Cesar Dantas de Castro

Sistema de navegação para veículos robóticos aéreos baseado na observação e mapeamento do ambiente

Dissertação de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Engenharia de Computação.

Orientador: Paulo Augusto Valente Ferreira Co-orientador: Alessandro Corrêa Victorino Co-orientador: Samuel Siqueira Bueno

Campinas, SP 2007

Este exemplar corresponde à redação final da tese defendida por: CESAR DANTAS DE CATTLO
Julgada em 24/04/2007
Orientador

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

_

C279s	Castro, Cesar Sistema de navegação para veículos robóticos aéreos baseado na observação e mapeamento do ambiente / Cesar Dantas de Castro. – Campinas, SP: [s.n.], 2007.
	Orientador: Paulo Augusto Valente Ferreira, Alessandro Corrêa Victorino, Samuel Siqueira Bueno. Dissertação (Mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.
	1. Navegação de robôs móveis. 2. Localização. 3. Mapeamento do solo. 4. Kalman, Filtragem de. 5. Robôs móveis. I. Ferreira, Paulo Augusto Valente. II. Victorino, Alessandro Corrêa. III. Bueno, Samuel Siqueira. IV. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. V. Título

Navigation system for aerial robotic vehicles based on the
observation and mapping of the environment at the School of
Electrical and Computer Engineering
SLAM, Localization, Mapping, Kalman filtering, Robotic
airships.
Automação
Mestre em Engenharia Elétrica
Basílio Ernesto de Almeida Milani, Geovany Araújo Borges e
Rafael Santos Mendes
24/04/2007
Engenharia Elétrica

Resumo

Este trabalho disserta sobre o desenvolvimento e a implementação de um sistema de localização e mapeamento simultâneos (SLAM) para um veículo robótico aéreo. Utilizando tal sistema, um robô que sobrevoe determinada área, até então desconhecida, deve ser capaz de conhecer sua postura no ambiente e mapeá-lo, sem o auxílio de mapas ou outras informações externas. Para alcançar este objetivo, o sistema recebe informações de uma unidade de medição inercial e de uma câmera, que observa características do ambiente e, indiretamente, a posição e a atitude do robô. Para fundir as informações dos dois conjuntos sensoriais embarcados, é utilizada uma arquitetura baseada no filtro de Kalman estendido, que atua como um estimador tanto da localização do dirigível quanto do mapa. Este sistema representa um primeiro passo em direção a uma solução de SLAM em seis graus de liberdade para o Projeto AURORA, que visa o desenvolvimento de tecnologia em robótica aérea. Desta forma, a abordagem proposta é validada em um ambiente de simulação composto de sensores virtuais e do simulador dinâmico do projeto AURORA. Os resultados apresentados mostram a eficácia da metodologia.

Palavras-chave: SLAM, localização, mapeamento, filtro de Kalman, dirigíveis robóticos.

Abstract

This work addresses the development and implementation of a simultaneous localization and mapping (SLAM) system for aerial robotic vehicles. Through this system, a robot flying over an unknown region must be capable of detecting its position accurately and, at the same time, constructing a map of the environment without the help of maps or any other external information. To reach that goal, the system receives input data from an inertial measurement unit and a single camera, which observes features in the environment and, indirectly, the robot's position and attitude. The data from both onboard sensors are then fused using an architecture based on an extended Kalman filter, which acts as an estimator of the robot pose and the map. This system represents a first step towards a six degrees of freedom SLAM solution for Project AURORA, whose goal is the development of technology on aerial robotics. As such, the proposed methodology is validated in a simulation environment composed of virtual sensors and the aerial platform simulator of the AURORA project based on a realistic dynamic model. The reported results show the efficiency of the approach.

Keywords: SLAM, localization, mapping, Kalman filtering, robotic airships.

Agradecimentos

Ao meu orientador e co-orientadores. Ao Dr. Samuel Siqueira Bueno, agradeço pela ajuda sempre que precisei e pelos ensinamentos ao longo destes anos. Ao Prof. Dr. Alessandro Victorino agradeço por compartilhar seu conhecimento. Ao Prof. Dr. Paulo Valente pelo aprendizado. Obrigado a todos pela revisão e sugestões.

Aos meus amigos, pelos incontáveis momentos de descontração.

Aos meus pais, Marina e Durval, pelos conselhos e pela sabedoria. Vocês me fizeram o que eu sou.

À Mariana, pelo apoio incondicional ao longo desta jornada. Obrigado pela ajuda, compreensão e carinho. Ao seu lado, tudo é mais fácil.

À FAPESP, pelo apoio financeiro através do processo 03/13845-7.

Nihil est in intellectu quod non fuit prius in sensu. John Locke

Sumário

Li	Lista de Figuras		xi
Gl	Glossário x		
Li	sta de	e Símbolos	XV
1	Intr	odução	1
	1.1	Percepção, localização e mapeamento	1
		1.1.1 Localização	3
		1.1.2 Utilizando um mapa pré-definido	5
		1.1.3 Localização e Mapeamento Simultâneos	6
	1.2	O projeto AURORA	9
	1.3	Objetivos	11
	1.4	Contribuições	13
	1.5	Estrutura da dissertação	14
2	Auto	o-localização e mapeamento simultâneos	15
	2.1	Sistemas de Coordenadas	16
	2.2	O Estado do Sistema e a Representação de Incerteza	18
	2.3	Modelo cinemático não-linear	20
		2.3.1 Jacobiano em relação ao estado do dirigível	23
		2.3.2 Jacobiano em relação aos sinais da IMU	24
	2.4	Relação entre o mapa e a observação	24
		2.4.1 Jacobiano em relação à observação	26
		2.4.2 Jacobiano em relação ao estado do veículo	26
	2.5	O modelo não-linear de observação	27
		2.5.1 Jacobiano em relação ao estado	28
	2.6	Estimação	29

		2.6.1 Filtro de Kalman estendido	30
		2.6.2 Predição	33
		2.6.3 Atualização	34
		2.6.4 Correspondência entre observações e balizas	35
		2.6.5 Aumentando o Estado	38
	2.7	Conclusão	40
3	Estr	utura de software	41
	3.1	Biblioteca matemática	41
		3.1.1 Vetores, matrizes e quaternions	42
		3.1.2 Funções matemáticas	43
	3.2	Sistema de auto-localização e mapeamento	45
		3.2.1 Estimador	45
	3.3	Ambiente de simulação	46
		3.3.1 Dinâmica do dirigível	47
		3.3.2 Porta de visualização	48
		3.3.3 Sensores virtuais	50
	3.4	Integração	51
	3.5	Conclusão	54
4	Sim	ulação e resultados	55
	4.1	Configuração da simulação	55
		4.1.1 Condições do vôo	55
		4.1.2 Configuração dos sensores	56
	4.2	Desempenho do sistema	58
		4.2.1 Posição	58
		4.2.2 Atitude	60
		4.2.3 Velocidade	62
		4.2.4 Mapa	63
	4.3	Análise dos resultados	66
		4.3.1 Número de associações e a dimensionalidade do sistema	67
		4.3.2 Configuração do sistema e convergência	70
	4.4	Conclusão	73
5	Con	clusões	81

Referências bibliográficas

Tr	Trabalhos Publicados Pelo Autor 89		89
A	Gravação si	ncronizada de dados	91
	A.0.1	A biblioteca	92
	A.0.2	Estrutura de Software	93
B	Quaternions	3	95

Lista de Figuras

1.1	Exemplo de erro de odometria. O mapa, não utilizado para a navegação, mostra em	
	preto os obstáculos do ambiente. A linha mostra o trajeto estimado do robô utilizando	
	os odômetros. A maior fonte de erro é a estimativa da rotação, quando a derrapagem	
	das rodas é mais acentuada, resultando em grandes disparidades do outro lado do mapa.	4
1.2	Funcionamento da metodologia de SLAM. A observação das mesmas balizas em di-	
	ferentes instantes permite a correção dos erros de posicionamento	8
1.3	O dirigível AS800B, plataforma de desenvolvimento e validação experimental do	
	projeto AURORA.	10
2.1	Sistemas de coordenadas relevantes.	17
2.2	Considera-se que apenas uma rotação diferencia o referencial da IMU (B) e o refe-	
	rencial da câmera (C)	18
2.3	O modelo cinemático do dirigível utiliza $\mathbf{u}(k)$ e $\mathbf{w}(k)$ para levar de $\mathbf{x}_{v}(k-1)$ a $\mathbf{x}_{v}(k)$.	20
2.4	Os eixos do dirigível e a representação da direção dos sinais da IMU	21
2.5	O dirigível observa uma baliza	25
2.6	Estrutura do sistema de SLAM. Cada círculo representa uma etapa do sistema (aque-	
	las referentes ao filtro de Kalman estendido estão destacadas) e as setas indicam o	
	fluxo de dados	31
2.7	Exemplo 2D da correspondência das observações e da ambigüidade de predições e	
	observações. As predições $h(\cdot, \cdot)$ possuem elipses indicando o limite δ . As obser-	
	vações são representadas por $z_a e z_b$. As setas indicam a distância de Mahalanobis.	
	Percebe-se que \mathbf{z}_a é uma observação válida para as duas predições, enquanto \mathbf{z}_a e \mathbf{z}_b	
	podem ambas corresponder a $h(\mathbf{x}_v, \mathbf{x}_{mi})$	37
3.1	A figura mostra a representação dos vetores, matrizes e quaternions. A dupla imple-	
	mentação de matrizes e vetores agiliza o desenvolvimento e aumenta a velocidade das	
	operações envolvendo os vetores tridimensionais	42

3.2	Implementações da interface de função matemática. A generalidade obtida facilita	
	a implementação no dirigível real, já que os modelos dos sensores são passíveis de	
	alteração	43
3.3	Organização do sistema de SLAM. O Filtro de Kalman estende a interface do estima-	
	dor, que está contido na classe que representa o sistema de SLAM em si	46
3.4	Renderizações do POV-Ray.	48
3.5	Mais exemplos de renderizações do POV-Ray.	49
3.6	A projeção da baliza m_1 no plano imagem está fora dos limites da imagem, de modo	
	que a equação 3.3 é falsa. A baliza m_2 tem, no referencial da câmera, coordenada z	
	menor que a distância focal, de modo que a equação 3.2 não é verdadeira. A baliza	
	m_3 é visível na imagem da câmera	52
3.7	Estrutura global do ambiente de simulação e do módulo de SLAM: O simulador di-	
	nâmico e o renderizador funcionam como fontes de dados externas. A ponte entre o	
	ambiente de simulação e o estimador são apenas as leituras dos sensores, permitindo	
	que o ambiente de simulador seja substituído pelos sensores reais implementando	
	apenas a interface pré-estipulada.	53
4.1	Esquema do funcionamento dos sensores inercial e visual.	57
4.2	A linha azul mostra o trajeto real do dirigível e a linha verde mostra a estimativa da	
	posição do dirigível utilizando apenas os sensores inerciais. O erro aumenta sem limites.	59
4.3	A linha vermelha exibe a posição do dirigível conforme determinado pelo sistema	
	de auto-localização e mapeamento, enquanto em azul é mostrada a posição real. A	
	observação das balizas impede a divergência da estimativa.	60
4.4	Comparação direta das estimativas obtidas apenas da IMU (linha verde) e da IMU	
	juntamente com o sistema de visão (linha vermelha). O trajeto real é mostrado em azul.	61
4.5	Estimativas da posição para cada eixo do referencial de terra. Em verde (losangos), a	
	estimativa utilizando apenas a IMU, em vermelho (triângulos) a estimativa do sistema	
	de SLAM e em azul o valor real (círculos)	62
4.6	Erro em cada um dos eixos do sistema de coordenadas georreferenciado	63
4.7	Norma do erro de posicionamento do sistema de auto-localização	64
4.8	Valor de cada componente do quaternion que representa a orientação do dirigível.	
	Em azul (círculos) o estado real, em vermelho (triângulos) o estimado pelo sistema	
	de SLAM e em verde (losangos) o obtido apenas do INS	65
4.9	Seguindo a mesma notação de cores do gráfico anterior, são exibidos os valores da	
	atitude do dirigível em ângulos de Euler	66
4.10	Erros de orientação do sistema de SLAM em cada componente do quaternion	67

