão sensorial, sempre que uma nova imagem fica disponível (como ilustrado na Figura 3.6) além de se executar a medição do FKU da fusão, é também executado todo o ciclo do FKU do

estimador de parâmetros.

As demais variáveis de estado que estavam presentes no vetor do FKE e não estão mais presentes neste novo vetor de estimações foram retiradas por alterações na modelagem do problema, como será evidenciado nos tópicos seguintes.

## Função de Propagação de Estados

A função de propagação de estados tem como tarefa realizar a predição do estado  $x_t$  a partir do estado anterior  $x_{t-1}$ . No caso das variáveis de estado apresentadas na Equação 3.157, ou seja,  $\alpha, \beta, \phi, u_0$  e  $v_0$ , tem-se que todas elas são, idealmente, constantes. Como apresentado no capítulo 2, as alterações nos valores dos parâmetros internos são causadas por choques mecânicos, trepidações ou qualquer outra avaria que a câmera venha a sofrer no período em que o robô se movimenta.

Desta forma, a função de predição  $f$  do FKU do estimador de parâmetros pode ser expressa, matematicamente, como:

$$x_t = x_{t-1} + \varepsilon_x \quad (3.158)$$

onde  $\varepsilon_x$  corresponde ao ruído gaussiano branco de média nula que atinge a propagação de estados.

Outra justificativa para este modelo de predição se justifica pelo fato de ser praticamente impossível modelar matematicamente a forma e os instantes pelos quais os parâmetros se alteram, uma vez que sua mudança está atrelada a eventuais colisões mecânicas que a câmera venha a sofrer a depender do percurso que o robô que a carrega desempenha.

## Medição

Antes de apresentar o modelo de medição adotado pelo FKU estimador de parâmetros é necessário primeiro conhecer qual é o vetor de medições que será utilizado para auxiliar na estimação. Aqui o leitor perceberá algumas das mudanças mais drásticas em relação ao estimador baseado no FKE apresentado na seção anterior.

O vetor de medições do FKU estimador é dado por:

$$y_t = \begin{bmatrix} u_{1,t} \\ v_{1,t} \\ u_{2,t} \\ v_{2,t} \\ \vdots \\ u_{n,t} \\ v_{n,t} \\ e_{u,t} \\ e_{v,t} \end{bmatrix}. \quad (3.159)$$

Só de olhar já da pra perceber que seu tamanho é variável e que ele possui  $2n + 2$  elementos. Isto é, a depender do instante de tempo em que o ciclo de estimação está sendo executado, a quantidade de elementos

presentes no vetor de medições poderá ser diferente. Cada par de elementos  $u_{i,t}$  e  $v_{i,t}$ , presentes no vetor de medição, correspondem às coordenadas da imagem nas quais o  $i$ -ésimo ponto de interesse rastreado se encontra no instante de tempo  $t$ . Já as medidas  $e_{u,t}$  e  $e_{v,t}$  correspondem às coordenadas na imagem de onde se encontra a epípola no instante de tempo  $t$ .

No caso do FKE apresentado na seção anterior, existia um número fixo de medidas realizadas pois se rastreava apenas um único ponto entre as imagens. Com isto, calculava-se sempre o deslocamento da posição deste ponto em relação a sua posição de origem na primeira imagem, isto é, calculava-se  $\Delta u_t$  e  $\Delta v_t$ .

Mas no presente estimador a abordagem foi alterada. Ao invés de sempre se rastrear um único ponto que precisa aparecer em todas as imagens desde a primeira captura realizada pela câmera, rastreia-se todos os pontos que sejam comuns a duas imagens capturadas consecutivamente. Toma-se como medidas, para o vetor de medições, as coordenadas na imagem nas quais os pontos rastreados estão presentes na imagem mais recente.

Em outras palavras, os  $2n$  primeiros elementos do vetor de medições  $y_t$  do FKU estimador correspondem as coordenadas  $(u, v)$  de todos os pontos da imagem  $I_t$  que também estejam presentes na imagem  $I_{t-1}$ .

Tendo sido apresentado o vetor de medições, é necessário agora apresentar como os elementos do mesmo podem ser modelados como funções do vetor de estados  $x_t$ .

A equação matricial em coordenadas homogêneas

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{Z_{cam}} \begin{bmatrix} \alpha & -\alpha \cot(\phi) & u_0 & 0 \\ 0 & \frac{\beta}{\sin(\phi)} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \\ 1 \end{bmatrix}, \quad (3.160)$$

tão utilizada ao longo deste trabalho, é a responsável por modelar a forma com a qual se dá a projeção de um ponto 3D do espaço, representado no {CAM}, em um ponto  $(u, v)$  da imagem. É com esta equação que, mais uma vez, será realizada a modelagem dos elementos do vetor de medição.

Para o caso dos elementos correspondentes a coordenada  $u$  dos pontos na imagem, isto é, para todos os elementos identificados com  $u_{i,t}$  no vetor  $y_t$ , tem-se que pode-se enunciá-lo matematicamente como:

$$u_{i,t} = \frac{1}{Z_{i,t}} \alpha (X_{i,t} - \cot(\phi) Y_{i,t}) + u_0. \quad (3.161)$$

E para todos os elementos correspondentes a coordenada  $v$  dos pontos na imagem, isto é, para todos os elementos identificados com  $v_{i,t}$  no vetor  $y_t$ , tem-se que pode-se enunciá-lo matematicamente como:

$$v_{i,t} = \frac{1}{Z_{i,t}} \frac{\beta}{\sin(\phi)} Y_{i,t} + v_0. \quad (3.162)$$

Nas Equações 3.161 e 3.162, percebe-se que os  $2n$  primeiros elementos do vetor de medições podem ser expressos como função do vetor de estados  $x_t$ , como desejado, mas também como função do ponto tridimensional equivalente a projeção do ponto na imagem. Isto é, o ponto  $(u_i, v_i)$  na imagem  $I_t$ , é uma função do ponto tridimensional  $\begin{pmatrix} X_{i,t} & Y_{i,t} & Z_{i,t} \end{pmatrix}^T$ . As coordenadas deste ponto não se encontram no vetor de estados, mas são fornecidas através de uma medição a parte do FKU projetado. Esta medição se dá através de uma triangulação de pontos, que serve justamente para recuperar as coordenadas 3D de pontos que estão presentes em duas imagens consecutivas, das quais se conhecem as coordenadas de pontos equivalentes em ambas

as imagens (por isso a necessidade de o ponto presente no vetor  $y_t$  estar presente também na imagem  $I_{t-1}$ ) [68, 17].

Para finalizar a apresentação dos modelos de medição dos elementos do vetor  $y_t$ , as coordenadas da epípola,  $e_{u,t}$  e  $e_{v,t}$ , podem ser modeladas pela ideia de formação da epípola a partir da movimentação da câmera, isto é, de que o ponto onde se encontra a epípola na imagem  $I_t$  é exatamente o ponto onde se encontraria a projeção do ponto de origem da câmera no instante em que a imagem  $I_{t-1}$  foi obtida. Em termos matemáticos:

$$\mathbf{e}_t = \mathbf{K} \mathbf{t}_{t-1}^t, \quad (3.163)$$

onde  $\mathbf{K}$  corresponde a matriz da câmera e o  $\mathbf{t}_{t-1}^t$  corresponde ao vetor de translação  $\begin{pmatrix} t_{1,t} & t_{2,t} & t_{3,t} \end{pmatrix}^T$  correspondente ao deslocamento da câmera entre os instantes de tempo  $t-1$  e  $t$ . Este vetor de deslocamento pode ser obtido a partir da posição estimada pela fusão sensorial do agente robótico no instante de tempo  $t-1$  e  $t$  em conjunto com o conhecimento da posição da câmera relativo ao corpo do robô.

Seguindo então na representação matemática da epípola, tem-se que:

$$\mathbf{e}_t = \begin{bmatrix} e_{1,t} \\ e_{2,t} \\ e_{3,t} \end{bmatrix} = \begin{bmatrix} \alpha & -\alpha \cot(\phi) & u_0 \\ 0 & \frac{\beta}{\sin(\phi)} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} t_{1,t} \\ t_{2,t} \\ t_{3,t} \end{bmatrix} \quad (3.164)$$

o que leva às expressões algébricas:

$$e_{1,t} = \alpha t_{1,t} - \alpha \cot(\phi) t_{2,t} + u_0 t_{3,t}, \quad (3.165)$$

$$e_{2,t} = \beta \frac{t_{2,t}}{\sin(\phi)} + v_0 t_{3,t}, \quad (3.166)$$

$$e_{3,t} = t_{3,t}. \quad (3.167)$$

Convertendo então o ponto  $\mathbf{e}_t$ , dado em coordenadas homogêneas, para coordenadas em pixel na imagem  $I_t$ , tem-se que  $e_{u,t}$  é dado por:

$$e_{u,t} = \frac{e_{1,t}}{e_{3,t}} \quad (3.168)$$

$$e_{u,t} = \frac{\alpha t_{1,t} - \alpha \cot(\phi) t_{2,t} + u_0 t_{3,t}}{t_{3,t}} \quad (3.169)$$

$$e_{u,t} = \frac{\alpha}{t_{3,t}} (t_{1,t} - \cot(\phi) t_{2,t}) + u_0 \quad (3.170)$$

e que  $e_{v,t}$  é dado por:

$$e_{v,t} = \frac{e_{2,t}}{e_{3,t}} \quad (3.171)$$

$$e_{v,t} = \frac{\beta \frac{t_{2,t}}{\sin(\phi)} + v_0 t_{3,t}}{t_{3,t}} \quad (3.172)$$

$$e_{v,t} = \frac{\beta t_{2,t}}{\sin(\phi) t_{3,t}} + v_0. \quad (3.173)$$

Assim, conclui-se a apresentação de todas as etapas do Filtro de Kalman Unscented projetado para servir de estimador dos parâmetros internos da câmera. Lembrando que o filtro foi projetado para ser executado em paralelo ao outro FKU apresentado nesta seção, responsável pela fusão sensorial da Odometria Visual com as leituras dos sensores inerciais da UMI.

### 3.3.2.5 Considerações Finais

Fica concluída então a apresentação do algoritmo de Localização Visual Inercial com Auto-Calibração da Câmera (LOVIACC) proposto neste trabalho. O algoritmo surgiu como uma espécie de aperfeiçoamento do método de auto-calibração proposto na seção 3.2 e é baseado no conjunto de dois FKUs que funcionam em paralelo.

Um dos filtros é responsável pela fusão sensorial entre as estimativas de posição e orientação obtidas por um sistema de Odometria Visual com as leituras dos acelerômetros e giroscópios da UMI carregada pelo agente robótico. Já o outro FKU é responsável pela estimação dos parâmetros internos da câmera.

A análise de desempenho deste algoritmo, bem como possíveis pontos a serem melhorados serão apresentados no próximo capítulo, que apresentará alguns experimentos realizados para fins de validação dos algoritmos propostos.

# 4

## ANÁLISE DE DESEMPENHO

---

### 4.1 INTRODUÇÃO

O capítulo 3 apresentou dois algoritmos desenvolvidos ao longo da realização da pesquisa deste trabalho. Inicialmente, foi apresentado um primeiro algoritmo desenvolvido para se implementar um sistema de auto-calibração de uma câmera monocular. Este algoritmo funciona a partir das imagens capturadas pela câmera e com o auxílio de uma Unidade de Medição Inercial (UMI), sem qualquer necessidade de conhecimento, ou preparo, prévio do ambiente em que a câmera captura suas imagens.

Em seguida foi apresentado um segundo algoritmo que veio para expandir o primeiro. Este segundo algoritmo tinha como objetivo não somente auto-calibrar a câmera, mas também estimar, de maneira confiável, a posição e orientação de um agente robótico que porta os dois sensores (câmera e UMI). Chamou-se este segundo algoritmo de Sistema LOVIACC (Localização Visual Inercial com Auto-Calibração de Câmera).

Chega-se então ao presente capítulo, que tem como objetivo reportar os experimentos que foram realizados para validar e testar os dois algoritmos apresentados. A seção 4.2 descreve o experimento realizado para a validação da Auto-Calibração de Câmera com Auxílio de uma UMI bem como faz uma análise dos resultados encontrados na estimação dos parâmetros internos da câmera com o FKE implementado. Já a seção 4.3 trata dos experimentos realizados para validar o sistema de LOVIACC projetado e também traz análises e comparações a cerca dos resultados encontrados tanto na estimação dos parâmetros internos da câmera quanto na estimativa da localização a partir das leituras dos sensores envolvidos.

### 4.2 VALIDAÇÃO DA AUTO-CALIBRAÇÃO DE CÂMERA COM AUXÍLIO DE UMA UMI

Para a validação do algoritmo de Auto-Calibração de Câmera com Auxílio de uma UMI, que tem como objetivo realizar a estimativa dos parâmetros internos da câmera, realizou-se um experimento no mundo real com uma câmera capturando imagens e uma UMI, acoplada à câmera, fazendo leituras com seus sensores inerciais.

A câmera utilizada no experimento foi uma Firefly FMVU-03MTM/C *global shutter* capturando imagens a uma taxa de cinco quadros por segundo. Já a Unidade de Medição Inercial utilizada foi uma Memsense Nano IMU que, como dito no capítulo 3, realizava suas medições em sincronia com a captura das imagens da câmera, ou seja, sempre que uma nova imagem era capturada, uma nova leitura da UMI também ficava disponível.

A Figura 4.1 apresenta a montagem dos sensores que foram utilizados para a realização do experimento. Como pode-se perceber, a UMI é fixamente acoplada ao corpo da câmera, e a transformação de corpo rígido entre as origens referenciais dos sensores é medida.



Figura 4.1: Montagem dos sensores utilizados na realização do experimento. Tem-se uma câmera Firefly FMVU-03MTM/C *global shutter* com a UMI Memsense Nano IMU fixamente acoplada ao corpo da câmera.

#### 4.2.1 Descrição do Experimento

Uma vez tendo sido apresentados os modelos dos sensores, suas taxas de captura e a configuração física na qual os mesmos foram dispostos, pode-se seguir com a descrição do experimento.

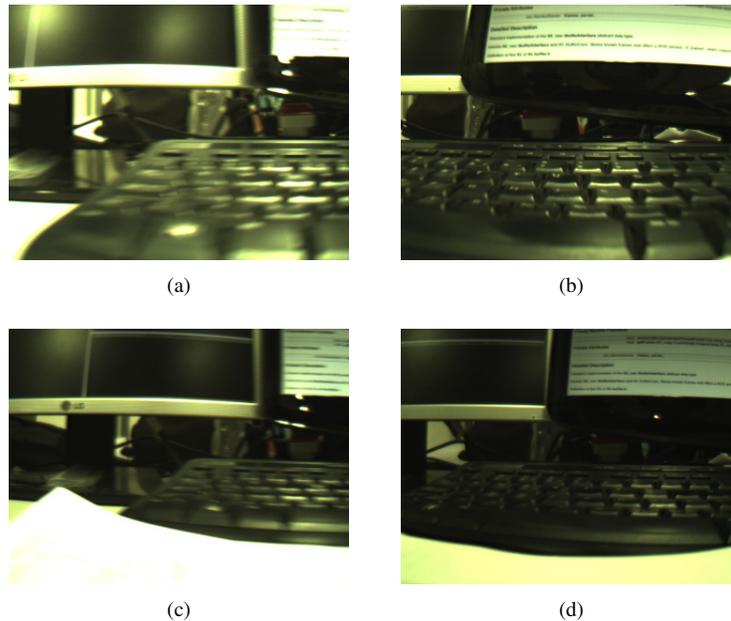


Figura 4.2: Imagens capturadas pela câmera enquanto a mesma era submetida a um movimento circular sobre uma mesa de laboratório.