LISTA DE FIGURAS

4.11	Norma do erro da estimativa do quaternion. A variação acompanha a percebida na	
	posição.	68
4.12	Velocidade do dirigível em cada eixo do sistema georreferenciado. Em azul (círculos)	
	os valores reais; em verde (losangos) e vermelho (triângulos), respectivamente, a	
	estimativa dos valores utilizando apenas o INS e utilizando a IMU e o sensor visual	69
4.13	Erro presente na estimação da velocidade para cada um dos eixos	70
4.14	Norma do erro de estimação da velocidade.	71
4.15	Início do vôo: apenas algumas balizas detectadas (cruzes) do total de balizas existen-	
	tes (círculos).	72
4.16	Após 25 segundos de vôo: o dirigível evolui, as balizas já observadas diversas vezes	
	tiveram suas posições corrigidas.	73
4.17	Após 50 segundos de vôo: Novas balizas são detectadas e incluídas no mapa. Neste	
	trecho há um menor número de balizas associadas a cada iteração e o erro de posici-	
	onamento de algumas delas se torna visível	74
4.18	Após 75 segundos de vôo: todas as balizas já foram incluídas no mapa	74
4.19	Fim do experimento (130 segundos): apenas correções menores são feitas no mapa.	75
4.20	Média da norma do erro de mapeamento para as balizas presentes no mapa	75
4.21	Distância média para as balizas reais em cada um dos eixos do referencial de terra.	76
4.22	Resultado do mapeamento de outro ângulo, juntamente com a trajetória completa do	
	dirigível e sua estimativa	77
4.23	Número de correspondências a cada observação do ambiente	77
4.24		78
4.25	Seqüência correspondente à inovação no eixo x do referencial de câmera, da primeira	
	observação associada ao mapa a cada iteração	78
4.26	Seqüência correspondente à inovação no eixo y do referencial de câmera, da primeira	
	observação associada ao mapa a cada iteração	79
4.27	Seqüência correspondente à inovação no eixo z do referencial de câmera, da primeira	
	observação associada ao mapa a cada iteração	79
4.28	Esperança de q para 1, 5 e 10 execuções da simulação	80
4.29	Média de q para 10 execuções, juntamente com o intervalo de confiança de 95%	80
A.1	Estrutura do sistema de gravação	94

Lista de Símbolos

AURORA	-	<u>A</u> utonomous <u>U</u> nmanned, <u>R</u> emote Monitoring <u>R</u> obotic <u>A</u> irship
CenPRA	-	Centro de Pesquisas Renato Archer
GPS	-	Global Positioning System
DV	-	Digital Video
HALE	-	High-Altitude, Long-Endurance
UAV	-	Unmanned Aerial Vehicle
VANT	-	Veículo Aéreo Não Tripulado
IMU	-	Inertial Measurement Unit
SLAM	-	Simultaneous Localization and Mapping
KF	-	Kalman Filter
EKF	-	Extended Kalman Filter
INS	-	Inertial Navigation System
PD	-	Proporcional/Derivativo
PID	-	Proporcional/Integral/Derivativo
POV-Ray	-	Persistence of Vision Raytracer
F	-	Referencial de flutuação do dirigível
В	-	Referencial da IMU do dirigível, adotado como referencial do dirigível em si
С	-	Referencial da câmera embarcada no dirigível
W	-	Referencial geodésico, fixo
X	-	Estado do sistema
Â	-	Estimativa do estado do sistema
$\mathbf{X}_{\mathcal{V}}$	-	Estado do dirigível
$\mathbf{\hat{x}}_{v}$	-	Estimativa do estado do dirigível
X _m	-	Estado do mapa
$\hat{\mathbf{x}}_m$	-	Estimativa do estado do mapa
\mathbf{x}_{mi}	-	Estado da <i>i</i> -ésima baliza do mapa
$\hat{\mathbf{x}}_{mi}$	-	Estimativa do estado da <i>i</i> -ésima baliza do mapa
р	-	Vetor que representa posição do dirigível
p	-	Estimativa do vetor que representa posição do dirigível
q	-	Quaternion que representa a atitude do dirigível
Ŷ	-	Estimativa do quaternion que representa a atitude do dirigível
V	-	Vetor que representa a velocidade do dirigível
Ŷ	-	Estimativa do vetor que representa a velocidade do dirigível

Р	-	Matriz de covariância do estado do sistema
$f(\cdot, \cdot, \cdot)$	-	Modelo cinemático não-linear do dirigível
$\mathbf{u}(k)$	-	Sinais provenientes da IMU no instante k
$\alpha(k)$	-	Acelerações lineares no referencial do dirigível no instante k
$\omega(k)$	-	Taxas de rotação no referencial do dirigível no instante k
$\mathbf{w}(k)$	-	Ruído do modelo cinemático no instante k
w _p	-	Ruído do modelo cinemático em p
\mathbf{W}_{a}^{P}	-	Ruído do modelo cinemático em q
\mathbf{W}_{V}	-	Ruído do modelo cinemático em v
Δt	-	Intervalo de tempo
g	-	Vetor representando a aceleração causada pela gravidade
$\mathbf{\tilde{C}}_{\mathbf{X}}^{Y}$	-	Matriz que rotaciona da origem do referencial X para o Y
$\mathbf{R}_{X}^{\widehat{Y}}$	-	Função que rotaciona da origem do referencial X para o Y
$\mathbf{q}_X^{\widehat{Y}}$	-	Quaternion que rotaciona da origem do referencial X para o Y
$\mathbf{D}(k)$	-	Derivada de $\mathbf{q}(k)$ em relação aos ângulos de rotação da IMU
$\boldsymbol{\varepsilon}(k)$	-	Termo de correção da magnitude do quaternion $\mathbf{q}(k)$
ā	-	Conjugado do quaternion q
$h(\cdot)$	-	Modelo não-linear de observação
$g(\cdot)$	-	Inversa de $h(\cdot)$
$f_a(\cdot, \cdot)$	-	Função que aumenta o estado do sistema
\mathbf{Z}_i	-	Representação da <i>i</i> -ésima observação
σ^2	-	Variância
Q	-	Covariância do erro da leitura da IMU
R	-	Covariância do erro da leitura do sensor visual
v	-	Vetor de inovação
S	-	Covariância do vetor de inovação
χ^2	-	Distribuição qui-quadrado
$\gamma_{i,j}$	-	Distância de Mahalanobis entre a observação <i>i</i> e a predição <i>j</i>

Capítulo 1

Introdução

A presente dissertação trata do desenvolvimento e da implementação de uma metodologia de integração de informações sensoriais, visando a obtenção de uma estimativa confiável da postura – ou seja, a posição e a atitude – de um veículo robótico aéreo. Posto que a estratégia proposta utiliza não apenas sensores proprioceptivos (sensores inerciais) mas também um sensor exteroceptivo (câmera), sendo que todos estão embarcados no veículo, a observação do ambiente permite a construção incremental de um mapa dos arredores enquanto, ao mesmo tempo, a aeronave é posicionada dentro deste mapa, resultando em uma estratégia de mapeamento e auto-localização simultâneos.

Para um veículo autônomo, o posicionamento relativo no ambiente que o circunda é de essencial importância tanto para estratégias de navegação quanto para planejamento e tomada de decisões. Deste modo, a grande vantagem de simultaneamente determinar a posição da aeronave e realizar o mapeamento é permitir a navegação da aeronave sem nenhuma informação *a priori* do ambiente sobrevoado, o que aumenta drasticamente sua autonomia.

Este trabalho se insere no contexto do projeto AURORA, que visa o estabelecimento de tecnologia em robótica aérea autônoma tendo como plataforma um dirigível não tripulado. Considera-se, portanto, sem perda de generalidade, um dirigível robótico como veículo aéreo.

A fim de contextualizar o presente trabalho, esta seção apresenta inicialmente a problemática de percepção, localização e mapeamento, e sua evolução no âmbito da robótica; na seqüência, fornece-se uma visão resumida do Projeto AURORA. Finalmente, são explicitados os objetivos e contribuições do trabalho e a estrutura da dissertação.

1.1 Percepção, localização e mapeamento

"Não há nada na razão que não tenha, em primeiro lugar, sido experimentado pelos sentidos". A frase, atribuída ao filósofo empiricista John Locke, que viveu na inglaterra no século XVII, mostra que

o homem sempre questionou o funcionamento de seu mecanismo sensorial, reconhecendo-o como o princípio germinal do conhecimento. Ao menos desde Aristóteles, na grécia antiga, o estudo da percepção tem persistido, incorporando as mais variadas abordagens, com trabalhos realizados tanto nas áreas de biologia, fisiologia e neurociência quanto de filosofia e psicologia, como na filosofia da mente (Heil, 2004; Churchland, 1998) e na epistemologia empiricista de Hume, Locke e Berkeley (Honderich, 1995; Locke, 1973).

A procura da compreensão da relação entre sensação e conhecimento, que está na origem das ciências cognitivas, ainda na antiguidade levou à análise da forma como as informações de cada um dos sensores produz uma única percepção. Aristóteles defendia que todos os cinco sentidos convergiam em um só (Modrak, 1989) e, não apenas isso, mas refletiu também sobre como a força dos estímulos sensoriais influenciava a percepção final. De acordo com o filósofo grego, se, em um mesmo instante, dois ou mais estímulos sensoriais contraditórios fossem recebidos, o estímulo mais forte se sobreporia aos demais, que seriam suprimidos.

O conhecimento atual da fisiologia humana torna o problema de fusão sensorial ainda mais interessante, dado que os sensores produzem informações de maneiras completamente distintas: a audição detecta mudanças de pressão no tímpano, a visão detecta fótons na retina, o olfato reconhece moléculas específicas no bulbo olfativo. No entanto, mesmo em meio a esta diversidade, o sistema nervoso central consegue extrair dos dados os pontos em comum e produzir um único percepto. Estudos posteriores (Bower, 1974; Welch and Warren, 1986) indicam que a taxonomia da fusão sensorial pode ser dividida em níveis e que, em certo grau de interação, o cérebro quantifica a confiabilidade de cada estímulo sensorial e pondera esta incerteza para produzir uma única percepção intermediária entre as sensações conflitantes. Ao tentar identificar a localização de um objeto, a visão exerce maior influência. Já para identificar a origem de um som, a audição pode ser mais relevante.

Considere-se como exemplo o ato de estender a mão para agarrar uma bola. Neste simples gesto, está implícita a integração e a coordenação de diversas informações sensoriais: a sensação proprioceptiva da posição das articulações do braço e da mão, a percepção visual da posição do braço e da bola e a noção da movimentação que leva a mão até a bola. Se, após a análise inicial da situação, o movimento é realizado sem o auxílio da visão, percebe-se que tanto a estimativa da movimentação necessária quanto a própria coordenação dos movimentos do braço não são precisas. Por outro lado, utilizando as informações visuais ao longo do movimento, é possível, a partir da redundância dos dados provenientes dos sensores, reduzir drasticamente o erro de percepção e agarrar a bola com precisão.

O problema de navegação de um veículo robótico autônomo é análogo ao problema de agarrar uma bola com a mão. O robô precisa conhecer sua própria posição, a posição onde quer chegar e a relação entre as duas. Se apenas um sensor é utilizado para estimar estas informações, o sensor deve ser muito preciso, aumentando os gastos com a produção do sistema, ou a estimativa não será confiável. Por este motivo, assim como a visão auxilia a movimentação da mão, a utilização de um conjunto de sensores para a navegação ao mesmo tempo aumenta a precisão da estimativa e reduz o custo de construção. Dentro do conjunto sensorial de que um veículo autônomo deve dispor, o sistema de navegação é, talvez, o mais importante pois, sem ele, não há possibilidade de planejamento de movimento. O grau de precisão pode variar de acordo com a aplicação, mas é praticamente impossível desenvolver um veículo autônomo sem uma noção de posicionamento. Por este motivo, a comunidade científica tem, continuamente, desenvolvido novas e melhores estratégias para otimizar a determinação da postura do veículo, visando aumentar sua capacidade de navegação em todos os tipos de ambiente, sejam eles abertos ou fechados.

O restante da seção fornece uma visão geral dos esforços neste sentido. São descritas brevemente as diversas abordagens elaboradas para este fim, principiando com estratégias de localização e chegando até auto-localização e mapeamento simultâneos, atualmente considerada uma das metodologias mais eficientes para navegação.

1.1.1 Localização

Os primórdios dos sistemas de navegação em robótica datam da década de 1950, quando veículos na indústria podiam ser guiados por fios. O conceito evoluiu por volta de 1970, quando robôs passaram a ser capazes de seguir linhas pintadas no chão (Tsumura, 1986), eliminando a necessidade de infra-estrutura complexa. No entanto, estas estratégias limitavam a movimentação do veículo a um caminho pré-definido; fora dele, o robô ficava perdido. Não obstante, a intensa pesquisa desde então tornou o campo fértil de novas estratégias que deram início, efetivamente, à robótica autônoma. Gradualmente, os sistemas robóticos teleoperados tornaram-se sistemas telemonitorados.

Nos dias de hoje, uma das formas mais básicas de localização independente do ambiente se dá através da integração de estimativas da movimentação do veículo robótico. Esta técnica pode ser implementada utilizando qualquer sensor que forneça velocidades, acelerações ou deslocamentos, como por exemplo uma unidade de medição inercial (IMU – *Inertial Measurement Unit*) ou um odômetro. A cada nova informação obtida, os valores são integrados no tempo, se necessário, e acrescidos à estimativa da postura (definida como posição e atitude) atual, tal que a estimativa é obtida incrementalmente. Este tipo de estratégia resulta em um sistema de navegação inercial, também denominado INS (*Inertial Navigation System*). O problema deste tipo de sistema é que cada estimativa de mudança de postura inclui um componente de erro. Este erro, por sua vez, se acumula, como parte do processo de integração. Como pode-se observar na figura 1.1, extraída de (Thrun, 2002), ao longo do tempo a estimativa se torna cada vez menos precisa, de modo que a incerteza cresce constantemente. Por este motivo, depois de um certo intervalo de tempo as informações sensoriais não possuem mais



nenhuma utilidade para o posicionamento do veículo.

Fig. 1.1: Exemplo de erro de odometria. O mapa, não utilizado para a navegação, mostra em preto os obstáculos do ambiente. A linha mostra o trajeto estimado do robô utilizando os odômetros. A maior fonte de erro é a estimativa da rotação, quando a derrapagem das rodas é mais acentuada, resultando em grandes disparidades do outro lado do mapa.

Mesmo adotando sensores inerciais de grande precisão, o acúmulo de erro é apenas reduzido, e não eliminado. Isto significa que o tempo de operação eficiente pode ser estendido, mas o problema persiste, de forma que esta abordagem é imprópria para localização por longos períodos. No entanto, tipicamente, centrais inerciais e odômetros fornecem informações com freqüência superior a outros tipos de sensor. Aliando-se a isto o fato de que a curto prazo estes sensores possuem boa precisão, tais sensores constituem uma boa fonte auxiliar de navegação, desde que haja uma outra metodologia para sua constante correção.

Sem os problemas de degradação de sensores de integração, estratégias de navegação podem ser estruturadas utilizando sistemas de satélites para navegação global (GPS - *Global Positioning System*), que fornecem posicionamento absoluto. No entanto, muito embora a precisão da estimativa utilizando estes sensores não diminua com o tempo, sua precisão de curto prazo é pequena e a freqüência de atualização, baixa. Além disso GPS também está sujeito a problemas de disponibilidade, dado que é dependente do satélite e qualquer falha de comunicação inutiliza o sistema. A falha pode ser natural, causada por obstruções climáticas ou mesmo de prédios e vegetação, o que dificulta sua utilização em baixa altitude. Pesquisas recentes mostram que erupções solares podem comprometer seriamente

a precisão do GPS por intervalos de alguns minutos. Por outro lado, outro problema que pode surgir é a interrupção intencional do sinal do satélite.

Uma estratégia muito abordada pela comunidade científica é justamente a integração entre os sensores inerciais e o GPS (Snyder et al., 1992; Goel et al., 1999; Sukkarieh, 2000; Panzieri et al., 2002). Estas metodologias utilizam o sinal do GPS, menos freqüente, para corrigir os erros de posicionamento do sistema inercial. Como o GPS não deriva com o tempo, sua utilização garante a eficiência do sistema inercial mesmo em missões de longa duração. Por outro lado, o sistema é capaz de funcionar corretamente sem o auxílio do GPS por curtos períodos, tal que eventuais interrupções dos sinais dos satélites podem ser superadas.

Não obstante, muitas vezes a ausência do sinal da constelação de satélites é prolongada, deixando o sistema tão ineficaz quando se utilizasse apenas as informações inerciais, estando portanto sujeito a derivação da estimativa e incerteza crescente. Além disso, a fusão entre o GPS e o sistema inercial não fornece nenhuma informação sobre o ambiente que circunda o veículo robótico. Isto significa que ou o robô exigirá um outro sistema complementar para a análise do ambiente ou não poderá executar missões em que o conhecimento do posicionamento nos arredores seja necessário.

1.1.2 Utilizando um mapa pré-definido

Uma alternativa para a obtenção de informação do ambiente de atuação do veículo, e por conseguinte de localização absoluta no mesmo, é a utilização de um mapa construído *a priori*. Cabe aqui a descrição de mapa pertinente ao problema de localização robótica: um mapa é uma descrição do ambiente em termos dos vários objetos que podem ser detectados por um sensor exteroceptivo e que são apropriados para navegação (Guivante, 2002). Objetos não perceptíveis pelo sensor podem ser excluídos, assim como objetos perceptíveis mas não pertinentes. Muito embora um mapa com estas características pareça incompleto, ele é eficiente se o objetivo é localização.

Estratégias de localização baseadas em mapas pré-definidos tiveram início na década de 1970, com a utilização de balizas ativas no mapa, tal qual transdutores infra-vermelho (Giralt et al., 1979) ou ultra-sônicos (Kleeman, 1992). Muito embora o mapeamento de balizas ativas acarrete na necessidade de infra-estrutura e manutenção dispendiosos, muitas vezes este é o melhor método de detecção, de modo que conforme a necessidade de precisão e o meio de atuação do veículo o custo é justificado. Além disso, ao contrário do que acontece com as linhas de orientação, o robô pode se mover livremente pelo ambiente mapeado. Outras estratégias surgiram, utilizando balizas passivas (Brady, 1987), permitindo novamente a redução do custo de infra-estrutura. O próximo passo da evolução dos mapas para localização é a sua elaboração utilizando um ambiente não-estruturado; ou seja, os marcos do mapa são naturais do ambiente, e não inseridos artificialmente. Um exemplo desta abordagem consiste na utilização de um mapa preciso da métrica do ambiente, baseado nas plantas de construção de ambientes internos, para a navegação (Cox, 1991).

É interessante notar que o GPS é também uma forma de localização utilizando balizas ativas. Cada satélite da constelação envia uma identificação e um sinal temporal tal que, de posse das informações de quatro ou mais satélites, é possível calcular por triangulação a posição do receptor. No entanto, como a informação proveniente do sistema é o posicionamento literalmente global, não há nenhuma informação sobre o ambiente. Por outro lado, se um conjunto de coordenadas de latitude e longitude for construído para pontos específicos da área de atuação do robô, o GPS pode ser utilizado para localização neste mapa.

Seja qual for a estratégia utilizada, de posse do mapa do ambiente e de um sensor capaz de identificar regiões do mapa, como uma câmera ou sistema de sonares, o robô pode ser posicionado em relação às regiões identificadas (Borenstein and Koren, 1989). Esta correção de posicionamento pode ocorrer independentemente ou ser incluída no sistema de fusão sensorial.

Com a presença do mapa e do sensor que o reconhece, é possível, se necessário, abandonar algumas fontes de informação e utilizar: i) apenas o sensor exteroceptivo, desde que este seja preciso o bastante para as exigências de locomoção do veículo e que as regiões reconhecidas sejam próximas o bastante, evitando que o veículo se perca entre uma região e outra; ii) uma metodologia de fusão sensorial com outros sensores, como por exemplo uma central inercial ou odômetros, o que traz o benefício não apenas da estimação da postura do robô mesmo na ausência de informações do mapa, como também da redução da incerteza, advinda da combinação das informações sensoriais.

O problema de estratégias de navegação utilizando mapas pré-definidos é justamente a necessidade de que o ambiente seja explorado previamente. Isto implica que não apenas o local deve ser conhecido, como também os pontos marcantes da região devem ser mapeados, tal que quando o sensor os encontrar, o robô saiba onde está. Só então o robô será capaz de navegar de modo autônomo. A construção de um mapa *a priori* pode ser uma tarefa difícil e um mapa novo deve ser gerado para cada ambiente a ser circundado pelo robô. Além disso, o mapa é estático e não pode nem se adaptar a mudanças no ambiente nem ser incrementado caso o robô transite regiões ainda não exploradas.

A motivação para o desenvolvimento de estratégias de Auto-Localização e Mapeamento Simultâneos (SLAM) é justamente resolver os problemas do mapeamento prévio, permitindo ao robô transitar em um ambiente totalmente desconhecido mas, mesmo assim, obter o posicionamento do veículo em um mapa que é construído incrementalmente, conforme o robô explora a região.

1.1.3 Localização e Mapeamento Simultâneos

A origem de estratégias de SLAM se confunde com a do mapeamento robótico propriamente dito. Desta forma, pode-se considerar como o princípio da área de Localização e Mapeamento Simultâneos os algoritmos para a geração de grades de ocupação de Elfes, detalhados em (Elfes, 1987, 1989). Nesta abordagem, o mapa é representado por uma matriz, sendo que cada célula representa uma posição e indica em seu valor se a posição está ocupada por um objeto ou não. Esta estratégia foi utilizada em inúmeros sistemas de robótica autônoma, como por exemplo (Borenstein and Koren, 1989; Guzzoni et al., 1997; Burgard et al., 1999). Por outro lado, Chatila e Laumond (Chatila and Laumond, 1985) elaboravam, na mesma época, uma metodologia topológica para descrever a geometria do ambiente.

Embora os esforços de Elfes, Chatila e Laumond sejam de fundamental importância para a área de mapeamento, apenas com o trabalho de Smith, Self e Cheeseman (Smith et al., 1987, 1988), conhecido como *Mapeamento Estocástico*, um arcabouço estatístico rigoroso foi proposto para resolver simultaneamente o problema de mapear uma região e de localizar o robô em relação ao mapa crescente. A estratégia consiste em incluir no estado do sistema tanto a localização do veículo quanto o mapa em si e estimar não apenas o estado como também a incerteza a ele associada.

Desde então, diversas metodologias têm sido desenvolvidas para solucionar o problema de SLAM, sendo que uma das vertentes mais proeminentes envolve a utilização de um Filtro de Kalman, ou uma variação do mesmo, como estimador do estado do sistema, realizando a fusão sensorial entre os sensores envolvidos no processo (Csorba, 1997; Castellanos et al., 1999, 2001; Dissanayake et al., 2001).

Muito embora tenha principiado em ambientes fechados, estratégias de SLAM rapidamente evoluíram para espaços abertos, onde o tamanho do ambiente e, conseqüentemente, do mapa, se tornaram um problema. Diversas técnicas não apenas para gerenciamento de mapas como também para correspondência de balizas foram desenvolvidas, visando resolver os problemas causados pela dimensionalidade do mapa (Borges, 2002; Bailey, 2002; Guivante, 2002; Guivant and Nebot, 2003), como a utilização de representações híbridas métrico-topológicas, associadas a métodos de controle e navegação baseados no espaço sensorial (Victorino, 2002). Enfim, as estratégias de auto-localização e mapeamento atingiram robôs aéreos. Alguma soluções interessantes para o problema de SLAM foram propostas na comunidade científica internacional de pesquisa em robótica aérea, como a integração de informações inerciais com observações do ambiente feitas por uma câmera ou detector de distância. Estes trabalhos, bem como os realizados para veículos sub-aquáticos, resultaram nas primeiras implementações da metodologia em seis graus de liberdade (Ruiz et al., 2001; Kim, 2004; Langelaan and Rock, 2005; Wu et al., 2005).

Funcionamento

Localização e Mapeamento Simultâneos representam um problema de estimação estocástico cíclico: a localização das balizas ou outras características que compõem o mapa está sujeita aos erros de posicionamento do veículo robótico e vice-versa. No entanto, a observação das balizas permite a correção do posicionamento. Por sua vez, com o posicionamento corrigido, as novas observações das mesmas balizas permitem maior precisão no mapeamento. Conforme o ciclo se repete, a incerteza tanto da localização das balizas quanto da postura do veículo diminuem, atingindo um limite inferior.



Fig. 1.2: Funcionamento da metodologia de SLAM. A observação das mesmas balizas em diferentes instantes permite a correção dos erros de posicionamento.

A figura 1.2 mostra o funcionamento de uma estratégia de SLAM. Para a obtenção do ciclo descrito, basta que o veículo robótico seja equipado com um sensor exteroceptivo, capaz de observar características do ambiente (como a câmera da figura observa as balizas vermelhas). No instante k - 1 o robô detecta diversas balizas. No instante seguinte, k, a câmera mais uma vez observa as características do ambiente. Algumas das balizas observadas em k - 1 são detectadas novamente, enquanto que outras são observadas pela primeira vez. A existência de intersecção entre o conjunto de balizas observadas em k - 1 e em k permite:

- a observação dos erros da postura do dirigível, através da correlação entre as variáveis do mapa e do estado do veículo;
- correção do posicionamento balizas detectadas mais de uma vez.

No entanto, a câmera possui taxa de atualização restritiva, de modo que, no presente trabalho, além da câmera, são embarcados no veículo sensores inerciais, que podem operar em freqüência mais alta. Com isto, não apenas a observação das balizas mas também as informações da IMU aumentam a precisão do cálculo da postura do veículo robótico. Deste modo, com o robô iniciando o processo sem nenhum mapa ou conhecimento prévio do ambiente, a estimação do mapa e da postura do dirigível é composta dos seguintes passos:

1. Os sensores inerciais são utilizados em um INS para estimar o deslocamento do dirigível;

- 2. A câmera observa as balizas do ambiente;
- Se houver uma ou mais correspondências entre o mapa e a observação recém adquirida do ambiente, a postura do dirigível e o mapa são atualizados;
- As balizas observadas pela primeira vez são anexadas ao mapa interno, que deste modo é construído incrementalmente.

A correção da postura e do mapa só é possível graças à quantificação da correlação entre as variáveis sendo estimadas pelo sistema. Como todas estão correlacionadas, a correção de uma implica no ajuste de outra. No entanto, embora a observação constante de balizas aumente a precisão da navegação consideravelmente em comparação com um INS, a derivação do erro causado pelo sistema inercial imbutido no sistema de SLAM faz com que, muito lentamente, a incerteza continue aumentando.