O experimento consistiu basicamente em segurar o conjunto de sensores com as mãos e registrar, em um computador, suas leituras obtidas durante o movimento que os sensores foram submetidos. No caso, os sensores foram deslocados no ar realizando movimentos circulares (realizando uma trajetória semelhante a uma forma de pizza) com a câmera sempre apontando para a mesma direção e tentando manter a orientação global constante.

A Figura 4.2 apresenta quatro imagens capturadas enquanto os sensores eram deslocados sobre uma mesa de laboratório. Como pode-se observar, as imagens deixam claro que a câmera sempre esteve voltada para a mesma direção, maximizando assim o número de pixels que estavam presentes em todas as imagens capturadas ao longo da execução do movimento.

Como mencionado anteriormente, as imagens e leituras da UMI foram armazenadas em uma espécie de *dataset* para que fosse possível executar o algoritmo de forma independente, isto é, sem a necessidade de conectar os sensores ao computador e movimentá-los toda vez que se desejasse rodar o estimador. Sendo assim, tem-se que o experimento implementou o FKE estimador de parâmetros de maneira *offline*. Isto facilitou muito o desenvolvimento do código, já que permitiu a execução do mesmo sem os sensores.

No próximo tópico serão apresentadas as estimativas das variáveis de estado do FKE estimador.

## 4.2.2 Resultados das Estimações

Uma vez tendo sido obtidos os dados dos sensores, isto é, as imagens capturadas com a câmera e as leituras da UMI correspondentes aos instantes de captura das imagens, pode-se executar o Filtro de Kalman Estendido apresentado na seção 3.2 do capítulo anterior.

A Figura 4.3 apresenta a evolução das estimativas obtidas com a execução do FKE projetado para os parâmetros relacionados à distância focal da câmera,  $\alpha$  e  $\beta$ , para o parâmetro  $\phi$ , correspondente a angulação existente entre os eixos  $u$  e  $v$  do sensor da câmera, e para o parâmetro  $k_1$ , que corresponde ao primeiro fator de distorção radial, conforme descrito no capítulo 3.

Nos gráficos encontram-se representados duas medidas. Em azul está representada a estimação do parâmetro encontrado pelo FKE. Em vermelho, está o valor encontrado para o mesmo parâmetro interno da câmera só que obtido utilizando uma técnica de calibração convencional previamente executada (de maneira *offline* e com conhecimento das dimensões de um tabuleiro de xadrez apresentado para a câmera, como ilustrado na Figura 2.16).

Como é de se esperar, observa-se que as estimativas encontradas pela técnica convencional permanecem constantes, enquanto que as estimativas dos parâmetros obtidos pelas estimativas do FKE apresentam uma evolução ao longo das capturas das imagens. Os valores obtidos pela técnica convencional são utilizados como sendo os “valores verdadeiros” dos parâmetros. O que é uma assunção um tanto razoável, uma vez que a câmera foi calibrada (com a técnica convencional) instantes antes de ser utilizada para a captura do *dataset* e, durante as capturas e execução do movimento, não foi evidenciado nenhum choque mecânico da câmera ou qualquer outra avaria que justificasse a alteração dos seus parâmetros internos durante as capturas.

A evolução apresentada pelas estimativas do FKE se mostram, em um curto período de tempo, caminhar para o encontro dos valores calibrados pela técnica convencional. O que mostra que os resultados das estimativas do FKE são coerentes com o que se espera das calibrações.

Observa-se que todos os valores iniciais de todos os parâmetros apresentados foram escolhidos de modo que estivessem próximos ao valor “real”, como no caso do parâmetro  $\alpha$ , ou então relativamente distantes do

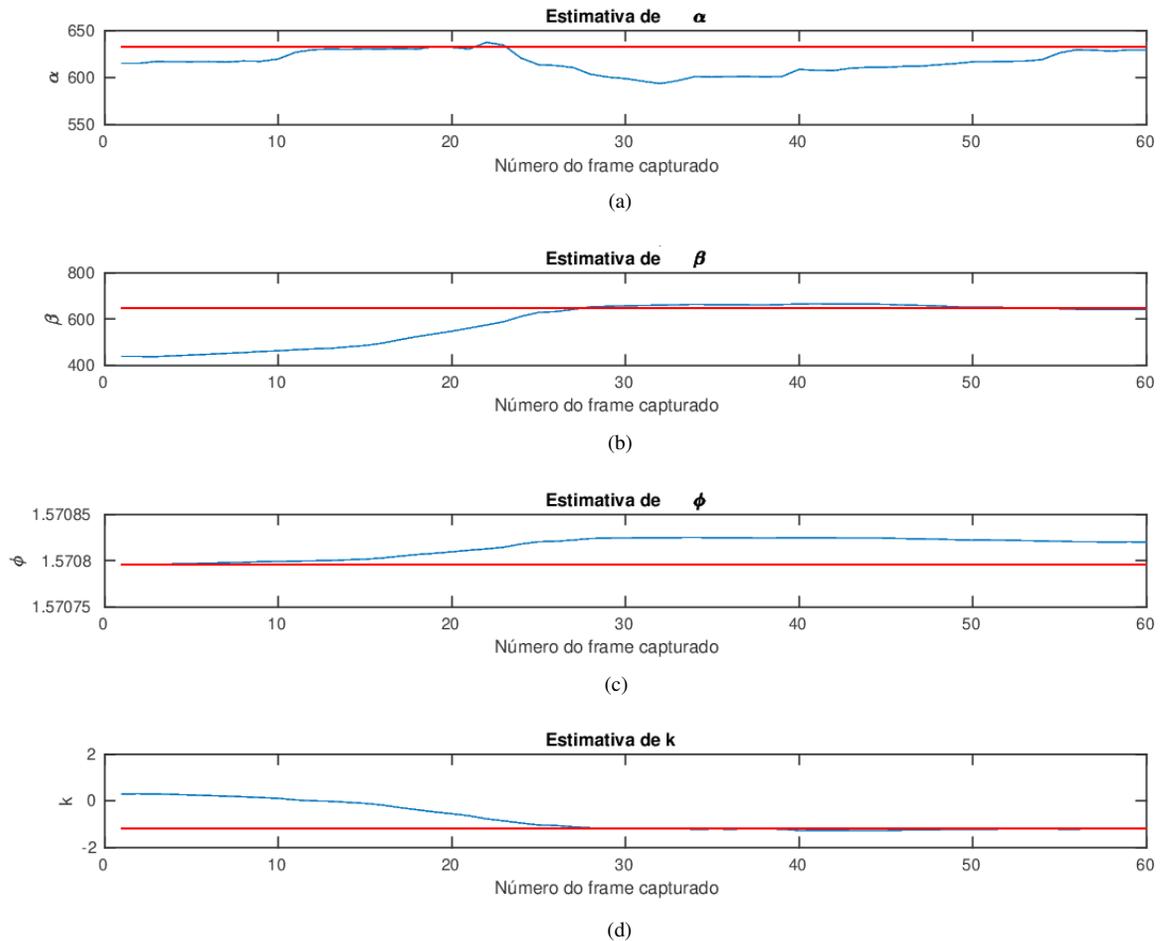


Figura 4.3: Resultados da aplicação do FKE para as estimativas dos parâmetros internos da câmera ao longo do número de *frame* capturado. (a) Estimação do parâmetro  $\alpha$ ; (b) Estimação do parâmetro  $\beta$ ; (c) Estimação do parâmetro  $\phi$ ; (d) Estimação do parâmetro  $k$ .

parâmetro “real”, como no caso dos parâmetros  $\beta$  e  $k_1$ , ou ainda com o mesmo valor do “real”, como no caso do parâmetro  $\phi$ . Importante notar que, mesmo os valores iniciais mais distantes dos “reais” não foram fisicamente incoerentes, como seria, por exemplo, um valor relativo a distância focal negativo.

Com a escolha dos parâmetros iniciais estando em diferentes posições relativas ao seu “valor real”, espera-se mostrar que essa escolha, ou “chute inicial”, pode ser feita de modo quase independente do valor real do parâmetro. Mas recomenda-se que este valor seja um valor próximo ao real e, claro, fisicamente compatível com o mundo real.

A Figura 4.3a apresenta a evolução da estimativa do parâmetro  $\alpha$  ao longo da sequência dos quadros capturados pela câmera. Como já mencionado, este foi um parâmetro que teve o seu valor inicial escolhido já bem próximo ao valor real e, talvez por isso, dentre todos os parâmetros apresentados, ele tenha sido o que mais rápido atingiu o valor “real” que deveria, o que aconteceu por volta da décima segunda imagem capturada. Esta estimativa se manteve constante pelo intervalo de tempo que correspondeu as dez capturas seguintes.

Porém, neste intervalo de tempo em que a estimativa de  $\alpha$  esteve constante, os demais parâmetros,  $\beta$ ,  $\phi$  e  $k_1$ , ainda estavam com suas estimativas relativamente distantes do valor real. Quando as estimativas destes

Tabela 4.1: Tabela resumo dos parâmetros.

Parâmetro	Valor Real	Valor Inicial	Valor Final	Erro médio
$\alpha$	633	615	629	$\sim 16$
$\beta$	649	440	640	$\sim 59$
$\phi$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\sim \frac{\pi}{2}$	$\sim 0$
$k_1$	$\sim -1,2$	0,3	$\sim -1,19$	$\sim 0,4$

últimos parâmetros atingiram valores bem próximos do que seria o seu valor “real”, o que aconteceu entre as capturas do *frame* número 20 e 30, a estimativa do parâmetro  $\alpha$  atingiu um pico de erro em sua estimativa. Mas, com a chegada de novos *frames*, a sua estimativa foi atingindo valores cada vez mais próximos do valor real. Terminando a estimativa com um erro de estimativa muito pequeno.

Analisando a Figura 4.3b, nota-se o padrão evolutivo da estimativa do parâmetro  $\beta$ , no qual percebe-se exatamente o comportamento descrito no parágrafo anterior. Até cerca do trigésimo quadro capturado pela câmera, a estimativa evolui gradativamente se aproximando do valor real do parâmetro. Uma vez este tendo sido praticamente atingido, a estimativa de  $\beta$  se mantém constante.

A Figura 4.3c apresenta as estimativas do parâmetro  $\phi$ . Observando a escala em que o gráfico está representado, nota-se que a estimativa foi virtualmente constante ao longo de todo o experimento. O que de certa forma faz sentido, uma vez que a angulação existente entre os eixos do sensor visual tende a ser de  $\frac{\pi}{2}$  radiano.

Já a Figura 4.3d mostra a evolução da estimativa do parâmetro  $k_1$  da câmera. Como se observa, o parâmetro teve um valor inicial distante do real, inclusive com sinal invertido em relação ao valor real, o que previa uma distorção radial com comportamento bem diferente do observado nas imagens. Porém, com o passar das capturas a estimativa foi se aproximando cada vez mais do valor real do parâmetro.

A Tabela 4.1 apresenta um resumo dos dados iniciais, valores reais, estimativas finais e erro médio ao longo do experimento para cada um dos parâmetros analisados. Esta tabela, bem como os gráficos de evolução das estimativas apresentados na Figura 4.3, mostram que o FKE projetado para realizar a auto-calibração dos parâmetros internos da câmera atingiu resultados que foram compatíveis com uma técnica convencional de calibração de câmera.

Claro não se pode assumir, com toda a certeza, que os valores encontrados na calibração tradicional sejam de fato os valores reais, porém, como estas técnicas são confiáveis a ponto de serem utilizadas na maioria das aplicações em que se necessita dos parâmetros internos da câmera, pode-se dizer que o algoritmo de auto-calibração da câmera proposto apresentou estimativas muito boas.

Apesar dos resultados das estimações dos parâmetros terem sido muito semelhantes aos encontrados por uma técnica de calibração convencional, os resultados da estimação da trajetória percorrida pela câmera ao realizar as capturas das imagens foram ainda mais impressionantes.

A Figura 4.4 apresenta, de maneira gráfica, a estimativa encontrada para a movimentação da câmera ao longo da realização do experimento. Com a ausência de um *ground truth* que possibilite uma análise quantitativa da estimativa, resta perceber, pela análise qualitativa, que o FKE foi capaz de estimar corretamente a trajetória descrita pela câmera. Esta análise qualitativa faz com que o FKE auto-calibrador de parâmetros também possa ser entendido como um algoritmo de estimação de trajetória.

Chega-se assim ao final da apresentação dos resultados do experimento realizado para validar o algoritmo

### Percurso Reconstruído

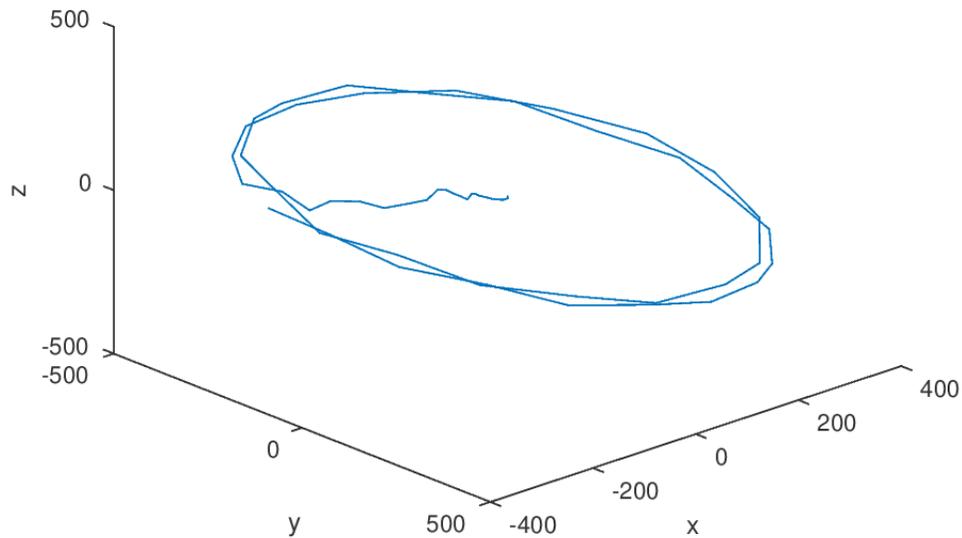


Figura 4.4: Reconstrução da trajetória percorrida pela câmera enquanto foram realizadas as capturas das imagens.

de auto-calibração de câmera proposto. Como visto, foi possível efetuar a calibração da câmera a partir de imagens capturadas por ela em conjunto com as leituras de uma UMI sem que fosse necessário qualquer conhecimento, ou preparo, prévio a respeito do ambiente. É válido ressaltar que as estimções obtidas foram muito semelhantes a uma calibração realizada com a mesma câmera através de uma técnica convencional.

Esta proximidade nos resultados das calibrações colocam o método desenvolvido como uma possível alternativa a métodos convencionais de calibração, só que com as vantagens de não precisar de nenhum preparo do ambiente e poder ser executado em paralelo a outras aplicações que usem a câmera. Além disso, caso ocorra alguma avaria à câmera, que por ventura altere seus parâmetros internos, o FKE será capaz de corrigir a estimativa e atualizar o valor estimado para o parâmetro.

Mas também é importante observar que o resultado aqui exposto serve, principalmente, para mostrar a viabilidade do algoritmo proposto. Isto é, ainda é possível perceber que algumas alterações no algoritmo são necessárias para que o mesmo fique ainda mais robusto às mais diversas situações. Por exemplo, um ponto fraco a ser melhorado do presente algoritmo é que ele ainda não conta com um dispositivo de troca do ponto tridimensional que está sendo rastreado. Isto faz com que exista a necessidade de o ponto tridimensional estar sempre presente nas capturas das imagens. Se por um acaso este ponto for perdido ou sair do enquadramento da imagem, o algoritmo, da forma que está, não será capaz de realizar estimções a respeito dos parâmetros internos da câmera.