Um dos aspectos mais interessantes da metodologia de SLAM é justamente a capacidade de correção destes erros integrados, causados pelo sistema inercial (conforme mostrado na figura 1.1). Se eventualmente o laço é fechado, ou seja, se uma baliza é observada novamente depois de muito tempo, o erro acumulado da estimativa da postura do dirigível pode ser observado. Neste momento, como a posição original da baliza está relacionada a todas as outras variáveis do sistema, tanto a postura do veículo quanto o mapa como um todo podem ser corrigidos, o que resulta em um salto de qualidade do mapa.

Deste modo, conforme o veículo continua transitando o ambiente e prossegue observando as mesmas balizas, o erro do mapeamento tende a um limite inferior, que é função da incerteza inicial do veículo quando a primeira baliza é observada (Dissanayake et al., 2000). Vale lembrar que a correção da postura do veículo através da observação das balizas está atrelada à precisão do mapa. Deste modo, neste ponto a atualização da postura do dirigível a cada iteração também atinge seu limite de precisão, já que a incerteza do mapeamento não pode mais ser reduzida.

1.2 O projeto AURORA

Veículos aéreos não tripulados possuem um grande espectro de aplicações, não apenas militares como também civis. Aeronaves autônomas ou tele-operadas podem ser utilizadas na inspeção de linhas de transmissão, gasodutos e oleodutos, monitoração ambiental, prospecção em áreas urbanas e rurais, vigilância em grandes áreas e análise de tráfego urbano e rodoviário.

Algumas destas atividades podem ser desempenhadas utilizando imagens adquiridas por satélites. No entanto, o seu uso civil não fornece a resolução necessária para observação mais precisa e o tempo de aquisição das imagens não permite agilidade de resposta. A solução é a utilização de um veículo aéreo, podendo este ser um avião, um helicóptero ou um dirigível.

Dirigíveis não tripulados podem pairar sobre a área de interesse, caracterizando uma das principais vantagens deste tipo de veículo. Além disso, possuem autonomia de vôo extensa e grande capacidade de carga em relação ao peso. Ainda, exigem menor esforço de sustentação e fornecem maior segurança de operação. Por estes motivos, são mais apropriados para missões de aquisição de dados em baixa altitude e velocidade.



Fig. 1.3: O dirigível AS800B, plataforma de desenvolvimento e validação experimental do projeto AURORA.

Neste contexto, o projeto AURORA (Autonomous, Unmanned, Remote mOnitoring Robotic Airship) foi iniciado pelo CenPRA (Centro de Pesquisas Renato Archer) em 1997, conforme descrito em (Elfes et al., 1998; Gomes and Ramos, 1998; Bueno et al., 2002; de Paiva et al., 2006), tendo como objetivos:

- o estabelecimento de tecnologia para a operação autônoma de veículos robóticos aéreos, utilizando um dirigível não tripulado como plataforma;
- contemplar aplicações em sensoriamento remoto, pesquisa e monitoração ambiental, inspeção de grandes estruturas e vigilância.

A plataforma atualmente utilizada pelo projeto AURORA é o dirigível AS800B, mostrado na figura 1.3. O AS800B possui 10,5 m de comprimento, 3 m de diâmetro e 34 m³ de volume, dispondo de uma capacidade de carga de 10 Kg. Ele é equipado com dois motores vetorizáveis e quatro superfícies de cauda, posicionadas a 45 graus do plano vertical e distribuídas uniformemente em torno do eixo principal. A infra-estrutura existente compreende:

- uma estação embarcada em PC/104 (arquitetura de hardware e software específica para sistemas embarcados) e Linux tempo real. A estação congrega como principais sensores: GPS, bússola, central inercial, sonda de vento para medição de velocidade e altitude barométrica e câmera DV (Digital Video), sendo que a última está conectada a um barramento no padrão IEEE 1394 (firewire) para maior velocidade na transmissão dos dados;
- uma estação em terra para operação e monitoramento do sistema do dirigível, além do armazenamento de dados fornecidos pelos sensores embarcados na aeronave;
- um sistema de comunicação entre ambas, utilizando radio-modems e links de vídeo sem fio de curto alcance.

Os resultados científico-tecnológicos já obtidos no AURORA, além da infraestrutura citada, compreendem também e principalmente metodologias de controle, que permitem ao dirigível de pequeno porte a execução autônoma de trajetórias definidas por pontos de passagem e perfis de altitude.

Outros importantes esforços de pesquisa com dirigíveis robóticos são descritos em (Zhang and Ostrowski, 1998; Kantor et al., 2001; Wimmer et al., 2002; Hygounenc et al., 2004) e, considerando missões de exploração planetária, em (Elfes et al., 2003). Em outra abordagem, pesquisas com dirigíveis para operação de longa duração na estratosfera (HALE – *High Altitude Long Endurance*) estão atualmente sendo desenvolvidas no Japão, na Coréia, na Europa e nos Estados Unidos, envolvendo tanto a indústria quanto o meio acadêmico.

1.3 Objetivos

Esta dissertação tem em seu cerne o objetivo de fornecer a um veículo robótico aéreo – sem perda de generalidade, um dirigível robótico – um sistema de navegação que lhe permita determinar seu posicionamento em um ambiente até então desconhecido. Ao mesmo tempo em que o sistema integra as informações sensoriais para localizar o dirigível, deve construir um mapa dos arredores, que pode tanto ser utilizado no desempenhar autônomo de uma dada missão quanto ser a meta de uma missão de aquisição de dados topográficos; ou ainda um conjunto de dados para o planejamento autônomo de uma ação futura.

A metodologia desenvolvida deve ser robusta e confiável, o que significa que, mesmo em situações precárias, o sistema até certo ponto deve continuar coerente e ter a capacidade de retomar atividade normal se possível. No sistema de navegação, os sensores não conseguem medir o deslocamento do robô com exatidão, o que implica em um erro no mapa. Por outro lado, como a postura do dirigível é originária do mapa, sua localização também é incerta. Deste modo, a robustez passa pela correta quantificação da incerteza. Assim, a estimativa da postura do dirigível e do mapa deve levar em conta

a correta modelagem das incertezas envolvidas. Portanto, se a estimativa da localização do dirigível e do mapa se tornarem imprecisas, esta incerteza deve ser modelada corretamente. Do mesmo modo, se a estimativa se aproximar novamente da situação real, a imprecisão deve cair de acordo, permitindo que o sistema continue funcionando normalmente.

A abordagem aqui utilizada para alcançar estes objetivos corresponde a uma solução para a problemática de SLAM. Como sensores, são utilizados uma unidade de medição inercial (IMU - *Inertial Measurement Unit*) e um sensor visual, compreendido por uma câmera e os algoritmos de processamento de imagens que resultam na análise dos marcos presentes no ambiente. Deste modo, para atingir estas metas, os objetivos desta dissertação são:

- Analisar as diversas técnicas para localização do dirigível, selecionando a mais conveniente, dados: i) o conjunto de sensores disponível no dirigível; ii) e os resultados esperados;
- Desenvolver uma metodologia para a integração das informações proprioceptivas da IMU (que incorporam um modelo de INS) e exteroceptivas do sensor visual. Desta integração, resultam um mapa do ambiente e a postura do dirigível neste mapa;
- Modelar o sistema sensorial do dirigível relevante para o projeto, compreendendo a IMU e o sensor visual;
- Elaborar um algoritmo de correspondência entre os marcos do mapa e os marcos detectados pelo sensor visual. Este algoritmo deve ser também capaz de perceber marcos não observados a muito tempo, permitindo que o ciclo de estimação seja fechado e o mapa como um todo, corrigido;
- Implementar os algoritmos elaborados de maneira computacionalmente eficiente e versátil, permitindo que eventuais alterações sejam realizadas sem a necessidade de modificações no restante do sistema;
- Implementar um ambiente de simulação para a realização de testes da metodologia em situações próximas das encontradas pelo dirigível real. O ambiente de simulação deve simular não apenas os sensores do dirigível, mas também a interface de comunicação, permitindo a reutilização da implementação da metodologia de SLAM;
- Realizar vôos simulados e investigar os resultados, validando a metodologia;

1.4 Contribuições

Esta dissertação faz parte de um projeto maior, que visa o desenvolvimento de tecnologia para a operação autônoma de um dirigível robótico. Dentro deste escopo, um sistema sensorial que permita a localização precisa do dirigível é de grande valia para a continuidade do projeto.

Ao mesmo tempo, o problema de SLAM tem recebido grande interesse da comunidade científica e tem sido evidenciado como a melhor alternativa para sistemas de navegação na presença de sensores exteroceptivos. A metodologia, além de estimar o mapa e localização, realiza a fusão entre as informações dos sensores, minimizando a incerteza. Este fator evita a necessidade de utilização de sensores de grande precisão, que tornariam o projeto mais dispendioso. Deste modo, as principais contribuições deste trabalho são:

- criação e integração de um ambiente de simulação para o dirigível AS800B, incluindo um visualizador tridimensional;
- estudo, implementação e teste de opções para gravação sincronizada de seqüências de imagens (sem perdas) e dados sensoriais embarcados (vide Apêndice A);
- desenvolvimento de uma metodologia de SLAM em seis graus de liberdade que estima eficientemente o mapa e a localização do dirigível;
- o sistema desenvolvido presume a utilização de sensores de baixo custo, o que aumenta a viabilidade de utilização da metodologia em sistemas comerciais;
- independência do GPS, importante já que a disponibilidade do sinal é incerta, especialmente em alguns ambientes em que o dirigível deve operar;
- organização e versatilidade do sistema fazem da implementação um arcabouço de testes para algoritmos de Localização e Mapeamento Simultâneos;
- implementação pronta para a utilização no dirigível AS800B real, o que aumentará a variedade de pesquisa desenvolvida sobre a plataforma.

Deste modo, o trabalho aqui apresentado representa um importante primeiro passo para a implantação de uma solução de Localização e Mapeamento Simultâneos para o dirigível real. No entanto, a fim de manter o escopo compatível com um programa de mestrado, não será abordado o sistema de visão computacional responsável pelo processamento e análise das imagens da câmera, que levará à detecção e ao posicionamento das balizas presentes no ambiente.

1.5 Estrutura da dissertação

Após este capítulo introdutório, o restante desta dissertação está organizado como segue:

- Capítulo 2 formula a metodologia de SLAM em seis graus de liberdade utilizando um filtro de Kalman estendido para integrar os dados sensoriais. No modelo do processo, são empregadas as informações provenientes da IMU e na atualização e mapeamento, as informações do sensor visual. Os modelos dos dois sensores são desenvolvidos, além do algoritmo de correspondência de marcos visuais. Otimizações de desempenho são mencionadas.
- **Capítulo 3** Descreve a estrutura de software do ambiente de simulação e da metodologia de Localização e mapeamento. Destaca como a organização do sistema faz da implementação um arcabouço de teste dos algoritmos que compõem a metodologia.
- **Capítulo 4** Detalha a configuração dos sensores durante a simulação e o trajeto a ser percorrido. Exibe os resultados obtidos com a estratégia de SLAM no vôo simulado e, posteriormente, analisa os dados obtidos sob a ótica da observabilidade do sistema.
- **Capítulo 5** Traça as conclusões finais sobre a dissertação e apresenta sugestões para o posterior desenvolvimento do trabalho realizado.

Referências Bibliográficas

- **Apêndice A** Descreve os estudos e a implementação de uma biblioteca para a gravação de seqüências de vídeo juntamente a outros dados sensoriais e informações de sincronia, para posterior análise dos vôos do dirigível AS800B.
- Apêndice B Breve descrição de quaternions e sua utilização para representação de rotações.

Capítulo 2

Auto-localização e mapeamento simultâneos

O problema em questão é evidenciado ao considerar-se a situação em que um veículo robótico aéreo é posicionado em uma localização desconhecida, sem nenhuma informação prévia sobre seus arredores. O veículo robótico é equipado com uma IMU, que fornece velocidades angulares e acelerações lineares no referencial do corpo do veículo, e uma câmera, que pode ser utilizada para fazer medições relativas entre características no ambiente (suas posições absolutas são desconhecidas) e a própria câmera.

Em uma metodologia de navegação inercial convencional, as leituras da IMU são fornecidas diretamente ao sistema de navegação inercial (INS – *inertial navigation system*) da aeronave para a determinação de sua postura, que compreende tanto sua posição quanto sua atitude (orientação). No entanto, imperfeições mecânicas e de medição, além de erros de discretização, fazem com que as leituras da IMU contenham um erro de média diferente de zero. Este erro é integrado no cálculo do deslocamento do veículo e, quando não compensado, faz com que a disparidade entre a postura real e a estimada aumenta indefinidamente ao longo do tempo, em um processo de deriva.