Este problema tem uma solução relativamente simples, basta fazer com que, sempre que o ponto 3D sendo rastreado for perdido, se realize uma nova escolha de um ponto tridimensional cuja projeção esteja na imagem atual. Assim que for realizada a nova escolha para o ponto, deve-se reiniciar as variáveis de estado que tem relação com o ponto 3D, deixando apenas as estimativas dos parâmetros como estavam. Se isso for realizado,

perde-se a restrição de orientação da câmera.

Mas um outro ponto positivo que estes resultados mostram é que foi possível fazer a estimativa dos parâmetros internos da câmera e da trajetória por ela desempenhada, rastreando apenas um único ponto na imagem. Isto faz com que os custos computacionais sejam muito menores do que se rastreassem vários pontos entre as imagens. Isso porque com a existência de um único ponto a rastrear, o vetor de estados fica relativamente compacto e de tamanho fixo. Além disso, as inversões matriciais necessárias para implementar o FKE ficam mais leves para o computador quanto menor for o vetor de estados.

Com as análises aqui apresentadas e com os pontos fortes e fracos do algoritmo tendo sido devidamente ressaltados, fica o saldo positivo de que o algoritmo se mostrou como uma possível solução para o problema de auto-calibração de câmera.

### **4.3 VALIDAÇÃO DO SISTEMA LOVIACC**

Para a validação do algoritmo do sistema de Localização Visual Inercial com Auto-Calibração de Câmera (LOVIACC), desenvolvido ao longo do trabalho e apresentado na seção 3.3 do capítulo anterior, não se realizou um experimento em mundo real, mas sim uma série de experimentos simulados que visam a comprovação da eficiência do algoritmo.

Estes experimentos serão devidamente apresentados e explicados nesta seção. Da mesma forma, ainda serão mostrados os resultados obtidos para as estimções dos parâmetros internos da câmera, bem como das estimções de localização e orientação obtidas com a fusão sensorial entre as leituras dos sensores inerciais da UMI com as estimções de um sistema de Odometria Visual. Desta forma, pretende-se verificar que o sistema LOVIACC proposto representa uma boa alternativa de sensoriamento para navegação de robôs móveis.

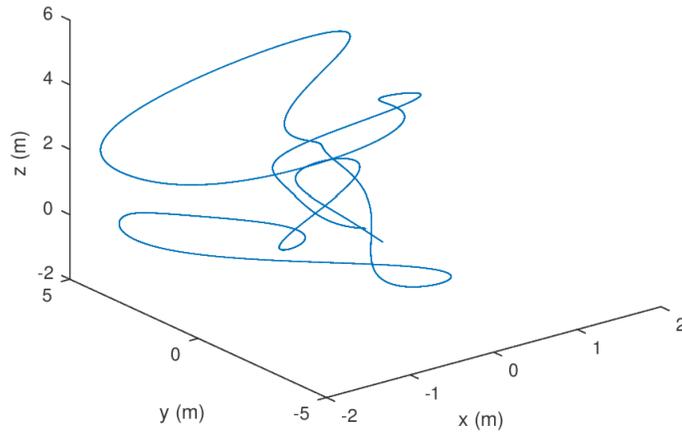
#### **4.3.1 Simulação**

Para realizar o experimento de validação do algoritmo LOVIACC desenvolvido, inicialmente fez-se uma simulação de um trajeto tridimensional no espaço percorrido por um agente robótico móvel. A partir deste trajeto, simulou-se medidas que uma UMI faria se estivesse sendo portada pelo robô em movimento no trajeto simulado. Além do trajeto, simulou-se também pontos 3D aleatoriamente posicionados no espaço. Estes pontos foram projetados com uma câmera emulada para gerar o uma imagem sintética. Essa câmera foi simulada como se estivesse sendo também carregada pelo robô. A transformação de corpo rígido entre a câmera emulada e o centro referencial da UMI simulada são conhecidos.

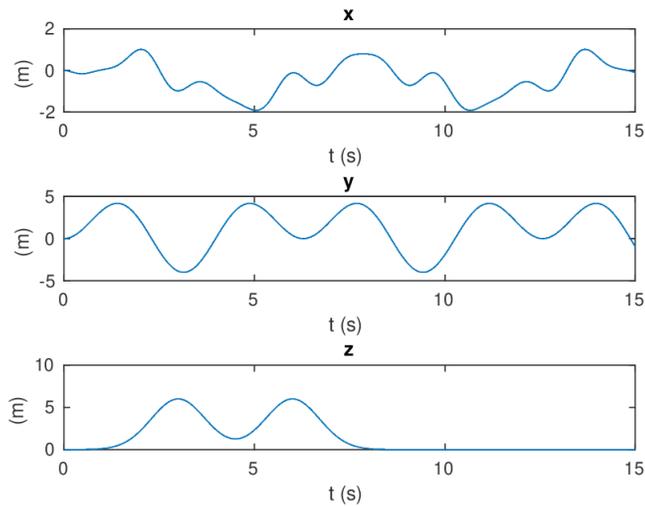
Sendo assim, todos os dados necessários para se realizar os experimentos com o algoritmo de LOVIACC foram gerados. A seguir serão apresentadas todas as simulações efetuadas que serviram de entrada para os experimentos realizados.

O trajeto simulado foi pensado como sendo uma movimentação livre no espaço 3D sem qualquer restrição de movimento ao robô, como por exemplo movimentá-lo apenas sobre o plano do chão. A ideia foi justamente testar se o algoritmo funciona para qualquer tipo de movimentação desempenhada pelo agente robótico.

A Figura 4.5 apresenta um conjunto de gráficos nos quais é possível ver o trajeto que foi simulado para a realização dos experimentos. A Figura 4.5a traz uma representação tridimensional da trajetória percorrida,



(a)



(b)

Figura 4.5: Simulação do percurso realizado pelo agente robótico. (a) Representação 3D do percurso; (b) Apresentação das três componentes ( $x$ ,  $y$  e  $z$ ) do percurso simulado.

onde é possível observar que o agente simulado percorre livremente pelo espaço. Já a Figura 4.5b mostra o mesmo percurso simulado, só que desta vez, apresentando a evolução temporal das três componentes,  $x$ ,  $y$  e  $z$ , do movimento.

A partir das imagens percebe-se que o trajeto foi inteiramente percorrido pelo robô em um intervalo de tempo de 15 segundos. Com a posição do agente robótico ao longo de todo o tempo de simulação, é possível também calcular a velocidade e a aceleração a que estava submetido o robô ao longo da realização do experimento.

A Figura 4.6 apresenta os gráficos gerados a partir da variação da velocidade e aceleração lineares ao longo do tempo da simulação.

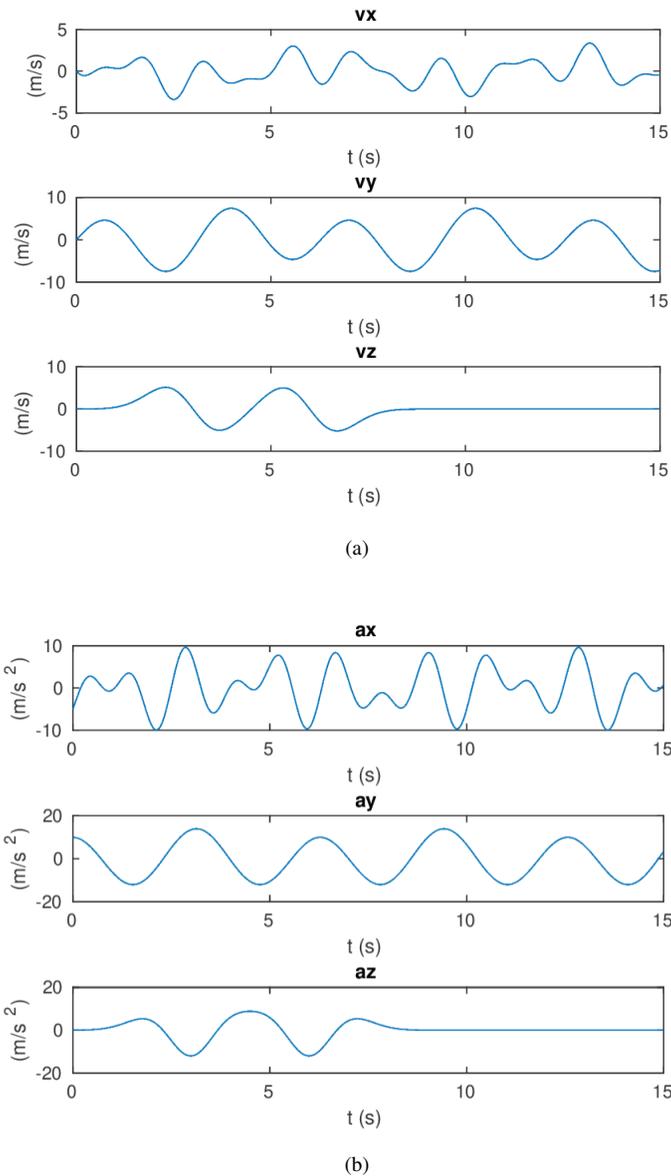


Figura 4.6: Simulação da velocidade e aceleração da movimentação realizada pelo robô durante o experimento. (a) Componentes da velocidade simulada ao longo do tempo; (b) Componentes da aceleração simulada ao longo do tempo.

Para finalizar a geração de dados relativos a movimentação do agente robótico do experimento, simulou-se também a orientação em que o centro de referência da UMI simulada estaria. Isto foi feito para que fosse comprovada a capacidade do algoritmo em lidar com movimentações que alterem a orientação do robô, dessa forma, será possível comprovar tanto que a fusão sensorial do algoritmo de LOVIACC consegue estimar a orientação quanto para mostrar que eventuais alterações na orientação do robô não afetam a estimativa dos parâmetros internos da câmera.

A Figura 4.7 apresenta os gráficos referentes a orientação dos eixos  $x$ ,  $y$  e  $z$ , das coordenadas da UMI em relação às coordenadas globais ao longo do tempo de simulação. As medidas nos gráficos estão apresen-

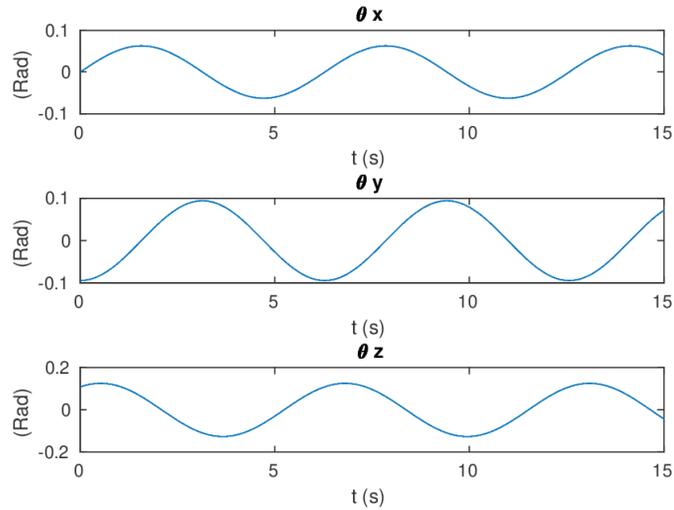


Figura 4.7: Ângulos de rotação ao longo dos três eixos principais ( $x$ ,  $y$  e  $z$ ) que correspondem a orientação do centro de referência da UMI simulada.

tadas em radianos. Esta foi a representação escolhida para geração dos gráficos pois a mesma é de mais fácil compreensão. Mas o algoritmo de LOVIACC, como mostrado na seção 3.3, trabalha com a representação da orientação por meio de um quatérnion unitário.

A partir de todas estas simulações a respeito da trajetória desempenhada pelo robô simulado, foi possível realizar as leituras simuladas da Unidade de Medição Inercial. Isto é, supondo que uma UMI acoplada a um robô móvel fosse submetida às movimentações descritas nas Figuras 4.5, 4.6 e 4.7, simulou-se quais seriam as leituras dos acelerômetros e giroscópios desta UMI.

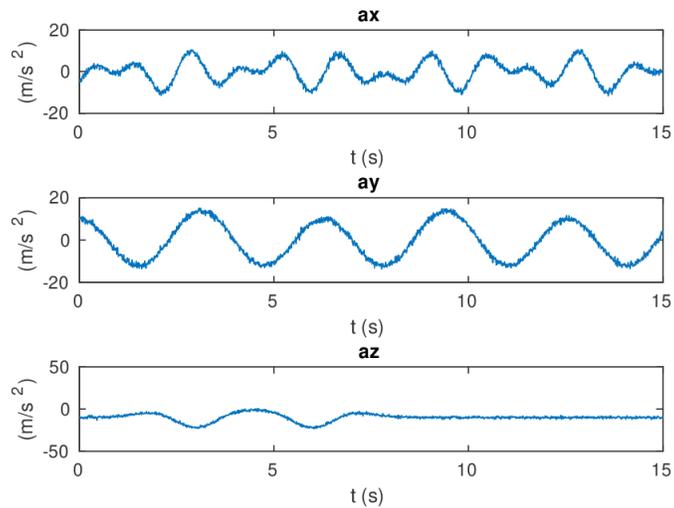


Figura 4.8: Simulação das leituras dos acelerômetros da UMI. Gráficos apresentam as componentes da aceleração percebidas pelos três acelerômetros.

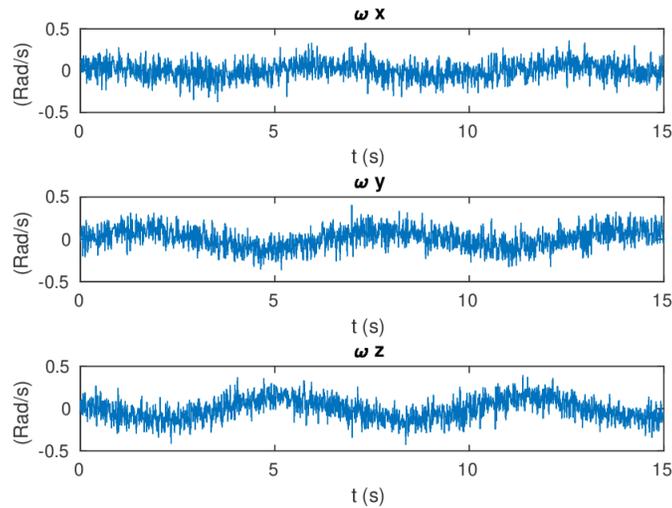


Figura 4.9: Simulação das leituras dos giroscópios da UMI. Gráficos apresentam as componentes da velocidade angular percebidas pelos três giroscópios.

Com as informações a cerca do funcionamento da UMI e de seus sensores apresentados na seção 2.2, tem-se que as simulações para as leituras dos sensores inerciais foram da velocidade angular, que seria lida pelos giroscópios, e da aceleração linear, lida pelos acelerômetros.

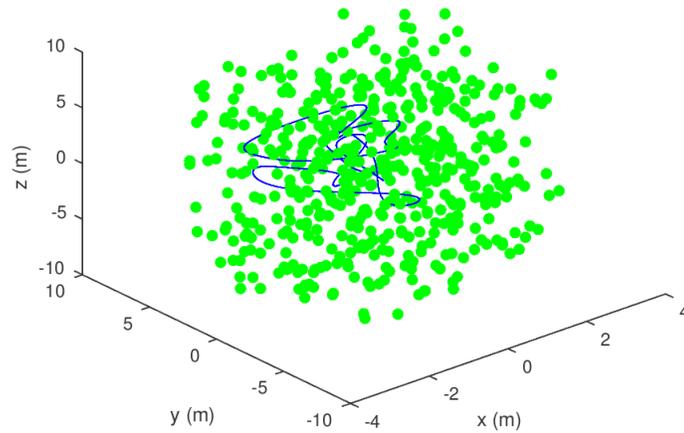
A Figura 4.8 apresenta as componentes da aceleração linear simuladas como leituras da UMI ao longo de todo o experimento. Já a Figura 4.9 mostra as componentes da velocidade angular, ao redor de cada um dos eixos de coordenadas, medidas pela simulação dos giroscópios ao longo, também, de todo o experimento.