No presente trabalho, no entanto, as equações do INS são inseridas no modelo cinemático do veículo robótico aéreo e, conforme este se move pelo ambiente, tal modelo é utilizado para computar o deslocamento relativo do veículo em um referencial geodésico. Considera-se também, por outro lado, que o ambiente sobrevoado pela aeronave apresenta aspectos visuais marcantes, caracterizados como balizas. Tais características são observadas pela câmera embarcada, e a posição relativa entre sensor e baliza, obtida da análise das imagens, é fornecida ao sistema de localização e mapeamento do veículo. A repetida detecção das mesmas balizas faz perceptível a postura do robô com relação às balizas. Implicitamente, é também observado o erro da IMU, que torna-se passível de correção. Deste modo, a cada nova baliza reconhecida no ambiente, não apenas o mapeamento do ambiente é realizado como também, simultaneamente, é efetuada a correção da posição das balizas e da postura do próprio veículo robótico, eliminando a derivação causada pela simples utilização dos sinais da

IMU.

Desta abordagem de fusão sensorial entre a IMU, um sensor proprioceptivo, e o sistema de visão, exteroceptivo, resulta o sistema de auto-localização e mapeamento simultâneos do veículo robótico aéreo.

Neste capítulo, a formulação matemática do problema é desenvolvida considerando, sem perda de generalidade, um dirigível como plataforma robótica. Inicialmente, é definido o estado do sistema e a representação da incerteza a ele associada. Posteriormente, são elaborados o modelo cinemático do dirigível – que relaciona o estado anterior da aeronave e os sinais da IMU para obter a postura atual do dirigível – e o modelo de observação do sistema de visão – que associa a observação de uma baliza pela câmera à sua posição georreferenciada (na qual o mapa é composto). A seguir, a abordagem para fusão sensorial e localização autônoma baseada do filtro de Kalman estendido é apresentada, descrevendo como o modelo cinemático é utilizado para prever o estado do sistema e como o modelo de observação é utilizado para atualizar tal predição e incrementar o mapa do ambiente a cada baliza descoberta durante o vôo.

2.1 Sistemas de Coordenadas

Nesta seção, são definidos os sistemas de coordenadas relevantes para a compreensão do restante do capítulo, já que cada sensor produz leituras em seu próprio referencial e um único sistema de coordenadas é utilizado para concatenar todas as informações.

A figura 2.1 ilustra o envelope do dirigível e a gôndola sob o envelope, onde são montados os sensores embarcados. Na figura, podem ser observados os referenciais de maior importância. O referencial \mathbf{F} mostra o centro de flutuação do dirigível, que coincide com o centro de massa do envelope. Dentro da gôndola do dirigível, é possível notar \mathbf{B} , que indica o referencial da IMU. Idealmente, o eixo $x_{\mathbf{B}}$ indica a direita do dirigível, enquanto $y_{\mathbf{B}}$ indica a frente e $z_{\mathbf{B}}$ a parte superior do corpo do dirigível, formando uma base ortogonal. Como todas as informações inerciais são references a \mathbf{B} , este é considerado o referencial do próprio dirigível.

Por outro lado, **C** mostra o sistema de coordenadas da câmera, também ortogonal, no qual são representadas as informações obtidas das balizas presentes no ambiente de atuação do veículo robótico aéreo. Considera-se que a origem de **C** está no centro focal da câmera, sendo que o eixo $x_{\rm C}$ aponta para a direita do plano da imagem e $y_{\rm C}$ para a parte superior do mesmo plano. Já $z_{\rm C}$ coincide com o eixo focal, mas é direcionado no sentido da luz incidente na câmera, de modo que coordenadas em frente à câmera possuem componente $z_{\rm C}$ negativo.

Tanto \mathbf{F} quanto $\mathbf{B} \in \mathbf{C}$ deslocam-se conforme o dirigível se move em relação ao ambiente. Para concatenar as informações obtidas nos referenciais $\mathbf{B} \in \mathbf{C}$, e gerar uma representação estática do am-



Fig. 2.1: Sistemas de coordenadas relevantes.

biente, é utilizado o sistema de coordenadas **W**, que é georreferenciado. Tanto a postura do dirigível quanto a posição das balizas que compõem o mapa são traduzidas para este referencial ao longo do processo de auto-localização e mapeamento.

A auto-localização do veículo corresponde a obter, a qualquer instante, a melhor estimativa para a sua postura; ou seja, para a relação entre **B** e **W**. No entanto, para o funcionamento do sistema aqui desenvolvido, é importante conhecer a transformação que leva de **B** a **C**, tal que da variação na postura do veículo obtenha-se a variação na postura da câmera. No entanto, a estimação desta transformação

foge do escopo deste trabalho, de modo que a relação entre **B** a **C** é tida como conhecida. Além disso, a fim de simplificar o equacionamento do problema, sem perda de generalidade, no estudo aqui apresentado considera-se a origem do sistema de coordenadas **B** coincidente com a de **C**, tal que ambos estão separados apenas por uma rotação também conhecida. A figura 2.2 apresenta esta situação.



Fig. 2.2: Considera-se que apenas uma rotação diferencia o referencial da IMU (**B**) e o referencial da câmera (**C**).

A notação aqui apresentada para referenciar os diferentes sistemas de coordenadas será utilizada ao longo do capítulo com o mesmo significado.

2.2 O Estado do Sistema e a Representação de Incerteza

O vetor do estado do sistema, \mathbf{x} , é definido como o estado do veículo, \mathbf{x}_{ν} , acrescido do vetor de estado \mathbf{x}_m do mapa, tal que

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_{\nu} \\ \mathbf{x}_{m} \end{bmatrix}.$$
(2.1)

Na equação acima, o vetor de estado do veículo é definido como

$$\mathbf{x}_{\nu} = \begin{bmatrix} \mathbf{p} \\ \mathbf{v} \\ \mathbf{q} \end{bmatrix}$$
(2.2)

em que \mathbf{p} e \mathbf{v} são, respectivamente, a posição e a velocidade do dirigível no referencial de terra \mathbf{W} . A orientação do robô aéreo, também georreferenciada, é representada pelo quaternion \mathbf{q} (mais detalhes no apêndice B). Embora seja uma representação não-mínima de orientação, um quaternion é utilizado a fim de evitar problemas de linearização e singularidades de outras representações – com ângulos de Euler, por exemplo, se uma rotação prévia leva o vetor transformado a um dos eixos de rotação, rotações subseqüentes naquele eixo não surtirão efeito e não serão representadas.

Já o vetor de estado do mapa consiste de um conjunto de características detectadas no ambiente de atuação do dirigível, assim como o estado do robô, também definidas no referencial de terra W. Como o mapa é construído incrementalmente, o tamanho do vetor de estado do mapa varia, aumentando conforme o ambiente é mapeado. Para um mapa com *N* balizas, define-se:

$$\mathbf{x}_{m} = \begin{bmatrix} \mathbf{x}_{m1} \\ \mathbf{x}_{m2} \\ \vdots \\ \mathbf{x}_{mN} \end{bmatrix}.$$
 (2.3)

onde $\mathbf{x}_{mi} = [x_{mi}, y_{mi}, z_{mi}]$ representa a localização georreferenciada de uma característica visual (baliza) em coordenadas cartesianas.

No entanto, o estado real do dirigível e do mapa não são conhecidos e, na metodologia aqui proposta, um filtro de Kalman estendido (EKF) é utilizado como um estimador de estado para o veículo e para o mapa ao mesmo tempo, propagando as incertezas de primeira ordem em uma abordagem conhecida como *mapeamento estocástico* (Smith et al., 1987).

Deste modo, define-se a estimativa do estado do sistema x por

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{x}}_v \\ \hat{\mathbf{x}}_m \end{bmatrix}.$$
(2.4)

-

O vetor de estado estimado $\hat{\mathbf{x}}$, por sua vez, é acompanhado por uma matriz de covariância \mathbf{P} , que representa a incerteza presente na estimativa e é definida como

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{v,v} & \mathbf{P}_{v,m1} & \mathbf{P}_{v,m2} & \dots & \mathbf{P}_{v,mN} \\ \mathbf{P}_{m1,v} & \mathbf{P}_{m1,m1} & \mathbf{P}_{m1,m2} & \dots & \mathbf{P}_{m1,mN} \\ \mathbf{P}_{m2,v} & \mathbf{P}_{m2,m1} & \mathbf{P}_{m2,m2} & \dots & \mathbf{P}_{m2,mN} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{mN,v} & \mathbf{P}_{mN,m1} & \mathbf{P}_{mN,m2} & \dots & \mathbf{P}_{mN,mN} \end{bmatrix} .$$
(2.5)

No processo de estimação, o modelo cinemático é utilizado para predizer a postura do dirigível e estimar o aumento da incerteza da postura, decorrente dos erros da IMU. O modelo de observação, por sua vez, é utilizado para atualizar a predição e reduzir a incerteza da estimação, tendo como base as informações exteroceptivas do sistema de visão.



Fig. 2.3: O modelo cinemático do dirigível utiliza $\mathbf{u}(k) \in \mathbf{w}(k)$ para levar de $\mathbf{x}_v(k-1)$ a $\mathbf{x}_v(k)$

2.3 Modelo cinemático não-linear

O modelo cinemático do robô aéreo consiste das equações de um sistema de navegação inercial (INS). Deste modo, o deslocamento do dirigível em um determinado intervalo de tempo pode ser estimado a partir dos sinais fornecidos pela IMU, $\mathbf{u}(k)$, e do ruído do processo, $\mathbf{w}(k)$. Assim, como mostra a figura 2.3, o vetor do estado do robô $\mathbf{x}_v(k)$ pode ser calculado a partir do estado anterior $\mathbf{x}_v(k-1)$:

$$\mathbf{x}_{\nu}(k) = f(\mathbf{x}_{\nu}(k-1), \mathbf{u}(k), \mathbf{w}(k)).$$
(2.6)

Contudo, antes de desenvolver o modelo cinemático do dirigível, é necessário analisar os sinais fornecidos pela IMU em um instante *k*:

$$\mathbf{u}(k) = [\alpha(k), \ \omega(k)], \tag{2.7}$$

onde

$$\alpha = \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \qquad (2.8)$$
$$\omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \qquad (2.9)$$

sendo que α representa a aceleração linear do dirigível e ω representa a velocidade angular do dirigível, ambos no referencial **B** do próprio dirigível, como mostra a figura 2.4.

Deste modo, a equação 2.6 pode ser reescrita em um modelo de navegação inercial, conforme



Fig. 2.4: Os eixos do dirigível e a representação da direção dos sinais da IMU.

abaixo:

$$\mathbf{x}_{\nu}(k) = f(\mathbf{x}_{\nu}(k-1), \mathbf{u}(k), \mathbf{w}(k))$$

=
$$\begin{bmatrix} \mathbf{p}(k-1) \\ \mathbf{v}(k-1) \\ \mathbf{q}(k-1) \end{bmatrix} + \begin{bmatrix} \Delta \mathbf{p}(k) \\ \Delta \mathbf{v}(k) \\ \Delta \mathbf{q}(k) \end{bmatrix} + \begin{bmatrix} \mathbf{w}_{\mathbf{p}}(k) \\ \mathbf{w}_{\mathbf{v}}(k) \\ \mathbf{w}_{\mathbf{q}}(k) \end{bmatrix}, \qquad (2.10)$$

onde o ruído $\mathbf{w}(k)$ é assumido gaussiano, não-correlato e de média zero.

Em diversos trabalhos (Bailey, 2002; Guivante, 2002) o modelo cinemático é elaborado em três graus de liberdade, assumindo movimento do veículo no plano. Já em (Kim, 2004), as equações do sistema inercial foram desenvolvidas em seis graus de liberdade, mas desprezando a aceleração no cálculo da posição. Neste trabalho, por sua vez, a variação da postura do dirigível, $\Delta \mathbf{x}_v(k)$, pode ser definida no referencial **W** através da integração dos sinais da IMU, de acordo com as equações de um

INS, tal que

$$\begin{bmatrix} \Delta \mathbf{p}(k) \\ \Delta \mathbf{v}(k) \\ \Delta \mathbf{q}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{v}(k-1)\Delta t + \mathbf{C}_{B}^{W}(k-1)(\alpha(k)+\mathbf{g})\frac{\Delta t^{2}}{2} \\ \mathbf{C}_{B}^{W}(k-1)(\alpha(k)+\mathbf{g})\Delta t \\ (\frac{1}{2}\mathbf{D}(k-1)\omega(k)+\varepsilon(k-1)\mathbf{q}(k-1))\Delta t \end{bmatrix}, \quad (2.11)$$

$$\mathbf{D}(k) = \begin{bmatrix} q_{i}(k) & q_{j}(k) & q_{k}(k) \\ -q_{r}(k) & q_{k}(k) & -q_{j}(k) \\ -q_{k}(k) & -q_{r}(k) & q_{i}(k) \\ q_{j}(k) & -q_{i}(k) & -q_{r}(k) \end{bmatrix}, \quad (2.12)$$

$$\varepsilon(k) = 1 - \|\mathbf{q}(k)\|^{2}, \quad (2.13)$$

sendo que **g** corresponde à aceleração gravitacional na região de operação do dirigível. As leituras de aceleração linear provenientes da IMU correspondem à aceleração provocada pelos movimentos de translação do dirigível acrescida da aceleração referente à força gravitacional, que deve ser compensada (por esta razão é necessária uma estimativa precisa de **g**). Na equação,
$$\varepsilon$$
 representa um ganho de correção da integração da orientação, importante quando a norma do quaternion é diferente de um (deixando, portanto, de refletir uma rotação). Além disso, $\mathbf{C}_B^W(k-1)$ é a matriz de cossenos diretores obtida de $\mathbf{q}(k-1)$. Neste caso, a utilização da matriz representa uma alternativa computacionalmente mais eficiente para o cálculo que leva da orientação do referencial do dirigível *B* à orientação do

referencial de terra *W*, e é especificada conforme a equação abaixo (com $\mathbf{q}(k-1) = [q_r, q_i, q_j, q_k]$):

$$\mathbf{C}_{B}^{W}(k-1) = \begin{bmatrix} q_{r}^{2} + q_{i}^{2} - q_{j}^{2} - q_{k}^{2} & 2(q_{i}q_{j} - q_{r}q_{k}) & 2(q_{r}q_{j} + q_{i}q_{k}) \\ 2(q_{i}q_{j} + q_{r}q_{k}) & q_{r}^{2} - q_{i}^{2} + q_{j}^{2} - q_{k}^{2} & 2(q_{j}q_{k} - q_{r}q_{i}) \\ 2(q_{i}q_{k} - q_{r}q_{j}) & 2(q_{j}q_{k} + q_{r}q_{i}) & q_{r}^{2} - q_{i}^{2} - q_{j}^{2} + q_{j}^{2} - q_{k}^{2} \end{bmatrix}$$
(2.14)

É importante lembrar que, como o cálculo da posição é sempre relativo à posição anterior, os erros se propagam de uma iteração para outra, tendendo a aumentar sem limites especialmente se o ruído for polarizado.