É possível observar que tanto as leituras dos acelerômetros quanto dos giroscópios simulados não estão livres de ruídos. Estes ruídos possuem causas diversas e, mesmo não alterando consideravelmente as curvas de evolução da aceleração em relação a aceleração original (comparando as Figuras 4.6b e 4.8), já não permitem mais a reconstrução da trajetória sem que exista um erro de estimação proporcional ao quadrado do tempo de simulação. Isto será comprovado em alguns tópicos e contornado com a fusão sensorial que será apresentada, também, em tópico futuro.

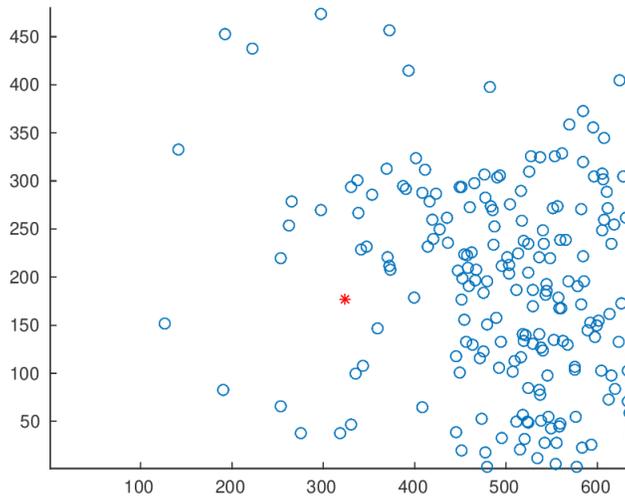
Para finalizar a simulação das leituras dos sensores utilizados no experimento, resta mostrar como foi feita a simulação das imagens capturadas pela câmera emulada, ou ainda, como foi o processo de sintetização de imagens para estes experimentos.

O processo de geração destas imagens sintéticas inicia-se com a criação de pontos tridimensionais espalhados, de maneira aleatória, ao longo do espaço de simulação. Esses pontos são representados na Figura 4.10a, que mostra aquele mesmo gráfico 3D com a representação da trajetória percorrida pelo robô ao longo do experimento só que envolvida por diversos pontos verdes espalhados pelo espaço. Estes pontos verdes são representações mínimas de pontos salientes que a câmera viria a capturar se estivesse observando uma cena real.

Partindo destes pontos espalhados no espaço, foi possível sintetizar uma imagem da cena vista pela câmera simulada com o auxílio da equação em coordenadas homogênea que modela a projeção dos pontos 3D já apresentada neste trabalho.



(a)



(b)

Figura 4.10: Geração de imagem sintética utilizada nos experimentos. (a) Nuvem de pontos 3D posicionados no espaço para simular pontos de interesse em uma imagem; (b) Imagem sintética gerada a partir da projeção dos pontos 3D do espaço no plano de imagem.

A geração destas imagens usou então o fato de se conhecer a posição tridimensional de cada ponto no espaço representado no que seria o {CAM} da câmera simulada em conjunto com a matriz da câmera,  $K$  (cujos parâmetros o sistema de LOVIACC ainda vai estimar).

Em termos matemáticos, a partir do conhecimento de:

$$K = \begin{bmatrix} \alpha & -\alpha \cot(\phi) & u_0 \\ 0 & \frac{\beta}{\sin(\phi)} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

e sabendo que as coordenadas  $u$  e  $v$  de cada pixel da imagem é a projeção de um ponto 3D cujas coordenadas no {CAM} são conhecidas e dadas por:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{Z_{cam}} \begin{bmatrix} \alpha & -\alpha \cot(\phi) & u_0 & 0 \\ 0 & \frac{\beta}{\sin(\phi)} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \\ 1 \end{bmatrix}, \quad (4.2)$$

foi possível sintetizar todos os *frames* que foram capturados pela câmera simulada ao longo de todo o experimento realizado.

A Figura 4.10b traz uma das imagens simuladas por este processo. Na imagem, cada círculo azul corresponde ao local de projeção de um ponto verde no espaço apresentado na Figura 4.10a, importante ressaltar que apenas são apresentados na imagem os pontos que estão no campo de visão da câmera simulada. Ainda na análise da imagem sintetizada, como todos os pontos projetados na imagem são conhecidos no mundo 3D, é possível realizar o rastreamento destes pontos entre quadros subsequentes sem qualquer problema. Em outras palavras, são conhecidas todas as coordenadas dos pontos que estão presentes em imagens consecutivas, o que é importante para o algoritmo LOVIACC, como mostrado na seção 3.3, seja na etapa de Odometria Visual ou na etapa de estimação de parâmetros da câmera.

Ainda na Figura 4.10b, é possível observar um asterisco vermelho em meio aos círculos azuis. Este asterisco corresponde a posição onde a epípola (entre as duas últimas visões da câmera simulada) se encontra. As coordenadas desta epípola também são importantes, e necessárias, para a execução do algoritmo LOVIACC.

Para finalizar a apresentação deste tópico, é válido dizer que o algoritmo de Odometria Visual implementado faz uso das imagens sintetizadas em conjunto com a última estimação dos parâmetros internos entregue pelo sistema LOVIACC (no caso da primeira iteração, utiliza-se os valores iniciais estimados). Desta forma, sempre que o LOVIACC atualizar a estimativa dos valores dos parâmetros, a estimação de movimento recuperada pela Odometria Visual tenderá a ser mais confiável, uma vez que, espera-se, os valores dos parâmetros internos se aproximem dos seus valores reais.

Os próximos tópicos apresentarão os resultados de alguns experimentos executados com o intuito de se verificar tanto a estimação dos parâmetros internos por parte do FKU estimador desenvolvido, quanto verificar as estimativas da localização desempenhadas pelo FKU da fusão sensorial, o que resulta no sistema LOVIACC.

### 4.3.2 Estimação dos Parâmetros

Para verificar se o FKU estimador de parâmetros internos do algoritmo LOVIACC realmente funcionam, formulou-se dois experimentos. Em ambos os experimentos foram utilizados como dados de entrada o conjunto de simulações a respeito da movimentação do robô (e de seus sensores) apresentada no tópico anterior.

O que se esperou destes experimentos foi verificar a capacidade do algoritmo realmente calibrar a câmera utilizada, isto é, partindo de um palpite inicial de quais seriam os valores dos parâmetros, conseguir encontrar os parâmetros que foram realmente utilizados para realizar as capturas das imagens sintéticas da simulação.

Aproveitando que os experimentos são realizados com dados simulados, foi proposto ainda que o algoritmo FKU não só fosse utilizado para calibrar a câmera para parâmetros estáticos, isto é, que apresentam valores fixos ao longo de todo o experimento, mas que o algoritmo fosse exposto a situações em que, além de estimar o valor correto dos parâmetros, que os “valores reais” destes parâmetros fossem alterados ao longo da coleta de dados

do experimento. Este tipo de experimento simula situações em que a câmera é submetida a alguma avaria que pode, de fato, alterar o valor de seus parâmetros internos. Em outras palavras, nos experimentos que serão apresentados, o algoritmo auto-calibrador deve ser capaz de estimar os valores corretos dos parâmetros

$$\begin{bmatrix} \alpha \\ \beta \\ \phi \\ u_0 \\ v_0 \end{bmatrix} \quad (4.3)$$

em tempo real, com eles sendo alterados ou se mantendo fixos ao longo da simulação.

O que difere os dois experimentos é justamente a maneira que altera os valores dos parâmetros internos utilizados para sintetizar as imagens e que deverão ser constantemente estimados. No primeiro experimento, utilizou-se uma função degrau para alterar os valores dos parâmetros e, no segundo experimento, foi proposto que os parâmetros da câmera seguissem uma função seno.

#### 4.3.2.1 Variação dos parâmetros seguindo um degrau

A primeira forma implementada de alteração dos valores dos parâmetros internos da câmera foi a utilização de uma função degrau. Isto é, os parâmetros iniciam a simulação com um valor e permanecem com este valor até o instante de tempo  $t = 9$  segundos. A partir deste instante de tempo, todos os valores de todos os parâmetros são alterados instantaneamente e permanecem com este novo valor até o final da simulação.

Este tipo de alteração dos parâmetros simula a situação em que a câmera sofre um choque mecânico que, no momento do impacto, altera os valores dos parâmetros.

Tabela 4.2: Valores dos parâmetros utilizados na sintetização de imagens ao longo da simulação do primeiro experimento.

Parâmetro	Valor antes do choque	Valor depois do choque
$\alpha$	360	378
$\beta$	340	323
$\phi$	$\frac{\pi}{2}$	$\frac{\pi}{2} - 0.03$
$u_0$	320	332
$v_0$	240	236

A Tabela 4.2 apresenta quais foram os valores de cada parâmetro utilizados ao longo do experimento antes e depois do choque simulado.

A Figura 4.11 apresenta um gráfico de evolução da estimação de cada um dos parâmetros internos da câmera. Nota-se que para alguns parâmetros, o choque mecânico alterou o valor para cima (como no caso dos parâmetros  $\alpha$  e  $u_0$ ) enquanto, para outros parâmetros, o choque alterou seu valor para baixo (parâmetros  $\beta$ ,  $\phi$  e  $v_0$ ).

Os resultados mostram também que todos os parâmetros partiram de um valor inicial diferente do valor real que desejava-se estimar, isto é, antes de qualquer coisa, o algoritmo deveria ser capaz de estimar o valor correto

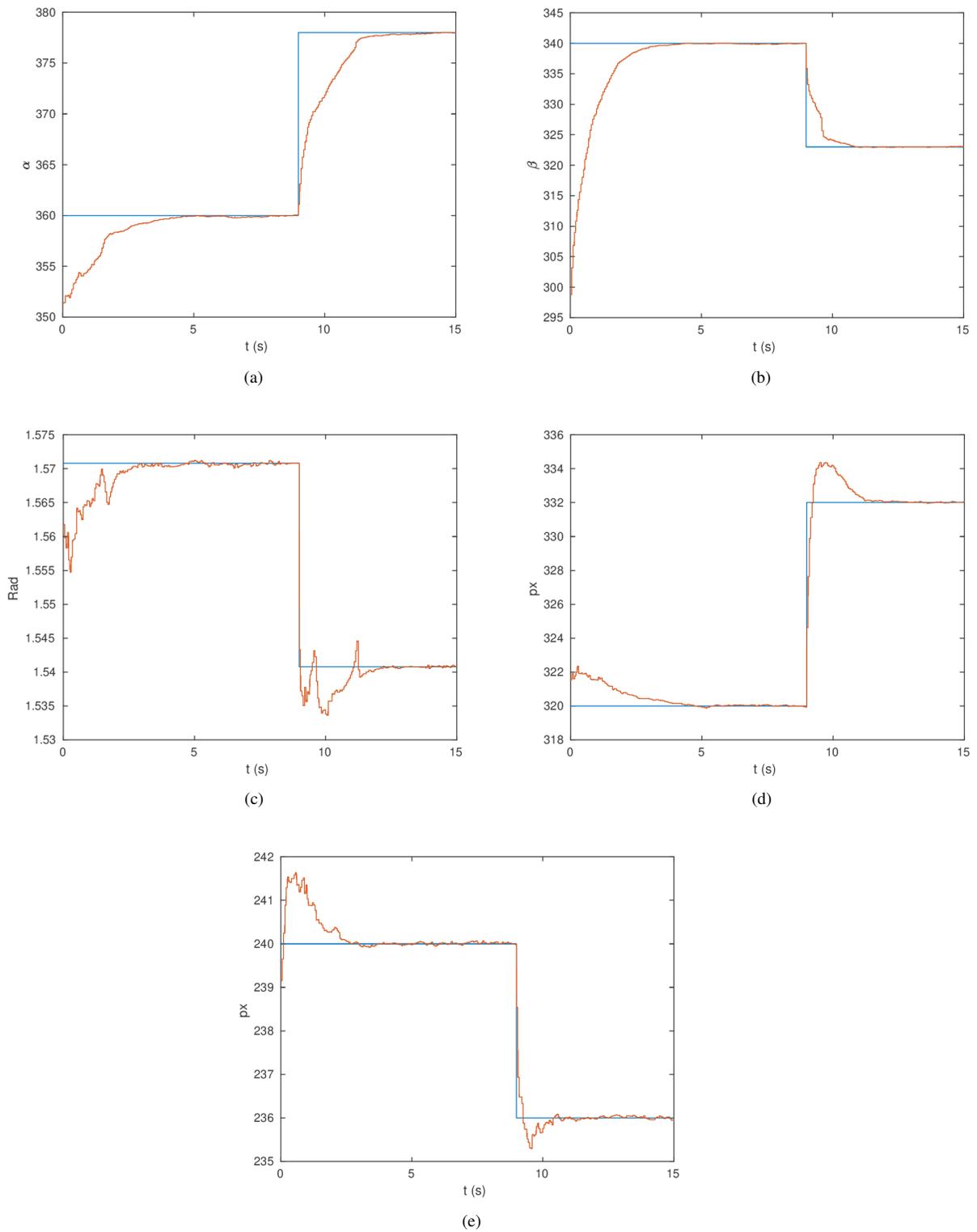


Figura 4.11: Resultados das estimações dos parâmetros com os valores reais seguindo uma função degrau. Em azul estão representados os valores reais dos parâmetros e, em vermelho, os valores estimados com o FKU. (a) Estimação do parâmetro  $\alpha$ ; (b) Estimação do parâmetro  $\beta$ ; (c) Estimação do parâmetro  $\phi$ ; (d) Estimação do parâmetro  $u_0$ ; (e) Estimação do parâmetro  $v_0$ .

dos parâmetros. E isso foi realizado. Todos os gráficos de evolução das estimações mostram que o parâmetro representado atingiu o valor esperado em um curto espaço de tempo (cerca de 5 segundos de simulação) independentemente se estavam com uma estimativa inicial próxima ou distante da real. Por exemplo, a Figura 4.11a, que mostra a evolução da estimação do parâmetro  $\alpha$ , demonstra que a estimativa para este parâmetro estava bem mais próxima do valor real do que a estimativa inicial do parâmetro  $\beta$ , cuja curva da evolução da estimação se encontra na Figura 4.11b. Para ambos os casos, de uma estimativa inicial próxima ou distante do valor real, o FKU se mostrou eficiente em recuperar o valor correto do parâmetro.

Sendo assim, o algoritmo se mostrou muito eficiente em realizar a calibração da câmera, uma vez que esta tarefa consiste em encontrar os parâmetros internos que compõem a matriz de projeção da câmera. Mas o experimento, como enunciado, vai além e explora ainda outra possibilidade: a de acontecer alguma coisa com a câmera e alterar os valores dos parâmetros no meio da execução do algoritmo de auto-calibração. Este efeito é simulado com a alteração dos valores dos parâmetros reais no meio da simulação, como mencionado anteriormente, por meio de uma função degrau. Os gráficos da Figura 4.11 mostram, mais uma vez, o sucesso do algoritmo em encontrar o valor real dos parâmetros.

O choque simulado que altera os valores reais dos parâmetros acontece no instante de tempo  $t = 9$  segundos. A partir deste instante de tempo, o algoritmo, mesmo já tendo encontrado o valor correto para todos os parâmetros, tem que encontrar o novo valor de cada parâmetro, ou seja, realizar a calibração da câmera novamente.

E os resultados das estimações comprovam que o algoritmo foi capaz de realizar a auto-calibração de maneira dinâmica ao longo da simulação, ou seja, a variação nos valores reais dos parâmetros não foi um grande problema para o algoritmo.