A seguir, são calculados os Jacobianos do modelo cinemático em relação ao estado do dirigível e aos sinais da IMU, cuja relevância para o processo de estimação será detalhada na seção 2.6 deste capítulo.

de

2.3.1 Jacobiano em relação ao estado do dirigível

O Jacobiano do modelo cinemático não-linear do dirigível, representado na equação 2.10, em função do estado do dirigível, é importante para a estimação do estado do sistema e pode ser calculado como

$$\nabla f_{x_{\nu}} = \begin{bmatrix} \frac{\partial \mathbf{p}(k)}{\partial \mathbf{p}(k-1)} & \frac{\partial \mathbf{p}(k)}{\partial \mathbf{v}(k-1)} & \frac{\partial \mathbf{p}(k)}{\partial \mathbf{q}(k-1)} \\ \frac{\partial \mathbf{v}(k)}{\partial \mathbf{p}(k-1)} & \frac{\partial \mathbf{v}(k)}{\partial \mathbf{v}(k-1)} & \frac{\partial \mathbf{v}(k)}{\partial \mathbf{q}(k-1)} \\ \frac{\partial \mathbf{q}(k)}{\partial \mathbf{p}(k-1)} & \frac{\partial \mathbf{q}(k)}{\partial \mathbf{v}(k-1)} & \frac{\partial \mathbf{q}(k)}{\partial \mathbf{q}(k-1)} \end{bmatrix},$$
(2.15)

Após algumas substituições, à luz das equações 2.10 e 2.11, a equação acima resulta em

$$\nabla f_{x_{\nu}} = \begin{bmatrix} \mathbf{I} \quad \Delta t \, \mathbf{I} & \frac{\Delta t^2}{2} \frac{\partial \mathbf{C}_B^W(k-1) \,\alpha(k)}{\partial \mathbf{q}(k-1)} \\ 0 \quad \mathbf{I} & \Delta t \frac{\partial \mathbf{C}_B^W(k-1) \,\alpha(k)}{\partial \mathbf{q}(k-1)} \\ 0 & 0 \quad \mathbf{I} + \Delta t \left(\frac{1}{2} \frac{\partial (\mathbf{D}(k-1) \,\omega(k))}{\partial \mathbf{q}(k-1)} + \frac{\partial (\varepsilon(k-1) \,\mathbf{q}(k-1))}{\partial \mathbf{q}(k-1)} \right) \end{bmatrix},$$
(2.16)

A matriz obtida da derivada

$$\frac{\partial \mathbf{C}_B^W(k-1)\,\boldsymbol{\alpha}(k)}{\partial \mathbf{q}(k-1)},$$

considerando $\mathbf{q}(k-1) = [q_r, q_i, q_j, q_k]$, é por sua vez definida como

$$\frac{\partial (\mathbf{C}_{B}^{W}(k-1)\,\alpha(k))}{\partial \mathbf{q}(k-1)} = 2 \begin{bmatrix} uq_{r} - vq_{k} + wq_{j} & uq_{k} + vq_{r} - wq_{i} & -uq_{j} + vq_{i} + wq_{r} \\ uq_{i} + vq_{j} + wq_{k} & uq_{j} - vq_{i} - wq_{r} & uq_{k} + vq_{r} - wq_{i} \\ -uq_{j} + vq_{i} + wq_{r} & uq_{i} + vq_{j} + wq_{k} & -uq_{r} + vq_{k} - wq_{j} \\ -uq_{k} - vq_{r} + wq_{i} & uq_{r} - vq_{k} + wq_{j} & uq_{i} + vq_{j} + wq_{k} \end{bmatrix}^{T}.$$
 (2.17)

Já a derivada

$$\frac{\partial(\mathbf{D}(k-1)\boldsymbol{\omega}(k))}{\partial \mathbf{q}(k-1)}$$

é obtida da definição de $\mathbf{D}(k-1)$, presente na equação 2.12, e pode ser computada pela seguinte matriz:

$$\frac{\partial(\mathbf{D}(k-1)\,\boldsymbol{\omega}(k))}{\partial\mathbf{q}(k-1)} = \begin{bmatrix} 0 & p & q & r \\ -p & 0 & -r & q \\ -q & r & 0 & -p \\ -r & -q & p & 0 \end{bmatrix}.$$
 (2.18)

Finalmente, para que o Jacobiano ∇f_{x_v} especificado na função 2.15 seja definido completamente,
é necessário determinar a matriz

$$\frac{\partial \boldsymbol{\varepsilon}(k-1) \, \mathbf{q}(k-1)}{\partial \mathbf{q}(k-1)}$$

que, observando a equação 2.13, é calculada por:

$$\frac{\partial \left(\boldsymbol{\varepsilon}(k-1)\,\mathbf{q}(k-1)\right)}{\partial \,\mathbf{q}(k-1)} = -2\,\mathbf{q}(k-1)\,\mathbf{q}^{T}(k-1) + \boldsymbol{\varepsilon}(k-1)\,\mathbf{I}$$
(2.19)

2.3.2 Jacobiano em relação aos sinais da IMU

O Jacobiano de $f(\mathbf{x}_{v}(k), \mathbf{u}(k), \mathbf{w}(k))$ em relação ao vetor $\mathbf{u}(k)$ dos sinais da IMU é dado por

$$\nabla f_{u} = \begin{bmatrix} \frac{\partial \mathbf{p}(k)}{\partial \alpha(k)} & \frac{\partial \mathbf{p}(k)}{\partial \omega(k)} \\ \frac{\partial \mathbf{v}(k)}{\partial \alpha(k)} & \frac{\partial \mathbf{v}(k)}{\partial \omega(k)} \\ \frac{\partial \mathbf{q}(k)}{\partial \alpha(k)} & \frac{\partial \mathbf{q}(k)}{\partial \omega(k)} \end{bmatrix}.$$
(2.20)

Contudo, observando a equação 2.10, percebe-se que vários elementos da matriz acima são nulos, e aqueles que não são podem ser definidos em função de valores já determinados anteriormente. Deste modo a equação 2.20 pode ser reescrita conforme mostrado abaixo:

$$\nabla f_{u} = \begin{bmatrix} \frac{\Delta t^{2}}{2} \frac{\partial (\mathbf{C}_{B}^{W}(k-1) \alpha(k))}{\partial \alpha(k)} & \mathbf{0} \\ \Delta t \frac{\partial (\mathbf{C}_{B}^{W}(k-1) \alpha(k))}{\partial \alpha(k)} & \mathbf{0} \\ \mathbf{0} & \frac{\Delta t}{2} \frac{\partial (\mathbf{D}(k-1) \omega(k))}{\partial \omega(k)} \end{bmatrix}$$
(2.21)

$$= \begin{bmatrix} \frac{\Delta t^2}{2} \mathbf{C}_B^W(k-1) & \mathbf{0} \\ \Delta t \, \mathbf{C}_B^W(k-1) & \mathbf{0} \\ \mathbf{0} & \frac{\Delta t}{2} \, \mathbf{D}(k-1) \end{bmatrix}$$
(2.22)

2.4 Relação entre o mapa e a observação

Neste trabalho, a observação do sistema se dá através da percepção de características presentes no ambiente. Mais especificamente, é considerado que o sensor visual presente na aeronave observa balizas dispostas nos arredores e retorna ao sistema de SLAM a posição relativa entre a câmera e as balizas em coordenadas cartesianas, como mostrado na figura 2.5. Em outros trabalhos, assume-se que o sistema de visão retorna a direção da baliza (dois ângulos) e a distância, o que se enquadra em arcabouço matemático bastante explorado (Bailey, 2002; Kim, 2004; Langelaan and Rock, 2005). Ambos os espaços de observação são equivalentes através de uma mudança de sistema de coordenadas.



Fig. 2.5: O dirigível observa uma baliza.

Assim sendo, é preciso investigar como as observações das balizas, coordenadas tridimensionais feitas no referencial da câmera **C**, estão relacionadas às suas posições no referencial de terra **W**, no qual o mapa é construído. Em outros trabalhos (Kim, 2004), a câmera é considerada posicionada a uma translação do sistema de coordenadas da central inercial, mas assumida apontando sempre diretamente para baixo, o que simplifica consideravelmente as equações. Aqui, considera-se que o sistema de coordenadas da câmera possui a mesma origem do referencial **B** da central inercial; no entanto, a rotação da câmera é arbitrária, o que torna o equacionamento mais abrangente. Deste modo, é possível especificar cada baliza \mathbf{x}_{mi} do mapa em função do estado do dirigível e da observação da característica no instante *k* (no instante *k* em que o mapa possui *N* características, $1 \le i \le N$):

$$\mathbf{x}_{mi} = g(\mathbf{x}_{v}(k), \mathbf{z}_{i}(k))$$

= $\mathbf{p}(k) + R_{B}^{W}(R_{C}^{B}(\mathbf{z}_{i}(k))),$ (2.23)

onde $R_C^B(\cdot)$ é uma função de transformação que leva do referencial da câmera ao referencial do dirigível através do quaternion $\mathbf{q}_C^B \in R_B^W(\cdot)$ faz o mesmo levando do referencial do dirigível para o do mundo (utilizando o quaternion presente no estado do dirigível no instante k, $\mathbf{q}(k)$). Ou seja:

$$\begin{bmatrix} 0 \\ \mathbf{x}_{mi} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{p}(k) \end{bmatrix} + \mathbf{q}(k) \otimes \mathbf{q}_C^B \otimes \begin{bmatrix} 0 \\ \mathbf{z}_i(k) \end{bmatrix} \otimes \overline{\mathbf{q}}_C^B \otimes \overline{\mathbf{q}}(k), \qquad (2.24)$$

tal que \otimes representa a multiplicação entre quaternions e $\overline{\mathbf{q}}(k)$ e $\overline{\mathbf{q}}_C^B$ representam, respectivamente, os conjugados do quaternion $\mathbf{q}(k)$ e do quaternion \mathbf{q}_C^B

2.4.1 Jacobiano em relação à observação

O cálculo do Jacobiano da equação 2.23 em relação a uma observação z_i é importante para determinar como a incerteza na observação de uma baliza se reflete na obtenção de sua posição em coordenadas georreferenciadas (conforme será detalhado posteriormente na seção 2.6). Este cálculo pode ser simplificado se as rotações forem representadas por uma matriz de cossenos diretores, o que pode ser obtido da equação 2.14. Deste modo, a equação 2.23 pode ser reescrita como

$$\mathbf{x}_{mi} = g(\mathbf{x}_{\nu}(k), \mathbf{z}_{i}(k))$$

= $\mathbf{p}(k) + \mathbf{C}_{B}^{W} \mathbf{C}_{C}^{B} \mathbf{z}_{i}(k),$ (2.25)

o que evidencia a forma do Jacobiano em relação à observação, que é dado por

$$\nabla g_{z_i} = \mathbf{C}_B^W \, \mathbf{C}_C^B. \tag{2.26}$$

2.4.2 Jacobiano em relação ao estado do veículo

Com \mathbf{x}_{mi} já definido em função de matrizes de cossenos diretores, o Jacobiano ∇g_{x_v} pode ser descrito, após algumas substituições, como:

$$\nabla g_{x_{\nu}} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \frac{\partial \left(\mathbf{C}_{B}^{W} \mathbf{z}_{i}^{B}(k)\right)}{\partial \mathbf{q}(k)} \end{bmatrix}, \qquad (2.27)$$

sendo $\mathbf{z}_i^B(k)$ a observação no referencial do dirigível, ao invés do referencial da câmera, obtida da equação

$$\mathbf{z}_i^B(k) = \mathbf{C}_C^B \mathbf{z}_i \tag{2.28}$$

Deste modo, o terceiro termo do Jacobiano torna-se similar ao exposto na equação 2.17, com