É importante lembrar que este procedimento de auto-calibração não é executado de maneira independente. Ele faz parte do algoritmo de LOVIACC, apresentado na seção 3.3, e é executado em paralelo com o FKU que implementa a fusão sensorial da Odometria Visual com as leituras da UMI que o agente robótico carrega. Para finalizar a apresentação dos resultados que validam o sistema LOVIACC como um todo, ainda faltam os resultados da estimativa de localização e orientação, mas estes serão apresentados após a exibição de um segundo experimento realizado para verificar a capacidade do sistema de auto-calibração encontrar os valores dos parâmetros.

#### 4.3.2.2 Variação dos parâmetros seguindo uma senoide

A fim de atestar que o algoritmo de auto-calibração consegue estimar os parâmetros internos da câmera nas mais diversas situações, foi proposto um segundo experimento mais “radical”. Se no primeiro experimento os valores dos parâmetros internos eram alterados uma única vez ao longo da simulação, representando um choque mecânico sofrido pela câmera, neste segundo experimento os valores serão alterados constantemente, cada parâmetro seguindo uma função senoidal.

É muito difícil de uma alteração como esta acontecer em uma situação realista, mas não deixa de ser um experimento válido para testar as capacidades de estimação do FKU auto-calibrador.

A Figura 4.12 apresenta um conjunto de gráficos que mostram a evolução temporal das estimações de todos os parâmetros estimados pelo FKU. Como pode-se perceber, mais uma vez o algoritmo se mostrou bem sucedido na tarefa de estimar os parâmetros corretamente. Claro que não foram observados os mesmos erros baixos do primeiro experimento, até porque os valores reais não se estabilizam, mas também não foram vistos erros crescentes ou muito grandes ao longo de toda a simulação.

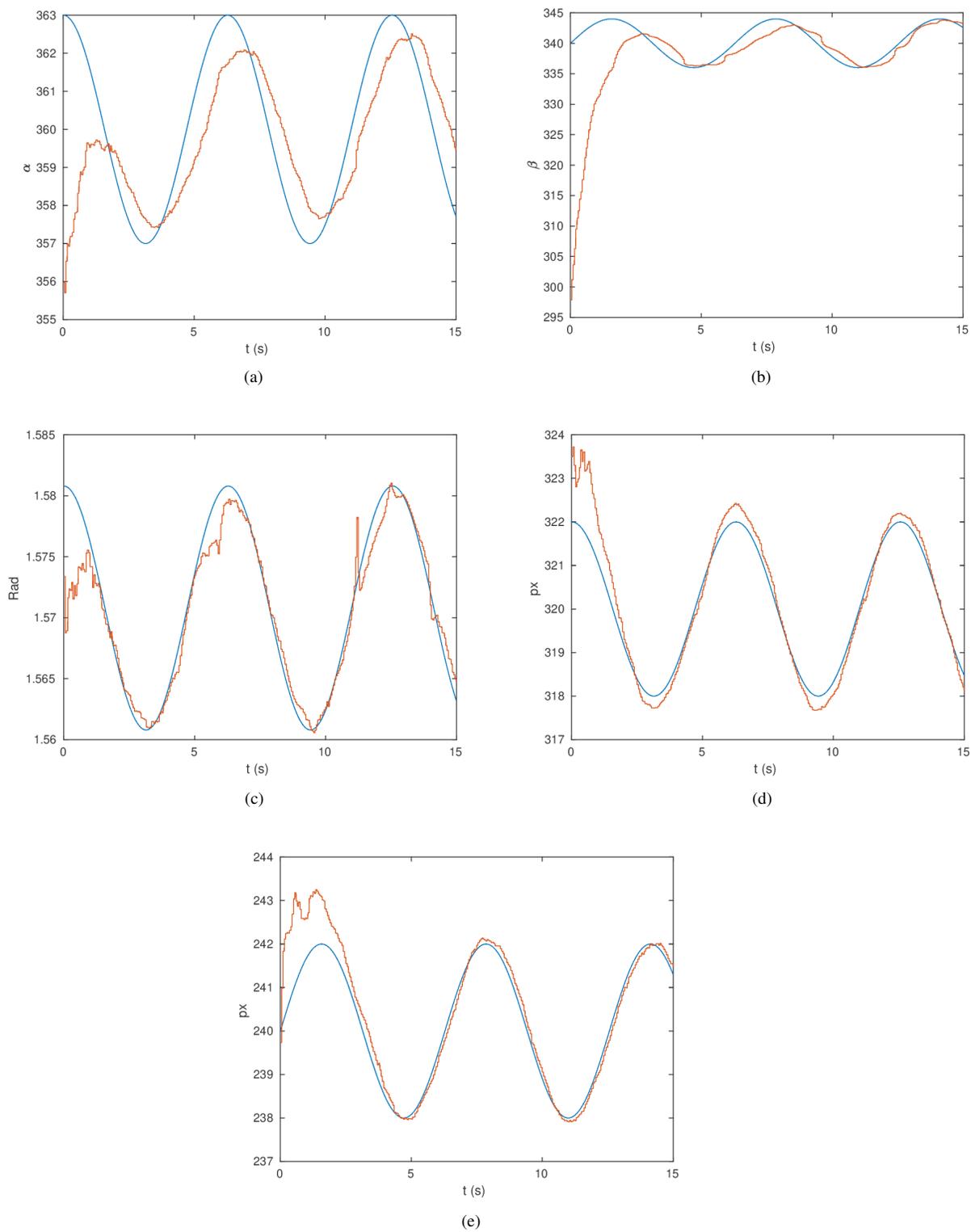


Figura 4.12: Resultados das estimações dos parâmetros com os valores reais seguindo uma função seno. Em azul estão representados os valores reais dos parâmetros e, em vermelho, os valores estimados com o FKU. (a) Estimação do parâmetro  $\alpha$ ; (b) Estimação do parâmetro  $\beta$ ; (c) Estimação do parâmetro  $\phi$ ; (d) Estimação do parâmetro  $u_0$ ; (e) Estimação do parâmetro  $v_0$ .

Ainda olhando para todos os gráficos, pode-se observar que mais uma vez foi adotada a estratégia de inicializar as estimativas dos parâmetros com valores ora próximos do valor real ora com valores mais discrepantes. Em ambas as situações, nota-se que o algoritmo foi capaz de realizar as estimativas de maneira que conduzisse a evolução da estimação de encontro ao valor real do momento.

Analisando agora apenas os gráficos referentes a evolução das estimativas dos parâmetros  $\alpha$ ,  $\beta$  e  $\phi$ , apresentados respectivamente nas Figuras 4.12a, 4.12b e 4.12c, percebe-se que estas foram as estimativas que apresentaram maiores “erros relativos” ao valor do parâmetro real. Nota-se até uma espécie de “atraso” da estimativa de  $\alpha$  em relação a seu valor real na Figura 4.12a, porém, estes erros ainda são muito pequenos e perfeitamente justificáveis quando se analisa a modelagem extremamente não linear dos parâmetros em conjunto com a variação dos valores reais.

Agora, as estimativas dos parâmetros  $u_0$  e  $v_0$ , apresentados nas Figuras 4.12d e 4.12e, respectivamente, conseguiram seguir muito bem os seus respectivos valores reais. Tal desempenho da estimativa pode ser justificado pela modelagem dos parâmetros ter como medição a epípola na imagem, que depende diretamente dos valores reais destes dois parâmetros.

Mesmo não tendo sido obtidos os mesmos resultados do caso da variação dos parâmetros seguindo uma função degrau, este segundo experimento, na qual a situação hipotética de haver parâmetros internos que variam no tempo seguindo uma função seno, demonstrou que o algoritmo de auto-calibração consegue estimar corretamente os parâmetros da câmera. O que corrobora mais uma vez para a comprovação de que o algoritmo foi bem sucedido em seu propósito.

O próximo tópico finaliza a apresentação dos resultados do sistema LOVIACC mostrando as estimativas encontradas com o FKU responsável pela fusão sensorial.

### 4.3.3 Estimação da Localização

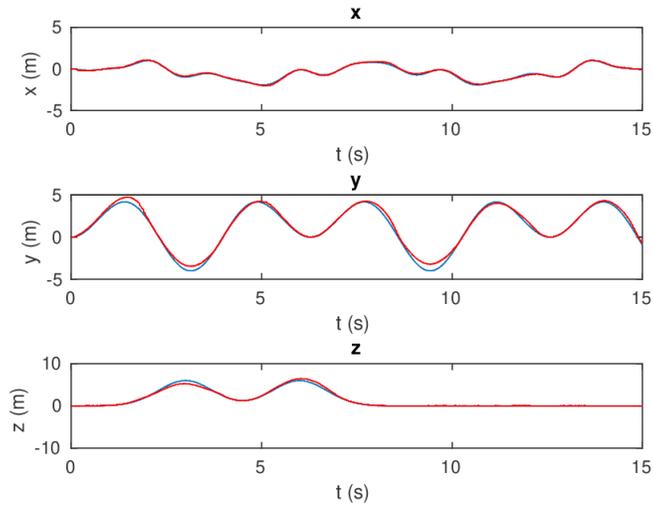
Até aqui foram apresentados os resultados para a estimação dos parâmetros internos da câmera, importante etapa do algoritmo de LOVIACC, porém ainda faltam os resultados em relação a estimação da Localização e Orientação para que, enfim, o sistema seja completamente validado.

Chega-se então ao tópico em que serão apresentados os resultados referentes as estimações realizadas pelo FKU responsável pela fusão sensorial da Odometria Visual com as leituras dos sensores inerciais. Como apresentado no tópico 4.3.1, tanto as imagens utilizadas pela Odometria Visual quanto as leituras da UMI são simuladas. A fusão sensorial foi implementada utilizando estes dados simulados já apresentados.

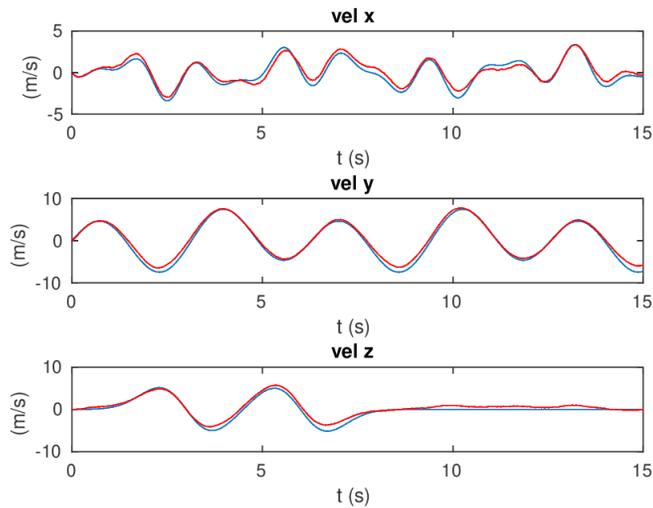
A fusão sensorial implementada é executada em paralelo com o FKU que realiza a estimação dos parâmetros internos da câmera e tem a função de, a cada nova leitura dos sensores, seja da UMI ou da câmera, atualizar a estimativa das variáveis de estado:

$$x_t = \begin{bmatrix} s_t \\ v_t \\ q_t \\ \lambda \end{bmatrix}. \quad (4.4)$$

A Figura 4.13 apresenta os gráficos da evolução das estimações das duas primeiras variáveis de estado,  $s_t = \begin{pmatrix} s_x & s_y & s_z \end{pmatrix}^T$  e  $v_t = \begin{pmatrix} v_x & v_y & v_z \end{pmatrix}^T$ , ao longo da execução da fusão sensorial. A Figura 4.13a



(a)



(b)

Figura 4.13: Resultado das estimações da fusão sensorial para a posição e velocidade. Em azul está representado o *ground truth* da movimentação e, em vermelho, as estimações. (a) Evolução das estimativas das componentes da posição do robô ao longo do tempo; (b) Evolução das estimativas das componentes da velocidade do robô ao longo do tempo.

apresenta a evolução das três componentes da posição estimada. Na imagem, a curva em azul corresponde ao percurso real, simulado e apresentado na imagem 4.5b, e a curva vermelha corresponde a posição estimada com a fusão sensorial implementada. Como pode ser observado, o FKU foi muito bem sucedido na estimação da posição do agente robótico, sendo muito difícil observar os erros existentes entre as curvas das estimações e dos trajetos reais nesta escala.

A Figura 4.13b apresenta resultados semelhantes para a estimação da velocidade do robô. Comparando as curvas da velocidade real (simulada) com as curvas das velocidades estimadas com a fusão, observa-se,

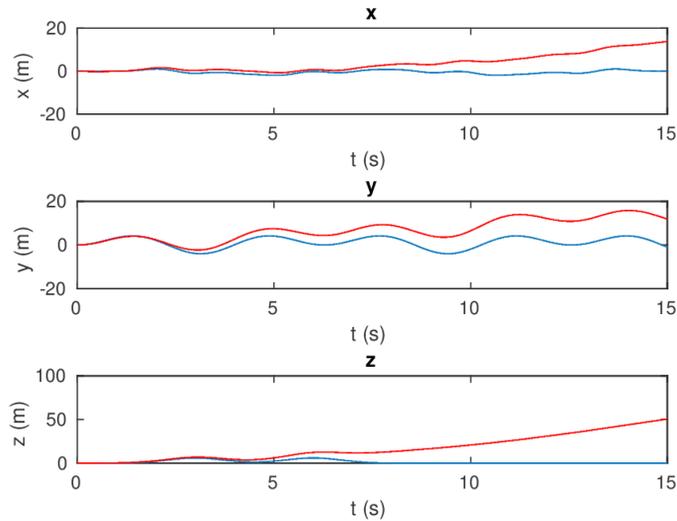


Figura 4.14: Estimativas das componentes da posição do agente robótico sendo rastreado apenas com o uso de um Sistema de Navegação Inercial. Foram utilizadas as mesmas leituras da UMI simuladas para o experimento.

mais uma vez, um erro muito pequeno na estimativa. Porém, já é possível observar erros maiores do que os obtidos nas estimativas da posição. Isso pode ser creditado ao fato de a velocidade do robô não ser medido diretamente por nenhum dos sensores envolvidos na fusão, uma vez que os acelerômetros medem a aceleração e a Odometria Visual estima a posição. Mas mesmo com estes erros um pouco maiores, as estimativas da velocidade ainda foram muito boas.

Para se ter uma boa noção da qualidade dos resultados obtidos com a fusão sensorial, a Figura 4.14 apresenta a mesma estimativa de rastreamento de posição do agente robótico percorrendo o mesmo trajeto, porém, fazendo uso apenas das leituras da UMI. Como previsto e descrito na seção 2.2, fazendo uso apenas dos sensores inerciais de uma UMI é muito fácil perder a confiança na posição estimada. Isso acontece por conta dos ruídos nas leituras dos sensores, que acabam se propagando quadraticamente para a estimativa de posição.

Mas de qualquer forma, a comparação entre os resultados da estimativa de posição fazendo uso apenas da UMI e da estimativa realizada pela fusão sensorial, mostradas respectivamente pelas figuras 4.14 e 4.13a, pode-se dizer que o FKU da fusão sensorial foi muito eficiente, fazendo uso dos dois sensores disponíveis para realizar estimativas muito mais precisas.

Claro que é importante observar que ambos os sensores realizam medidas não absolutas. Como já mostrado no capítulo 2, a UMI realiza medições inerciais e a Odometria Visual faz estimativas incrementais, isso faz com que, a longo prazo, a estimativa de posição se distancie da posição real do robô. Mas, como mostrado nos resultados apresentados, ao longo de toda a simulação realizada, os erros de estimação se mantiveram bastante contidos.

Para finalizar esta etapa de apresentação das estimativas da localização, a Figura 4.15 apresenta uma visualização gráfica da comparação entre o trajeto percorrido pelo robô na simulação e o trajeto estimado. Na imagem, a curva em azul corresponde ao trajeto simulado e a curva vermelha representa a trajetória estimada. Como é possível perceber, o resultado da estimação foi bastante próximo ao percurso verdadeiro.

O FKU de fusão sensorial ainda realiza estimações a respeito da orientação do agente robótico. A variável de estado  $q_t$  corresponde ao quatérnio unitário que representa a orientação do referencial do robô em relação

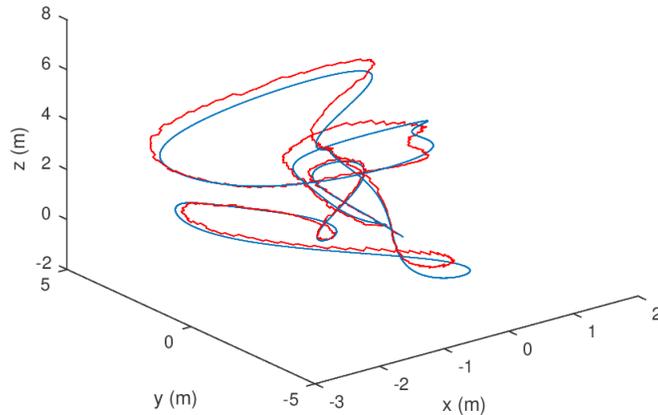


Figura 4.15: Representação 3D da comparação entre o percurso estimado e o simulado. Em azul se encontra a trajetória real do robô e, em vermelho, a trajetória estimada.

ao  $\{G\}$ . A Figura 4.16 apresenta a evolução das estimativas a respeito da orientação do robô ao longo do tempo de simulação. Na imagem, a curva azul corresponde a posição angular de cada um dos eixos do referencial do robô em relação ao sistema de coordenadas globais, da mesma forma como mostrado na Figura 4.7, e a curva vermelha corresponde a estimativa da posição angular de cada um dos eixos da referência do robô. É válido ressaltar, mais uma vez, que a representação da orientação neste gráfico não está na forma de quatérnio para que seja de mais fácil compreensão, porém, o FKU da fusão sensorial trata a orientação como um quatérnio.

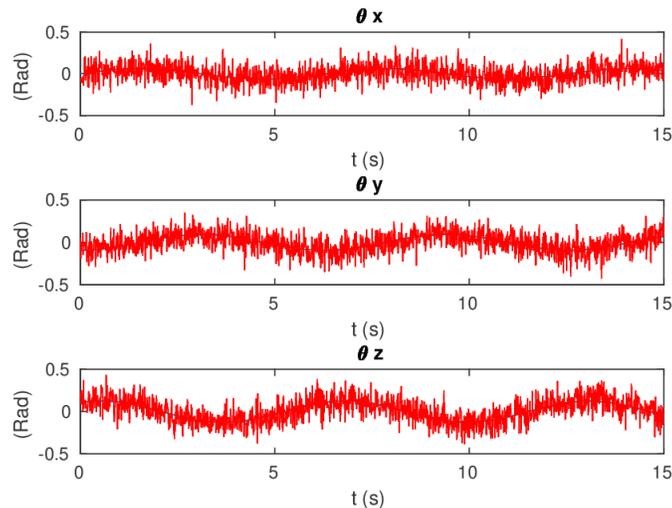


Figura 4.16: Evolução das estimativas das componentes da orientação do agente robótico. Em azul está a verdadeira angulação simulada e, em vermelho, os valores estimados para cada uma das rotações ao redor dos três eixos.

Apesar de a estimativa da orientação ser muito mais ruidosa, percebe-se que o comportamento da curva da

estimação é o mesmo da orientação real. Dessa forma, pode-se dizer que, mais uma vez, a fusão sensorial foi capaz de atingir seu objetivo.

Resta ainda uma última estimativa a ser apresentada. Como apresentado na seção 2.3, um sistema de Odometria Visual monocular não é capaz de descobrir, sozinho, a escala  $\lambda$  em que se dão suas estimações de translação. Fazer uso do sistema de Navegação Inercial composto pelos sensores da UMI é uma solução viável para recuperar esta escala e, como apresentado na seção 3.3, a fusão sensorial realizada pelo FKU é a técnica que conseguirá utilizar as informações dos dois sensores para descobrir esta escala  $\lambda$ .

É importante lembrar que para validar o algoritmo de LOVIACC foram realizados dois experimentos. Um com os parâmetros internos da câmera variando seguindo uma função degrau e outro com estes mesmos parâmetros seguindo uma senoide. Em ambos os experimentos foram realizadas estimativas dos parâmetros da câmera e da localização do robô. Este tópico que está apresentando os resultados das estimativas a cerca da localização mostra os resultados obtidos no experimento em que os parâmetros da câmera seguem a função degrau.

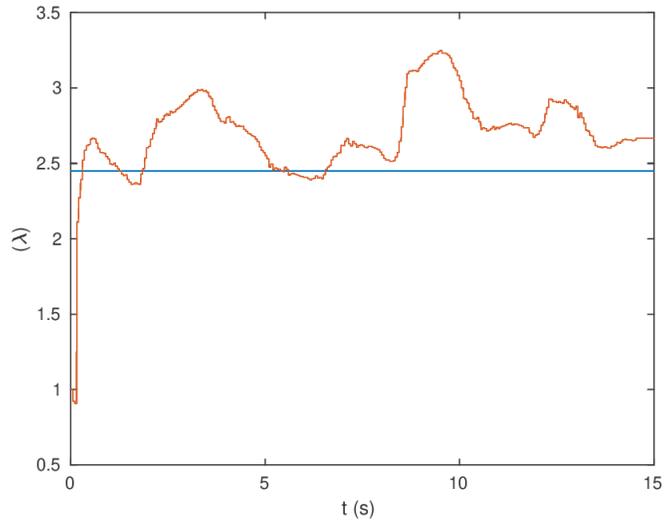
Em todas as simulações, as estimativas da Odometria Visual foram forçadas a apresentar uma escala  $\lambda$  conhecida. Dessa forma, pôde-se realizar as comparações com as estimativas da escala com um maior controle. Se esta ação não tivesse sido tomada, a escala em cada uma das execuções do algoritmo de Odometria Visual, mesmo com os mesmos dados simulados, poderia ser diferente.

A Figura 4.17a apresenta uma representação gráfica da evolução da estimação da escala  $\lambda$  ao longo do tempo. Na imagem, a curva vermelha corresponde a estimação da escala e a linha azul corresponde ao valor real da escala da OV. Como pode-se observar, a estimação apresenta  $\lambda = 1$  como valor inicial, ou seja, para o momento em que não se sabia absolutamente nada em relação à  $\lambda$ , o algoritmo optou por assumir que a Odometria Visual fornecia estimações na mesma ordem de grandeza que a UMI. Porém, em menos de 1 segundo de simulação, o FKU já foi capaz de estimar que este palpite inicial não condizia com os dados obtidos e fez com que a estimativa passasse a oscilar entre 2, 4 e 3. Esta oscilação foi ficando cada vez mais suave e com amplitude reduzida, indicando uma aproximação gradual para que a estimação estabilizasse no seu valor real.

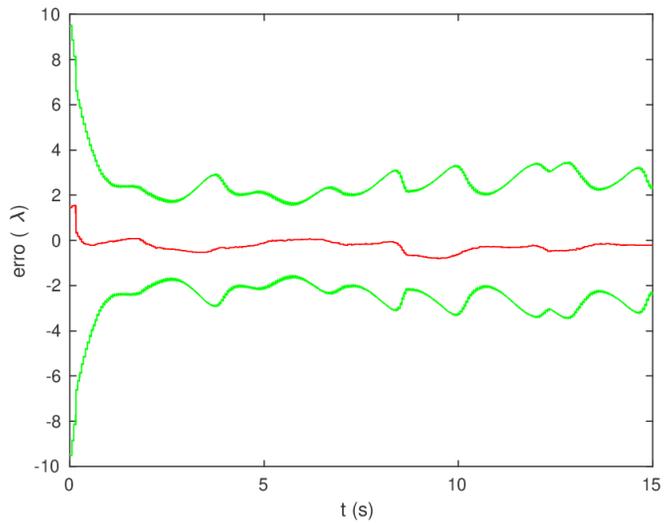
Mas no instante de tempo  $t = 9$  segundos da simulação, a função degrau alterou os parâmetros internos da câmera. Isto fez com que as estimações da Odometria Visual não mais contassem com os valores corretos dos parâmetros (ao menos naquele instante até o FKU estimador conseguir recuperar os novos valores dos parâmetros) e, conseqüentemente, observa-se que o valor estimado para  $\lambda$  apresentou seu maior erro desde o início da simulação exatamente neste instante. A medida que o tempo de simulação foi seguindo, as estimações voltaram a oscilar, atenuando a amplitude de oscilação, caminhando para uma estabilização bem próxima ao verdadeiro valor da escala.

Apesar de não haver a possibilidade de se medir diretamente o valor de  $\lambda$ , o FKU da fusão sensorial foi muito bem na tarefa de estimar esta escala, mesmo que indiretamente. A partir de uma suposição inicialmente genérica e errônea, o algoritmo conseguiu realizar estimações que foram coerentes com o que estava sendo observado. A utilização desta escala, com valor sempre muito próximo do valor real, possibilitou a Odometria Visual entregar uma boa contribuição em relação a suas estimações, o que possibilitou a fusão sensorial ser capaz de realizar boas estimações a respeito da localização e orientação do agente robótico rastreado.

A Figura 4.17b ainda apresenta uma outra análise a respeito da estimação da escala  $\lambda$ . Na imagem, a curva em vermelho apresenta o erro entre a estimativa de  $\lambda$  e o seu valor real ao longo do tempo. E em verde está representado o intervalo de confiança de  $\pm 3\sigma_\lambda$ , onde  $\sigma_\lambda$  foi extraído da diagonal da matriz de covariância



(a)



(b)

Figura 4.17: Resultado da estimação da escala  $\lambda$  da Odometria Visual obtida com a fusão sensorial. (a) Evolução da estimação de  $\lambda$  ao longo do tempo. Em azul está o valor real da escala e, em vermelho, o valor estimado; (b) Comparativo do erro da estimação, representado em vermelho, com  $\pm 3\sigma$ , apresentado em verde, referente a incerteza da estimação.

calculada com o FKU da fusão sensorial. É importante perceber que o erro da estimativa esteve sempre dentro deste intervalo de confiança. Outra observação possível é verificar que a amplitude do intervalo foi diminuindo com o tempo. Isto demonstra que, apesar de ter se começado com uma incerteza muito grande a respeito da estimação de  $\lambda$ , com o passar do tempo da simulação, esta incerteza foi diminuindo.

#### 4.3.4 Considerações finais

Com isto chega-se ao final da apresentação dos resultados dos experimentos realizados a fim de se validar o sistema de LOVIACC desenvolvido. Como pode-se perceber ao longo deste tópico, o sistema desenvolvido foi capaz de realizar corretamente as estimativas dos parâmetros internos da câmera e também da localização e orientação do agente robótico que carrega os sensores.

É crucial perceber que uma estimativa depende da outra, isto é, a fusão sensorial depende de uma boa estimativa dos parâmetros internos, pois assim a Odometria Visual consegue melhores estimativas, e o FKU estimador dos parâmetros também depende de uma boa estimativa a cerca da localização da câmera no espaço, que é obtida com a fusão sensorial.

Os bons resultados encontrados em todas as etapas dos experimentos permitem colocar o sistema LOVIACC desenvolvido como uma boa alternativa de sensoriamento para um robô móvel. Claro que, como observado ao longo da apresentação dos resultados, a localização do robô é feita com uso de sensores inerciais e incrementais. Isso faz com que, ao longo do tempo a estimativa vá aumentando sua incerteza. Isso pode ser corrigido caso haja a possibilidade de se fundir os resultados deste sistema LOVIACC com um sensor de GPS, por exemplo, que fornece, a uma taxa de captura bem mais baixa, a posição absoluta do robô em relação à Terra.

## 5.1 CONSIDERAÇÕES FINAIS

Assim, chega-se ao final da apresentação deste trabalho. O objetivo desta dissertação foi muito maior do que simplesmente apresentar as metodologias e algoritmos, implementados ou projetados, empregadas por traz das soluções propostas para a estimação dos parâmetros internos e da localização de um agente robótico a partir da leitura de sensores inerciais combinada com um sistema de Odometria Visual. Além de realizar a apresentação dos resultados da pesquisa desenvolvida ao longo do período de trabalho, teve-se o cuidado e a intenção de imergir o leitor na problemática que cerca a pesquisa.

Essa imersão do leitor foi elaborada de maneira que ele conseguisse perceber a real necessidade, utilidade e importância dos métodos apresentados. Para isso, teve-se a atenção de embasar muito bem o leitor a respeito da Localização Visual Inercial. Isso foi feito a partir de uma longa e detalhada explicação a respeito do assunto no capítulo 2, onde foi feita uma revisão bibliográfica em conjunto com embasamento teórico do tema.

A estratégia utilizada consistiu em, inicialmente, quebrar as técnicas de Localização nas categorias de “Navegação Inercial” e “Localização Visual”. Foi então apresentado todo o funcionamento dos sensores inerciais presentes em uma UMI, ou seja, de acelerômetros e giroscópios. A apresentação do funcionamento dos sensores foi relativamente detalhada, passando pelos princípios físicos que estavam envolvidos nas leituras destes sensores, mostrando possíveis configurações e formas de manufatura dos mesmos, exibindo estruturas biológicas presentes nos seres vivos que executam funções semelhantes e concluindo fazendo uma análise de como, matematicamente, pode-se fazer uso das leituras dos sensores para medir as grandezas físicas da aceleração e orientação de um agente robótico que porte os sensores.

Essa explicação teve esse nível de aprofundamento para deixar o leitor a par das principais características dos sensores inerciais da UMI. Foi possível se ter uma boa noção de quais são as vantagens e desvantagens no uso deste tipo de tecnologia. E principalmente, foi possível entender onde estão as limitações de um Sistema de Navegação Inercial, ou seja, onde estão as dificuldades, ou atuais impossibilidades, de se reconstruir a trajetória percorrida por um agente portando uma UMI a partir, unicamente, das leituras dos acelerômetros e giroscópios: nos erros apresentados pelas leituras dos sensores.

Depois de aprofundar o leitor no conhecimento da Navegação Inercial, teve-se o objetivo de apresentar o funcionamento de um tipo de sistema muito útil na robótica móvel: a Odometria Visual, que é o sistema de localização que usa somente imagens sequenciais capturadas por uma câmera para realizar uma estimativa a cerca da posição e orientação do agente robótico que carrega a câmera.

Foram apresentadas definições, classificações e principalmente, o algoritmo de Odometria Visual Monocular implementado neste trabalho. Foi transmitido para o leitor o algoritmo que consegue extrair, de maneira incremental, a estimação de movimentação da câmera a partir do deslocamento de pixels entre duas imagens. Com o aprofundamento apresentado, foi possível entender os pontos fortes e fracos da Odometria Visual. Talvez o maior problema de se utilizar apenas a OV para fazer a estimação da posição é que é impossível recuperar, apenas pelas imagens, a escala em que se deram as translações. Assim, o leitor foi familiarizado com a necessidade de informações complementares para o sistema conseguir recuperar esta escala das estimações.

Além disso, ficou claro, ao se apresentar o algoritmo de extração da movimentação, que o sistema de

Odometria Visual depende diretamente da Calibração da Câmera, isto é, de se conhecer, com a maior exatidão possível, os parâmetros internos que modelam matematicamente a câmera.