$\mathbf{z}_{i}^{B} = [z_{x}^{B} z_{y}^{B} z_{z}^{B}]$ substituindo α :

$$\frac{\partial \mathbf{C}_{B}^{W} \mathbf{z}_{i}^{B}}{\partial \mathbf{q}(k)} = 2 \begin{bmatrix} z_{x}^{B} q_{r} - z_{y}^{B} q_{k} + z_{z}^{B} q_{j} & z_{x}^{B} q_{k} + z_{y}^{B} q_{r} - z_{z}^{B} q_{i} & -z_{x}^{B} q_{j} + z_{y}^{B} q_{i} + z_{z}^{B} q_{r} \\ z_{x}^{B} q_{i} + z_{y}^{B} q_{j} + z_{z}^{B} q_{k} & z_{x}^{B} q_{j} - z_{y}^{B} q_{i} - z_{z}^{B} q_{r} & z_{x}^{B} q_{k} + z_{y}^{B} q_{r} - z_{z}^{B} q_{i} \\ -z_{x}^{B} q_{j} + z_{y}^{B} q_{i} + z_{z}^{B} q_{r} & z_{x}^{B} q_{i} + z_{y}^{B} q_{j} + z_{z}^{B} q_{k} & -z_{x}^{B} q_{r} + z_{y}^{B} q_{k} - z_{z}^{B} q_{j} \\ z_{z}^{B} q_{i} - z_{x}^{B} q_{k} - z_{y}^{B} q_{r} & z_{x}^{B} q_{r} - z_{y}^{B} q_{k} + z_{z}^{B} q_{j} & z_{x}^{B} q_{i} + z_{y}^{B} q_{j} + z_{z}^{B} q_{k} \end{bmatrix}^{T}$$
(2.29)

2.5 O modelo não-linear de observação

De posse da equação que relaciona as características do mapa às observações, é possível desenvolver o modelo não-linear de predição das observações realizadas pelo sistema de visão:

$$\mathbf{z}_i(k) = h(\mathbf{x}(k)) + \mathbf{v}_i(k),$$

sendo que $\mathbf{v}_i(k)$ representa o ruído de observação (considerado aditivo), gaussiano, não-correlato, de média zero e covariância $\mathbf{R}(k)$:

$$\mathbf{R}(k) = \begin{bmatrix} \sigma_x^2 & 0 & 0\\ 0 & \sigma_y^2 & 0\\ 0 & 0 & \sigma_z^2 \end{bmatrix}$$
(2.30)

Tomando a inversa da equação 2.23, as observações em qualquer instante podem ser preditas a partir do vetor de estado do veículo e do mapa, desde que as observações correspondam a características previamente observadas, com o seguinte modelo de observação:

$$\mathbf{z}_{i}(k) = h(\mathbf{x}_{v}(k), \mathbf{x}_{mi}) + \mathbf{v}_{i}(k)$$

$$= g^{-1}(\mathbf{x}_{v}(k), \mathbf{x}_{mi}) + \mathbf{v}_{i}(k)$$

$$= R_{B}^{C}(R_{W}^{B}(\mathbf{x}_{mi} - \mathbf{p}(k))) + \mathbf{v}_{i}(k), \qquad (2.31)$$

no qual as funções de troca de referencial de orientação seguem a mesma notação já especificada.

Resta agora desenvolver o Jacobiano da equação acima em relação ao vetor de estado do sistema $\mathbf{x} = [\mathbf{x}_v, \mathbf{x}_m].$

T

2.5.1 Jacobiano em relação ao estado

O Jacobiano ∇h_x é obtido da equação

$$\nabla h_{x} = \begin{bmatrix} \frac{\partial R_{B}^{C}(R_{W}^{B}(\mathbf{x}_{mi}-\mathbf{p}(k)))}{\partial \mathbf{p}(k)} & \mathbf{0} & \frac{\partial R_{B}^{C}(R_{W}^{B}(\mathbf{x}_{mi}-\mathbf{p}(k)))}{\partial \mathbf{q}(k)} & \cdots & \frac{\partial R_{B}^{C}(R_{W}^{B}(\mathbf{x}_{mi}-\mathbf{p}(k)))}{\partial \mathbf{x}_{mi}} & \cdots \end{bmatrix}, \quad (2.32)$$

sendo que a derivada em relação às características do mapa não correspondentes à observação são iguais a zero.

A fim de determinar completamente a equação acima, será inicialmente detalhada a derivada em relação a $\mathbf{p}(k)$, presente na primeira coluna. As sucessivas rotações aplicadas a $(\mathbf{x}_{mi} - \mathbf{p}(k))$ podem ser concatenadas em uma única transformação, tal que utilizando quaternions, pode-se escrever:

$$R_B^C(R_W^B(\mathbf{x}_{mi} - \mathbf{p}(k))) = R_W^C(\mathbf{x}_{mi} - \mathbf{p}(k))$$
(2.33)

$$= \mathbf{q}_W^C \otimes (\mathbf{x}_{mi} - \mathbf{p}(k)) \otimes \overline{\mathbf{q}}_W^C.$$
 (2.34)

Convertendo a rotação especificada por \mathbf{q}_W^C em uma matriz de cossenos diretores, é fácil perceber que

$$\frac{\partial R_B^C(R_W^B(\mathbf{x}_{mi} - \mathbf{p}(k)))}{\partial \mathbf{p}(k)} = \frac{\partial (\mathbf{q}_W^C \otimes (\mathbf{x}_{mi} - \mathbf{p}(k)) \otimes \overline{\mathbf{q}}_W^C)}{\partial \mathbf{p}(k)}$$
(2.35)

$$= \frac{\partial \left(\mathbf{C}_{W}^{C} \left(\mathbf{x}_{mi} - \mathbf{p}(k) \right) \right)}{\partial \mathbf{p}(k)}$$
(2.36)

$$= -\mathbf{C}_{W}^{C} \tag{2.37}$$

Já a derivada em função da orientação $\mathbf{q}(k)$ pode ser expressa como

$$\frac{\partial R_B^C(R_W^B(\mathbf{x}_{mi} - \mathbf{p}(k)))}{\partial \mathbf{q}(k)} = R_B^C\left(\frac{\partial R_W^B(\mathbf{x}_{mi} - \mathbf{p}(k))}{\partial \mathbf{q}(k)}\right)$$
(2.38)

$$= \mathbf{C}_{B}^{C} \frac{\partial R_{W}^{B}(\mathbf{x}_{mi} - \mathbf{p}(k))}{\partial \mathbf{q}(k)}$$
(2.39)

$$= \mathbf{C}_{B}^{C} \frac{\partial \mathbf{C}_{W}^{B} \left(\mathbf{x}_{mi} - \mathbf{p}(k) \right)}{\partial \mathbf{q}(k)}.$$
(2.40)

É importante observar que a derivada expressa na equação 2.40 é novamente equivalente à definida em 2.17, utilizando o conjugado de $\mathbf{q}(k)$ para gerar \mathbf{C}_W^B e substituindo α por $(\mathbf{x}_{mi} - \mathbf{p}(k))$. Deste modo,

com

$$\mathbf{d}_i(k) = \mathbf{x}_{mi} - \mathbf{p}(k) \tag{2.41}$$

$$= [d_x, d_y, d_z]^T, (2.42)$$

as equações 2.40 e 2.17 permitem determinar a derivada em relação à atitude do dirigível, $\mathbf{q}(k)$ (vide equação 2.2):

$$\frac{\partial R_B^C(R_W^B(\mathbf{d}_i(k)))}{\partial \mathbf{q}(k)} = 2 \mathbf{C}_B^C \begin{bmatrix} d_x q_r + d_y q_k - d_z q_j & -d_x q_k + d_y q_r + d_z q_i & d_x q_j - d_y q_i + d_z q_r \\ d_x q_i + d_y q_j + d_z q_k & d_x q_j - d_y q_i + d_z q_r & d_x q_k - d_y q_r - d_z q_i \\ d_y q_i - d_x q_j - d_z q_r & d_x q_i + d_y q_j + d_z q_k & d_x q_r + d_y q_k - d_z q_j \\ -d_x q_k + d_y q_r + d_z q_i & d_z q_j - d_x q_r - d_y q_k & d_x q_i + d_y q_j + d_z q_k \end{bmatrix}^T,$$
(2.43)

Resta desenvolver a derivada em relação à característica do mapa \mathbf{x}_{mi} correspondente à observação \mathbf{z}_i . Analisando a equação 2.31, percebe-se que

$$\frac{\partial R_B^C(R_W^B(\mathbf{x}_{mi} - \mathbf{p}(k)))}{\partial \mathbf{x}_{mi}} = -\frac{\partial R_B^C(R_W^B(\mathbf{x}_{mi} - \mathbf{p}(k)))}{\partial \mathbf{p}(k)}$$

e, portanto,

$$\frac{\partial R_B^C(R_W^B(\mathbf{x}_{mi} - \mathbf{p}(k)))}{\partial \mathbf{x}_{mi}} = \mathbf{C}_W^C$$
(2.44)

2.6 Estimação

De posse do modelo do sistema – que, para os efeitos deste estudo, compreende o modelo cinemático do dirigível a partir dos sinais da IMU e o modelo da observação das características do ambiente por uma câmera embarcada – e das leituras dos sensores do veículo robótico aéreo, resta agora desenvolver uma metodologia para fundir as informações tanto da IMU (proprioceptivas, inerciais) quanto do sistema de visão (exteroceptivas, referentes à detecção das balizas), gerando uma estimativa mais precisa da postura do dirigível e da localização das balizas que compõem o mapa, tal que o resultado seja um sistema de SLAM para o robô aéreo.

Nesta seção, é descrita a estratégia de fusão de dados e a conseqüente estimação utilizando um filtro de Kalman estendido. Este processo de estimação, inicialmente desenvolvido em (Smith et al., 1987) e amplamente utilizado desde então (Dissanayake et al., 2001; Panzieri et al., 2002; Langelaan and Rock, 2005), é representado na figura 2.6. Iniciando pela leitura da IMU, na *k*-ésima iteração, o

T

modelo cinemático do robô aéreo, desenvolvido na seção 2.3, é utilizado, juntamente com as informações obtidas da IMU ($\mathbf{u}(k) = [\alpha(k), \omega(k)]$), para computar a predição da localização do dirigível e do mapa $\hat{\mathbf{x}}(k|k-1)$. Além disso, dada a necessidade de manter atualizado o modelo do erro do estado do sistema, é também alterada a matriz de covariância $\mathbf{P}(k|k-1)$, que representa as incertezas de primeira ordem presentes em $\hat{\mathbf{x}}(k|k-1)$. Como os dados da IMU contêm erros e não fornecem informações georreferenciadas (sendo a IMU proprioceptiva), a incerteza das leituras da IMU é refletida apenas na incerteza do estado do veículo, que aumenta.

De posse da predição do estado do sistema, uma predição da observação $h(\hat{\mathbf{x}}(k|k-1))$ é então calculada e comparada à observação real $\mathbf{z}(k)$, obtida do sistema de visão. Esta comparação é feita por um subsistema que utiliza $\mathbf{P}(k|k-1)$ e o modelo de incerteza da própria observação para calcular a distância entre cada predição e observação, associando uma à outra para detectar quais das características observadas na *k*-ésima iteração já estão presentes no mapa, separando-as das demais (que estão sendo observadas pela primeira vez).

A observação das características já presentes no mapa (e associadas às predições correspondentes) é então utilizada para atualizar $\hat{\mathbf{x}}(k|k-1)$ e $\mathbf{P}(k|k-1)$, gerando a estimativa do estado do sistema $\hat{\mathbf{x}}(k|k)$ e da matriz de covariância $\mathbf{P}(k|k)$, sendo que a incerteza deve diminuir, já que a postura do dirigível com relação ao mapa pode ser observada e, portanto, corrigida.

Já as observações não associadas a uma característica são utilizadas para incrementar o vetor de estado com novas entradas do mapa. O aumento da dimensão do vetor de estado deve ser seguido do redimensionamento da matriz de covariância. Como a estimativa desta nova característica é dada a partir da postura do dirigível e da própria observação (conforme observa-se também na figura 2.6), qualquer erro na estimativa da postura é refletido diretamente no erro de localização da baliza. No entanto, de modo análogo, a correção da postura do dirigível deve permitir a correção da localização da baliza, indicando uma alta correlação entre os valores, que deve ser traduzida para a matriz de covariância $\mathbf{P}(k|k)$ durante a adição de características ao mapa.

2.6.1 Filtro de Kalman estendido

O filtro de Kalman (Kalman, 1960) é uma ferramenta para a estimação do estado de um sistema através da fusão de informações. Ele é aplicável a processos em que tanto a transição de estado quanto as observações do estado são lineares e as incertezas associadas são Gaussianas. Nesta situação, como um algoritmo recursivo estatístico, o filtro de Kalman fornece uma estimativa ótima do estado no instante k dadas as observações até o instante k; ou seja, produz estimativas com erro quadrado médio mínimo (minimiza o traço da matriz de covariância do estado).