Uma vez tendo inteirado o leitor a respeito do funcionamento, vantagens e desvantagens, das duas técnicas de navegação (Inercial e Visual) em separado, iniciou-se uma discussão a respeito da possibilidade de unir as duas técnicas com o objetivo de minimizar o efeito das desvantagens de cada sistema de navegação. Em outras palavras, foi exposto que é possível utilizar as estimativas da Odometria Visual para corrigir os efeitos prejudiciais dos erros nas leituras dos acelerômetros da UMI, especialmente quando o robô está desempenhando um movimento mais lento, em contrapartida, as estimações obtidas pelo sistema de Navegação Inercial são capazes de auxiliar na determinação da escala da Odometria Visual, bem como atuar em situações de movimentação desempenhadas pelo robô de alta aceleração, isso porque as imagens capturadas nesse cenário podem ser borradas, dificultando a execução da OV, porém, neste cenário os erros lidos pela UMI são menos significantes, o que faz com que as estimações da Navegação Inercial sejam mais confiáveis. Ou seja, ficou bastante clara a complementariedade dos dois sistemas e a ideia de utilizá-los em conjunto se mostra como uma opção muito realista para se obter estimações mais precisas.

A partir da ideia de utilizar informações vindas de dois sensores diferentes em conjunto, foi apresentado como é possível interpretar o Filtro de Kalman como uma técnica de fusão sensorial. Foi deixada clara a ideia de o próprio algoritmo do Filtro já realiza uma “fusão” das estimativas a priori e a posteriori, e a partir dessa ideia, foi mostrado que se pode fazer a estimativa a priori ser realizada por um dos sensores e a posteriori por outro sensor. Se por um acaso for realizada a fusão entre a Odometria Visual e a Navegação Inercial, o resultado será a obtenção de um sistema de Localização Visual Inercial. O nome já diz por si só, é um sistema que será capaz de estimar a Localização de um agente robótico a partir das leituras dos sensores inerciais de uma UMI em conjunto com as estimações de um sistema de Odometria Visual.

Até então, foi apresentado para o leitor todo o embasamento para ele realmente acredite que a fusão das leituras destes dois sistemas é uma boa ideia. Em seguida, mudou-se um pouco de foco, para que fosse também apresentado um outro detalhe muito importante que é trabalhado, em igual proporção, nas propostas desta dissertação: a calibração de câmeras.

O leitor foi introduzido à modelagem matemática de uma câmera, foi dada atenção especial para que ficasse claro o papel físico de cada um dos elementos que compõem o conjunto dos parâmetros internos da câmera. Após uma rápida recapitulação a respeito da importância de se conseguir modelar a câmera para a Odometria Visual (pois é a partir desse modelo matemático que o sistema consegue estimar as movimentações ocorridas entre capturas de imagens), foi apresentado o método tradicional de se realizar a chamada Calibração da câmera, isto é, como se medem os parâmetros internos de uma câmera.

Esta calibração tradicional, como mostrado, necessita ser executada de maneira *offline* e com conhecimento a respeito da cena em que a câmera se encontra. Como a técnica de calibração é normalmente executada antes da aplicação que faz uso dos parâmetros calibrados, acaba que os valores medidos têm que ser utilizados como constantes. Deixou-se claro então que, apesar de os valores serem praticamente constantes, existem situações nas quais a câmera pode ser submetida que fazem com que os valores dos parâmetros se alterem, fazendo com que seja necessário executar a calibração da câmera repetidas vezes a fim de se manter um conhecimento preciso a respeito dos parâmetros internos.

Neste contexto, o leitor foi levado a perceber que a existência de uma técnica de calibração de câmera que seja executada em paralelo com a aplicação que faz uso dos parâmetros internos da câmera é muito bem vinda, e se esta técnica puder ser executada sem que se precise ter qualquer conhecimento a cerca do ambiente em que a câmera se encontra é melhor ainda. Dessa forma, ao final do capítulo 2, além de se conhecer mais

profundamente a teoria por trás da Localização Visual Inercial, foi possível perceber qual foi a motivação para os algoritmos desenvolvidos nesta pesquisa, bem como seus possíveis impactos positivos para a área da robótica móvel.

Partindo então do embasamento teórico mostrado, e claro, da motivação por trás do trabalho, o capítulo 3 apresenta ao leitor as duas técnicas desenvolvidas ao longo da pesquisa. A primeira delas consistiu em um método baseado no Filtro de Kalman Estendido de se realizar a chamada Auto-Calibração da câmera com auxílio de uma UMI, isto é, uma técnica que, a partir das imagens capturadas pela câmera e das leituras dos giroscópios de uma UMI fixamente acoplada à câmera, conseguia estimar os parâmetros internos da câmera de maneira *online* e sem qualquer conhecimento prévio a respeito do ambiente.

Claro que esta técnica consistiu em uma primeira tentativa, mas os resultados obtidos, como o mostrado no capítulo 4, para um experimento no mundo real foram bastante animadores. O leitor é levado a entender que o sistema foi capaz de realizar o seu objetivo de estimar os parâmetros internos e isso fazendo uso de um único ponto comum às imagens capturadas. Esta técnica de auto-calibração rendeu um artigo publicado e foi o pontapé inicial para a segunda linha de pesquisa desta dissertação. Baseado nas análises de desempenho do primeiro algoritmo e nas sugestões de melhorias percebidas, foi apresentado o desenvolvimento do algoritmo que tem como objetivo estimar a localização e orientação de um agente robótico a partir das imagens capturadas e das leituras dos sensores da UMI, isto é, um sistema de Localização Visual Inercial, mas que, paralelamente a estimação da orientação e posição do robô, fosse capaz de realizar a auto-calibração dos parâmetros internos da câmera utilizada.

Este segundo algoritmo foi chamado de LOVIACC e teve sua metodologia detalhada na segunda metade do capítulo 3. O leitor pôde entender como o conjunto de dois Filtros de Kalman Unscented rodando em paralelo conseguem realizar a Auto-Calibração da câmera e também a fusão sensorial entre Odometria Visual e UMI. Ainda foi possível perceber como os parâmetros serem sempre atualizados em tempo real ajudam a Odometria Visual, e conseqüentemente a etapa de fusão sensorial, a realizar estimações mais precisas.

O capítulo 4 apresentou ainda resultados de experimentos que comprovaram o funcionamento do algoritmo proposto. Como pôde-se perceber, os experimentos consistiram em situações plausíveis e apenas hipotéticas, nas quais os parâmetros da câmera variavam conforme uma função degrau ou uma função senoidal. Em ambos os casos o sistema LOVIACC se mostrou capaz de atingir seus objetivos, conseguindo realizar uma estimação dos parâmetros internos da câmera em conjunto com a localização do agente robótico.

Ao final de todas as apresentações dos resultados obtidos, ficou evidente que o sistema LOVIACC projetado tem o potencial para ser considerado uma alternativa viável para se realizar a localização de um robô em paralelo com a auto-calibração de uma câmera por ele portada, especialmente em situações nas quais a câmera possa estar sujeita a riscos de alteração dos valores dos parâmetros internos.

Assim, espera-se que tenha ficado claro, ao longo de todo o texto da dissertação, qual foi a linha evolutiva da pesquisa realizada. Espera-se também que o leitor consiga ter compreendido, a partir de todas as explicações apresentadas, as motivações que guiaram todo este trabalho e que seja possível enxergar a contribuição que um sistema como o LOVIACC pode trazer para a robótica móvel.

## 5.2 PROPOSTAS PARA TRABALHOS FUTUROS

Espera-se que o presente trabalho tenha sido capaz de deixar evidente que a técnica de Localização Visual Inercial com Auto-Calibração de Câmera desenvolvida seja uma alternativa utilizável para aplicações em robótica móvel. Mas ainda ao longo do texto, ficou claro que a técnica faz uso de sensores inerciais (UMI) e da estimação incremental da Odometria Visual. Desta forma, passado um longo período de estimação da localização com esta técnica, será obtida uma estimação com um nível de confiança cada vez mais baixo.

Isso pode ser contornado se, além desses dois sensores, for possível fazer uso de um sensor que traga informações absolutas a respeito da posição do robô, como por exemplo um sensor GPS. Sendo assim, essa é uma primeira sugestão para trabalhos futuros, que seja implementado, em paralelo ao LOVIACC, uma fusão sensorial que faça uso de um GPS. Dessa forma, a técnica até deixaria de ser entendida apenas como uma Localização Visual Inercial, mas poderia vir a ser um arcabouço mais robusto de estimação da posição e orientação de robôs móveis.

Outra sugestão que os autores fazem para possíveis trabalhos futuros é um estudo a cerca do desempenho computacional do algoritmo de LOVIACC. Para isto, pode-se implementar o método em uma linguagem como C para que o código possa ser executado em tempo real, embarcado em um agente robótico. Tem-se evidências de que o método é aplicável em tempo real, uma vez que o tempo de execução das simulações foi inferior ao tempo simulado. Mas seria muito bom ter essa comprovação baseada em um experimento no mundo real.

Ainda a respeito do algoritmo LOVIACC, percebe-se que o método fez uso do conhecimento da transformação de corpo rígido existente entre a câmera e a UMI. Uma outra sugestão para possíveis novos trabalhos é que se tente realizar a calibração entre os dois sensores de maneira *online*, isto é, que o algoritmo também consiga estimar qual é a rotação e translação existentes entre os dois sensores, deixando a execução do método com menos pré-requisitos.



## Referências Bibliográficas

---

- [1] S. Thrun, “Toward robotic cars,” *Communications of the ACM*, vol. 53, no. 4, p. 99, Apr. 2010. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1721654.1721679>
- [2] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, “1 year, 1000 km: The Oxford RobotCar dataset,” *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, Jan. 2017. [Online]. Available: <http://journals.sagepub.com/doi/10.1177/0278364916679498>
- [3] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun, “Towards fully autonomous driving: Systems and algorithms,” in *2011 IEEE Intelligent Vehicles Symposium (IV)*. Baden-Baden, Germany: IEEE, Jun. 2011, pp. 163–168. [Online]. Available: <http://ieeexplore.ieee.org/document/5940562/>
- [4] C. Forster, M. Pizzoli, and D. Scaramuzza, “SVO: Fast semi-direct monocular visual odometry,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. Hong Kong, China: IEEE, May 2014, pp. 15–22. [Online]. Available: <http://ieeexplore.ieee.org/document/6906584/>
- [5] P. Santana, “Filtragem estocástica para sistemas híbridos e suas aplicações em robótica aérea,” Ph.D. dissertation, UnB, Brasília, 2011. [Online]. Available: <http://repositorio.unb.br/handle/10482/8759>
- [6] H. B. Mitchell, *Multi-sensor data fusion: an introduction ; with 59 tables*. Berlin: Springer, 2007, oCLC: 255665904.
- [7] D. Simon, *Optimal state estimation: Kalman, H [infinity] and nonlinear approaches*. Hoboken, N.J: Wiley-Interscience, 2006, oCLC: ocm64084871.
- [8] S. Thrun, “Probabilistic robotics,” *Communications of the ACM*, vol. 45, no. 3, Mar. 2002. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=504729.504754>
- [9] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, “Robust visual inertial odometry using a direct EKF-based approach,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Hamburg, Germany: IEEE, Sep. 2015, pp. 298–304. [Online]. Available: <http://ieeexplore.ieee.org/document/7353389/>
- [10] F. T. Gonzalez and J. Andrade-Cetto, “Title: Visual Inertial Odometry for Mobile Robotics,” Ph.D. dissertation, Universitat Politècnica de Catalunya, 2015.
- [11] J. Gui, D. Gu, S. Wang, and H. Hu, “A review of visual inertial odometry from filtering and optimisation perspectives,” *Advanced Robotics*, vol. 29, no. 20, pp. 1289–1301, Oct. 2015. [Online]. Available: <http://www.tandfonline.com/doi/full/10.1080/01691864.2015.1057616>
- [12] E. Hong and J. Lim, “Visual inertial odometry using coupled nonlinear optimization,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Vancouver, BC: IEEE, Sep. 2017, pp. 6879–6885. [Online]. Available: <http://ieeexplore.ieee.org/document/8206610/>
- [13] Q. Li, H. Gu, C. Han, W. Gong, S. Song, and M. Q.-H. Meng, “Robust visual inertial monocular using nonlinear optimization,” in *2017 IEEE International Conference on Information*

- and Automation (ICIA)*. Macau SAR, China: IEEE, Jul. 2017, pp. 483–488. [Online]. Available: <http://ieeexplore.ieee.org/document/8078956/>
- [14] G. Panahandeh and P. Handel, “Planar-based visual inertial navigation,” in *2015 International Conference on Location and GNSS (ICL-GNSS)*. Gothenburg, Sweden: IEEE, Jun. 2015, pp. 1–8. [Online]. Available: <http://ieeexplore.ieee.org/document/7217139/>
- [15] D. Scaramuzza and F. Fraundorfer, “Visual Odometry [Tutorial],” *IEEE Robotics & Automation Magazine*, vol. 18, no. 4, pp. 80–92, Dec. 2011. [Online]. Available: <http://ieeexplore.ieee.org/document/6096039/>
- [16] F. Fraundorfer and D. Scaramuzza, “Visual Odometry : Part II: Matching, Robustness, Optimization, and Applications,” *IEEE Robotics & Automation Magazine*, vol. 19, no. 2, pp. 78–90, Jun. 2012. [Online]. Available: <http://ieeexplore.ieee.org/document/6153423/>
- [17] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, “SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems,” *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 249–265, Apr. 2017. [Online]. Available: <http://ieeexplore.ieee.org/document/7782863/>
- [18] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge: Cambridge University Press, 2004, oCLC: 804793563. [Online]. Available: <http://public.ebib.com/choice/publicfullrecord.aspx?p=256634>
- [19] D. A. Forsyth and J. Ponce, *Computer Vision - A Modern Approach, Second Edition*. Pitman, 2012.
- [20] J. Heikkila and O. Silven, “A four-step camera calibration procedure with implicit image correction,” in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. San Juan, Puerto Rico: IEEE Comput. Soc, 1997, pp. 1106–1112. [Online]. Available: <http://ieeexplore.ieee.org/document/609468/>
- [21] R. Hartley, “In defence of the 8-point algorithm,” in *Proceedings of IEEE International Conference on Computer Vision*. Cambridge, MA, USA: IEEE Comput. Soc. Press, 1995, pp. 1064–1070. [Online]. Available: <http://ieeexplore.ieee.org/document/466816/>
- [22] M. Li, H. Yu, X. Zheng, and A. I. Mourikis, “High-fidelity sensor modeling and self-calibration in vision-aided inertial navigation,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. Hong Kong, China: IEEE, May 2014, pp. 409–416. [Online]. Available: <http://ieeexplore.ieee.org/document/6906889/>
- [23] A. Borges Pio and G. Araujo Borges, “Online Inertial-Aided Monocular Camera Self-Calibration,” in *2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE)*, Joao Pessoa, Nov. 2018, pp. 100–105. [Online]. Available: <https://ieeexplore.ieee.org/document/8588534/>
- [24] H. S. Hughes, *Sensory Exotica: A World Beyond Human Experience*. MIT Press, 2001.
- [25] P. Corke, J. Lobo, and J. Dias, “An Introduction to Inertial and Visual Sensing,” *The International Journal of Robotics Research*, vol. 26, no. 6, pp. 519–535, Jun. 2007. [Online]. Available: <http://journals.sagepub.com/doi/10.1177/0278364907079279>

- [26] J. Kelly and G. S. Sukhatme, "Visual-Inertial Sensor Fusion: Localization, Mapping and Sensor-to-Sensor Self-calibration," *The International Journal of Robotics Research*, vol. 30, no. 1, pp. 56–79, Jan. 2011. [Online]. Available: <http://journals.sagepub.com/doi/10.1177/0278364910382802>
- [27] J. R. Raol, *Multi-sensor data fusion with MATLAB*, ser. Engineering. Boca Raton, Fla.: CRC Press, 2010, oCLC: 845596120.
- [28] O. J. Woodman, "An introduction to inertial navigation," p. 37.
- [29] D. H. Titterton and J. L. Weston, *Strapdown inertial navigation technology*, 2nd ed., ser. IET radar, sonar, navigation and avionics series. Stevenage: The Inst. of Engineering and Technology, 2009, no. 17, oCLC: 838448572.
- [30] M. Achtelik, M. Achtelik, S. Weiss, and R. Siegwart, "Onboard IMU and monocular vision based control for MAVs in unknown in- and outdoor environments," in *2011 IEEE International Conference on Robotics and Automation*. Shanghai, China: IEEE, May 2011, pp. 3056–3063. [Online]. Available: <http://ieeexplore.ieee.org/document/5980343/>
- [31] A. T. Erdem and A. O. Ercan, "Fusing Inertial Sensor Data in an Extended Kalman Filter for 3d Camera Tracking," *IEEE Transactions on Image Processing*, vol. 24, no. 2, pp. 538–548, Feb. 2015. [Online]. Available: <http://ieeexplore.ieee.org/document/6983575/>
- [32] P. Corke, *Robotics, Vision and Control - Fundamental Algorithms in MATLAB®*, ser. Springer Tracts in Advanced Robotics. Springer, 2011, vol. 73.
- [33] R. Valenti, I. Dryanovski, and J. Xiao, "Keeping a Good Attitude: A Quaternion-Based Orientation Filter for IMUs and MARGs," *Sensors*, vol. 15, no. 8, pp. 19 302–19 330, Aug. 2015. [Online]. Available: <http://www.mdpi.com/1424-8220/15/8/19302>
- [34] J. Sola, "Quaternion kinematics for the error-state KF," p. 73.
- [35] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1, June 2004, pp. I–I.
- [36] J. Engel, J. Sturm, and D. Cremers, "Semi-dense Visual Odometry for a Monocular Camera," in *2013 IEEE International Conference on Computer Vision*. Sydney, Australia: IEEE, Dec. 2013, pp. 1449–1456. [Online]. Available: <http://ieeexplore.ieee.org/document/6751290/>
- [37] J.-S. Hu and M.-Y. Chen, "A sliding-window visual-IMU odometer based on tri-focal tensor geometry," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. Hong Kong, China: IEEE, May 2014, pp. 3963–3968. [Online]. Available: <http://ieeexplore.ieee.org/document/6907434/>
- [38] M. Jaimez and J. Gonzalez-Jimenez, "Fast Visual Odometry for 3-D Range Sensors," *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 809–822, Aug. 2015. [Online]. Available: <http://ieeexplore.ieee.org/document/7119600/>
- [39] G. Martinez, "Monocular visual odometry from frame to frame intensity differences for planetary exploration mobile robots," in *2013 IEEE Workshop on Robot Vision (WORV)*. Clearwater Beach, FL: IEEE, Jan. 2013, pp. 54–59. [Online]. Available: <http://ieeexplore.ieee.org/document/6521914/>

- [40] A. Martinelli, "Vision and IMU Data Fusion: Closed-Form Solutions for Attitude, Speed, Absolute Scale, and Bias Determination," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 44–60, Feb. 2012. [Online]. Available: <http://ieeexplore.ieee.org/document/5959226/>
- [41] G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart, "Fusion of IMU and Vision for Absolute Scale Estimation in Monocular SLAM," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1-4, pp. 287–299, Jan. 2011. [Online]. Available: <http://link.springer.com/10.1007/s10846-010-9490-z>
- [42] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004. [Online]. Available: <http://link.springer.com/10.1023/B:VISI.0000029664.99615.94>
- [43] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features," *Computer Vision - ECCV 2006*, vol. 3951, pp. 404–417, 2006.
- [44] E. Rosten and T. Drummond, "Machine Learning for High-Speed Corner Detection," in *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, vol. 3951, pp. 430–443.
- [45] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Proceedings of the 2011 International Conference on Computer Vision*, ser. ICCV '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 2564–2571. [Online]. Available: <http://dx.doi.org/10.1109/ICCV.2011.6126544>
- [46] J. K. Suhr, "Kanade-Lucas-Tomasi (KLT) Feature Tracker," *Computer Vision*, p. 36, 2009.
- [47] J.-Y. Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm," p. 9.
- [48] D. Nister and S. Corporation, "An Efficient Solution to the Five-Point Relative Pose Problem," p. 17.
- [49] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981. [Online]. Available: <http://doi.acm.org/10.1145/358669.358692>
- [50] G. Sibley, "A Sliding Window Filter for SLAM," p. 17.
- [51] G. Welch and G. Bishop, "An introduction to the kalman filter," Chapel Hill, NC, USA, Tech. Rep., 1995.
- [52] M. Alatise and G. Hancke, "Pose Estimation of a Mobile Robot Based on Fusion of IMU Data and Vision Data Using an Extended Kalman Filter," *Sensors*, vol. 17, no. 10, p. 2164, Sep. 2017. [Online]. Available: <http://www.mdpi.com/1424-8220/17/10/2164>
- [53] A. Concha, G. Loianno, V. Kumar, and J. Civera, "Visual-inertial direct SLAM," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. Stockholm, Sweden: IEEE, May 2016, pp. 1331–1338. [Online]. Available: <http://ieeexplore.ieee.org/document/7487266/>
- [54] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-Manifold Preintegration for Real-Time Visual-Inertial Odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, Feb. 2017. [Online]. Available: <https://ieeexplore.ieee.org/document/7557075/>

- [55] —, “IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation,” in *Robotics: Science and Systems XI*. Robotics: Science and Systems Foundation, Jul. 2015. [Online]. Available: <http://www.roboticsproceedings.org/rss11/p06.pdf>
- [56] G. Huang, M. Kaess, and J. J. Leonard, “Towards consistent visual-inertial navigation,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. Hong Kong, China: IEEE, May 2014, pp. 4926–4933. [Online]. Available: <http://ieeexplore.ieee.org/document/6907581/>
- [57] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, “Keyframe-based visual-inertial odometry using nonlinear optimization,” *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, Mar. 2015. [Online]. Available: <http://journals.sagepub.com/doi/10.1177/0278364914554813>
- [58] M. Li and A. I. Mourikis, “High-precision, consistent EKF-based visual-inertial odometry,” *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, May 2013. [Online]. Available: <http://journals.sagepub.com/doi/10.1177/0278364913481251>
- [59] T. Lupton and S. Sukkarieh, “Visual-Inertial-Aided Navigation for High-Dynamic Motion in Built Environments Without Initial Conditions,” *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 61–76, Feb. 2012. [Online]. Available: <http://ieeexplore.ieee.org/document/6092505/>
- [60] A. Martinelli, “Closed-Form Solution of Visual-Inertial Structure from Motion,” *International Journal of Computer Vision*, vol. 106, no. 2, pp. 138–152, Jan. 2014. [Online]. Available: <http://link.springer.com/10.1007/s11263-013-0647-7>
- [61] V. Peretroukhin, L. Clement, M. Giamou, and J. Kelly, “PROBE: Predictive robust estimation for visual-inertial navigation,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Hamburg, Germany: IEEE, Sep. 2015, pp. 3668–3675. [Online]. Available: <http://ieeexplore.ieee.org/document/7353890/>
- [62] J.-P. Tardif, M. George, M. Laverne, A. Kelly, and A. Stentz, “A new approach to vision-aided inertial navigation,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Taipei: IEEE, Oct. 2010, pp. 4161–4168. [Online]. Available: <http://ieeexplore.ieee.org/document/5651059/>
- [63] Z. Yang and S. Shen, “Monocular Visual-Inertial State Estimation With Online Initialization and Camera-IMU Extrinsic Calibration,” *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 1, pp. 39–51, Jan. 2017. [Online]. Available: <http://ieeexplore.ieee.org/document/7463059/>
- [64] R. Y. Tsai, “A Versatile Camera Calibration Techniaue for High-Accuracy 3d Machine Vision Metrology Using Off-the-shelf TV Cameras and Lenses,” p. 22.
- [65] C. Harris and M. Stephens, “A combined corner and edge detector,” in *In Proc. of Fourth Alvey Vision Conference*, 1988, pp. 147–151.
- [66] “Single camera calibrator app - matlab and simulink - mathworks,” 2019, accessed: 10-06-2019. [Online]. Available: <https://www.mathworks.com/help/vision/ug/single-camera-calibrator-app.html>
- [67] “Camera calibration with opencv - opencv 2.4.13.7 documentation,” 2019, accessed: 10-06-2019. [Online]. Available: [https://docs.opencv.org/2.4/doc/tutorials/calib3d/camera\\_calibration/camera\\_calibration.html](https://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html)
- [68] “triangulate - mathworks,” 2019, accessed: 10-06-2019. [Online]. Available: <https://www.mathworks.com/help/vision/ref/triangulate.html>

- [69] A. S. Corporation and A. Gelb, Eds., *Applied optimal estimation*. Cambridge, Mass: M.I.T. Press, 1974.
- [70] D. Aufderheide and W. Krybus, "Towards real-time camera egomotion estimation and three-dimensional scene acquisition from monocular image streams," in *2010 International Conference on Indoor Positioning and Indoor Navigation*. Zurich, Switzerland: IEEE, Sep. 2010, pp. 1–10. [Online]. Available: <http://ieeexplore.ieee.org/document/5648008/>
- [71] J. Bergen, "Sarnoff Corporation CN5300, Princeton NJ 08530 USA," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, p. 8, 2004.
- [72] F. Bonin-Font, A. Ortiz, and G. Oliver, "Visual Navigation for Mobile Robots: A Survey," *Journal of Intelligent and Robotic Systems*, vol. 53, no. 3, pp. 263–296, Nov. 2008. [Online]. Available: <http://link.springer.com/10.1007/s10846-008-9235-4>
- [73] A. Buyval and M. Gavrilencov, "Vision-based pose estimation for indoor navigation of unmanned micro aerial vehicle based on the 3d model of environment," p. 4, 2015.
- [74] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016. [Online]. Available: <http://ieeexplore.ieee.org/document/7747236/>
- [75] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007. [Online]. Available: <http://ieeexplore.ieee.org/document/4160954/>
- [76] F. Dellaert and M. Kaess, "Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing," *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, Dec. 2006. [Online]. Available: <http://journals.sagepub.com/doi/10.1177/0278364906072768>
- [77] P. Dempsey, "Is smartphone innovation slowing down?" p. 2.
- [78] J. J. Engel, "Autonomous Camera-Based Navigation of a Quadcopter," p. 103.
- [79] J. Engel, J. Sturm, and D. Cremers, "Camera-based navigation of a low-cost quadcopter," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vilamoura-Algarve, Portugal: IEEE, Oct. 2012, pp. 2815–2821. [Online]. Available: <http://ieeexplore.ieee.org/document/6385458/>
- [80] R. Goebel, R. G. Sanfelice, and A. R. Teel, "Hybrid dynamical systems," *IEEE Control Systems*, vol. 29, no. 2, pp. 28–93, Apr. 2009. [Online]. Available: <https://ieeexplore.ieee.org/document/4806347/>
- [81] A. Leon-Garcia and A. Leon-Garcia, *Probability, statistics, and random processes for electrical engineering*, 3rd ed. Upper Saddle River, NJ: Pearson/Prentice Hall, 2008, oCLC: ocn181079252.
- [82] M. Li and A. I. Mourikis, "3-D motion estimation and online temporal calibration for camera-IMU systems," in *2013 IEEE International Conference on Robotics and Automation*. Karlsruhe, Germany: IEEE, May 2013, pp. 5709–5716. [Online]. Available: <http://ieeexplore.ieee.org/document/6631398/>
- [83] J. Lim, N. Barnes, and Hongdong Li, "Estimating Relative Camera Motion from the Antipodal-Epipolar Constraint," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 10, pp. 1907–1914, Oct. 2010. [Online]. Available: <http://ieeexplore.ieee.org/document/5477425/>

- [84] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, “Estimation of IMU and MARG orientation using a gradient descent algorithm,” in *2011 IEEE International Conference on Rehabilitation Robotics*. Zurich: IEEE, Jun. 2011, pp. 1–7. [Online]. Available: <http://ieeexplore.ieee.org/document/5975346/>
- [85] A. I. Mourikis and S. I. Roumeliotis, “A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation,” in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. Rome, Italy: IEEE, Apr. 2007, pp. 3565–3572. [Online]. Available: <http://ieeexplore.ieee.org/document/4209642/>
- [86] J. Nikolic, M. Burri, I. Gilitschenski, J. Nieto, and R. Siegwart, “Non-Parametric Extrinsic and Intrinsic Calibration of Visual-Inertial Sensor Systems,” *IEEE Sensors Journal*, vol. 16, no. 13, pp. 5433–5443, Jul. 2016. [Online]. Available: <http://ieeexplore.ieee.org/document/7457220/>
- [87] F. Nobre, M. Kasper, and C. Heckman, “Drift-correcting self-calibration for visual-inertial SLAM,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. Singapore, Singapore: IEEE, May 2017, pp. 6525–6532. [Online]. Available: <http://ieeexplore.ieee.org/document/7989771/>
- [88] I. Parra Alonso, D. F. Llorca, M. Gavilan, S. Á. Pardo, M. Á. Garcia-Garrido, L. Vlacic, and M. Á. Sotelo, “Accurate Global Localization Using Visual Odometry and Digital Maps on Urban Environments,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1535–1545, Dec. 2012. [Online]. Available: <http://ieeexplore.ieee.org/document/6192327/>
- [89] P. H. R. Q. A. Santana, G. A. Borges, and J. Y. Ishihara, “Hybrid data fusion for 3d localization under heavy disturbances,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Taipei: IEEE, Oct. 2010, pp. 2425–2430. [Online]. Available: <http://ieeexplore.ieee.org/document/5650178/>
- [90] T. Schneider, M. Li, M. Burri, J. Nieto, R. Siegwart, and I. Gilitschenski, “Visual-inertial self-calibration on informative motion segments,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. Singapore, Singapore: IEEE, May 2017, pp. 6487–6494. [Online]. Available: <http://ieeexplore.ieee.org/document/7989766/>
- [91] S. Song, M. Chandraker, and C. C. Guest, “High Accuracy Monocular SFM and Scale Correction for Autonomous Driving,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 4, pp. 730–743, Apr. 2016. [Online]. Available: <http://ieeexplore.ieee.org/document/7206590/>
- [92] G. Vogiatzis and C. Hernández, “Video-based, real-time multi-view stereo,” *Image and Vision Computing*, vol. 29, no. 7, pp. 434–441, Jun. 2011. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0262885611000138>
- [93] M. Whitty, “Robotics, Vision and Control. Fundamental Algorithms in MATLAB,” *Industrial Robot: An International Journal*, vol. 39, no. 6, p. ir.2012.04939faa.005, Oct. 2012. [Online]. Available: <http://www.emeraldinsight.com/doi/10.1108/ir.2012.04939faa.005>
- [94] J. R. Winkler, “Numerical recipes in C: The art of scientific computing, second edition,” *Endeavour*, vol. 17, no. 4, p. 201, Jan. 1993. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/016093279390069F>
- [95] H. Yu and A. I. Mourikis, “Edge-based visual-inertial odometry,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Vancouver, BC: IEEE, Sep. 2017, pp. 6670–6677. [Online]. Available: <http://ieeexplore.ieee.org/document/8206582/>

- [96] K. Granstrom, S. Renter, M. Fatemi, and L. Svensson, “Pedestrian tracking using Velodyne data — Stochastic optimization for extended object tracking,” in *2017 IEEE Intelligent Vehicles Symposium (IV)*. Los Angeles, CA, USA: IEEE, Jun. 2017, pp. 39–46. [Online]. Available: <http://ieeexplore.ieee.org/document/7995696/>
- [97] Y. Lu, J. Huang, Y.-T. Chen, and B. Heisele, “Monocular localization in urban environments using road markings,” in *2017 IEEE Intelligent Vehicles Symposium (IV)*. Los Angeles, CA, USA: IEEE, Jun. 2017, pp. 468–474. [Online]. Available: <http://ieeexplore.ieee.org/document/7995762/>