No entanto, a maioria dos sistemas reais – aqui incluso também o sistema deste trabalho – não é governada por equações lineares, de modo que a forma básica do filtro de Kalman não pode ser



Fig. 2.6: Estrutura do sistema de SLAM. Cada círculo representa uma etapa do sistema (aquelas referentes ao filtro de Kalman estendido estão destacadas) e as setas indicam o fluxo de dados.

aplicada diretamente. Para acomodar a esta limitação, foi desenvolvido o filtro de Kalman estendido (EKF) (Anderson and Moore, 1979; Grewal and Andrews, 1993). No EKF, as funções dos modelos de processo e observação são linearizadas em torno do ponto da estimativa do estado. Deste modo, diferentemente do que ocorre no filtro de Kalman convencional, a estimativa da variância do estado não é independente da estimativa do próprio estado, uma vez que estes estão acoplados pelos cálculos dos Jacobianos.

Deste modo, a função que descreve a transição de estado é dada por

$$\mathbf{x}(k) = f(\mathbf{x}_{\nu}(k-1), \mathbf{u}(k), \mathbf{w}(k)), \qquad (2.45)$$

enquanto a função de observação é dada por

$$\mathbf{z}(k) = h(\mathbf{x}(k)) + \mathbf{v}(k). \tag{2.46}$$

Assim, a metodologia do EKF define a predição do estado e da covariância associada pelas equa-

ções

$$\hat{\mathbf{x}}(k|k-1) = f(\hat{\mathbf{x}}_{\nu}(k-1|k-1), \mathbf{u}(k), \mathbf{0})$$
(2.47)

$$\mathbf{P}(k|k-1) = \nabla f_x(k) \mathbf{P}(k-1|k-1) \nabla f_x^T(k) + \nabla f_u(k) \mathbf{Q}(k) \nabla f_u^T(k), \qquad (2.48)$$

onde o termo $\nabla f_x(k)$ é o Jacobiano da função de transição de estado não linear em relação a estimativa anterior do estado, $\mathbf{x}(k|k-1)$, e o termo $\nabla f_u(k)$ é a função de transferência de ruído linearizada, computada como o Jacobiano da mesma função em relação aos sinais de entrada $\mathbf{u}(k)$.

Finalizando a metodologia, a atualização do estado e da covariância, após uma observação, é dada por

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{W}(k)\mathbf{v}(k), \qquad (2.49)$$

$$\mathbf{P}(k|k) = \mathbf{P}(k|k-1) - \mathbf{W}(k)\mathbf{S}(k)\mathbf{W}(k)^{T}, \qquad (2.50)$$

onde v(k) é o vetor de inovação formado pela subtração da predição da observação com ruído zero da observação medida, corforme

$$\mathbf{v}(k) = \mathbf{z}(k) - h(\hat{\mathbf{x}}, \mathbf{0}). \tag{2.51}$$

Enfim, a covariância da inovação S e o ganho do sistema W são dados por

$$\mathbf{S}(k) = \nabla h_{\mathbf{x}}(k)\mathbf{P}(k|k-1)\nabla h_{\mathbf{x}}^{T}(k) + \mathbf{R}(k)$$
(2.52)

$$\mathbf{W}(k) = \mathbf{P}(k|k-1)\nabla h_{\mathbf{X}}^{T}(k)\mathbf{S}^{-1}(k), \qquad (2.53)$$

onde o termo $\nabla h_{\mathbf{x}}(k)$ representa o Jacobiano do modelo não linear de observação com relação à estimativa anterior do estado.

O EKF opera a partir do pressuposto de que as equações $\mathbf{f} \in \mathbf{h}$ são quase lineares, tal que as transformações linearizadas aproximam satisfatoriamente o valor correto das transformações das covariâncias. Assim como o filtro de Kalman convencional, o EFK também assume todas as incertezas com distribuições Gaussianas. Caso a não linearidade dos modelos seja muito acentuada ou os erros possuam distribuições muito distantes de Gaussianas, o desempenho do sistema de estimação pode tornar-se ineficiente.

Com base no descrito acima, nas seções seguintes cada estágio do sistema de estimação representado na figura 2.6 é descrito em detalhes.

2.6.2 Predição

A predição do vetor de estado do dirigível no instante k é obtida diretamente da equação 2.10, derivada na seção 2.3. Como as características que compõem o mapa são consideradas estáticas com relação ao referencial de terra, a equação de transição de estado do mapa é dada por

$$\hat{\mathbf{x}}_m(k|k-1) = \hat{\mathbf{x}}_m(k-1),$$
(2.54)

o que leva à equação completa de transição do estado do sistema:

$$\hat{\mathbf{x}}(k|k-1) = \begin{bmatrix} \hat{\mathbf{x}}_{\nu}(k|k-1) \\ \hat{\mathbf{x}}_{m}(k|k-1) \end{bmatrix}$$
(2.55)

$$= \begin{bmatrix} f(\hat{\mathbf{x}}_{\nu}(k-1|k-1), \mathbf{u}(k), \mathbf{0}) \\ \hat{\mathbf{x}}_{m}(k-1|k-1) \end{bmatrix}, \qquad (2.56)$$

tal que $f(\cdot, \cdot, \cdot)$ é o modelo cinemático do dirigível da equação 2.6, com $\mathbf{u}(k)$ sendo a leitura da IMU e considerando o ruído do sistema nulo ($\mathbf{w}(k) = 0$).

No entanto, como já foi mencionado, a predição implica em um aumento na incerteza da postura do dirigível, já que as leituras da IMU contém erros. Portanto, a matriz de covariância deve ser atualizada para refletir esta mudança. Isto é realizado através da seguinte equação:

$$\mathbf{P}(k|k-1) = \nabla f_x \, \mathbf{P}(k-1|k-1) \, \nabla f_x^T + \nabla f_u \, \mathbf{Q}(k) \, \nabla f_u^T, \qquad (2.57)$$

em que ∇f_x e ∇f_u são os Jacobianos de $f(\cdot, \cdot, \cdot)$ em relação ao vetor do estado e à entrada $\mathbf{u}(k)$ (composta dos sinais da IMU), respectivamente, e a matriz diagonal $\mathbf{Q}(k)$ define a covariância do erro na leitura da IMU, modelada como não-correlata e de média zero, sendo portanto definida por:

$$\mathbf{Q}(k) = \begin{bmatrix} \sigma_{\alpha}^2 & 0\\ 0 & \sigma_{\omega}^2 \end{bmatrix}.$$
 (2.58)

O primeiro termo da soma na equação 2.57 representa como a incerteza presente em $\hat{\mathbf{x}}(k-1|k-1)$ é propagada para $\hat{\mathbf{x}}(k|k-1)$ frente à variação de posição do dirigível; o segundo termo corresponde à covariância de $\mathbf{w}(k)$ da equação 2.6 e indica como o erro da leitura da IMU, $\mathbf{u}(k)$, de variância $\mathbf{Q}(k)$, aumenta a incerteza do estado do sistema.

Como o mapa é considerado estacionário, os Jacobianos ∇f_x e ∇f_u possuem diversos elementos

nulos e a equação 2.57 pode ser calculada de maneira mais eficiente:

$$\mathbf{P}(k|k-1) = \begin{bmatrix} \nabla f_{x_{v}} \mathbf{P}_{vv}(k-1|k-1) \nabla f_{x_{v}}^{T} + \nabla f_{u} \mathbf{Q}(k) \nabla f_{u}^{T} & \nabla f_{x_{v}} \mathbf{P}_{vm}(k-1|k-1) \\ (\nabla f_{x_{v}} \mathbf{P}_{vm}(k-1|k-1))^{T} & \mathbf{P}_{mm}(k-1|k-1) \end{bmatrix}, \quad (2.59)$$

com ∇f_x calculado conforme a equação 2.15 e ∇f_u calculado a partir da equação 2.20.

2.6.3 Atualização

Com o intuito de atualizar a predição $\hat{\mathbf{x}}(k|k-1)$, é utilizado o sistema de visão, que faz uso de uma câmera para detectar a posição relativa entre a própria câmera e as balizas presentes no ambiente. Para tanto, são utilizadas as equações derivadas na seção 2.4.

Se alguma baliza for observada e associada a uma característica presente no mapa (conforme será detalhado na seção 2.6.4), é calculado o vetor de inovação v(k), formado da união das correspondências entre predições e observações:

$$\mathbf{v}(k) = \begin{bmatrix} \mathbf{z}_{i_1} - h(\hat{\mathbf{x}}_v, \hat{\mathbf{x}}_{mj_1}) \\ \mathbf{z}_{i_2} - h(\hat{\mathbf{x}}_v, \hat{\mathbf{x}}_{mj_2}) \\ \vdots \\ \mathbf{z}_{i_n} - h(\hat{\mathbf{x}}_v, \hat{\mathbf{x}}_{mj_n}) \end{bmatrix}, \qquad (2.60)$$

onde cada linha do vetor v(k) é um vetor tridimensional correspondente à diferença entre a observação de uma baliza e a predição da observação da mesma, tal que para *n* observações, v(k) tem dimensão 3*n*.

A partir do modelo de observação do sistema de visão, é possível então calcular a incerteza desta inovação no instante k, S(k), que está relacionada ao estado do sistema e aos erros de observação:

$$\mathbf{S}(k) = \nabla h_{\mathbf{X}}(k)\mathbf{P}(k|k-1)\nabla h_{\mathbf{X}}^{T}(k) + \mathbf{R}(k), \qquad (2.61)$$

onde $\nabla h_{\mathbf{x}}(k)$ é o Jacobiano de $h(\mathbf{x})$ em relação a $\hat{\mathbf{x}}(k|k-1)$, definido na equação 2.32, e $\mathbf{R}(k)$ é a incerteza do sensor, modelada como uma matriz de covariância diagonal, já explicitada em 2.30.

Se o vetor de inovação não for de dimensão nula, ou seja, se alguma associação for feita, não apenas o estado é atualizado mas também a incerteza tanto do mapa quanto da postura do dirigível podem ser reduzidas, já que a correspondência entre as observações e as predições tornam observáveis

os erros de posicionamento relativos ao mapa:

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{W}(k)\mathbf{v}(k), \qquad (2.62)$$

$$\mathbf{P}(k|k) = \mathbf{P}(k|k-1) - \mathbf{W}(k)\mathbf{S}(k)\mathbf{W}(k)^{T}, \qquad (2.63)$$

onde a matriz de ganho $\mathbf{W}(k)$ é obtida de

$$\mathbf{W}(k) = \mathbf{P}(k|k-1)\nabla h_{\mathbf{X}}^{T}(k)\mathbf{S}^{-1}(k)$$
(2.64)

e a matriz S(k) é a mesma da equação 2.61.

É importante ressaltar que, após a atualização de $\hat{\mathbf{x}}(k|k)$, não há garantia de que o quaternion $\hat{\mathbf{q}}(k|k)$) continuará representando uma rotação; ou seja, que sua norma será unitária. Deste modo, o quaternion deve ser sempre normalizado após a atualização:

$$\hat{\mathbf{x}}(k|k)_{norm} = \begin{bmatrix} \hat{\mathbf{p}}(k|k) \\ \hat{\mathbf{v}}(k|k) \\ \frac{\hat{\mathbf{q}}(k|k)}{||\hat{\mathbf{q}}(k|k)||} \end{bmatrix}$$
(2.65)

2.6.4 Correspondência entre observações e balizas

O processo de associação das observações às balizas é essencial para o funcionamento do sistema. Sempre que uma associação não é feita, o sistema perde eficiência, já que a posição real da baliza será novamente calculada e no mapa haverá duas representações da mesma posição real. Apesar de existirem algoritmos para detectar tais situações e corrigir o mapa final, há um limite para o número de ocorrências deste tipo e para a disparidade entre as características repetidas. Por outro lado, uma correspondência errada pode corromper seriamente o sistema, ao ponto de impedi-lo de funcionar, já que as covariâncias estarão incorretas (como será mais detalhado na próxima seção). Um erro deste tipo não pode ser tolerado e, conceitualmente, nunca deve ocorrer.

Neste trabalho, o mapa é composto de características idênticas, representadas por pontos e, portanto, distinguíveis apenas por sua posição. Além disso, assume-se que o erro presente nas observações é gaussiano, o que significa que qualquer observação pode ser associada a qualquer característica (já que a cauda da gaussiada é infinita, nunca chegando a zero).

Mesmo assim, é importante rejeitar as associações pouco prováveis. Para tanto o procedimento de associação é feito no espaço de observação das características. Seja

$$\mathbf{z} = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n\}$$

o conjunto das observações realizadas no instante k e

$$\mathbf{h} = \{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_m\}$$

o conjunto das predições das observações no mesmo instante, cada uma correspondendo a uma diferente característica do mapa. Para analisar a correspondência entre \mathbf{z}_i $(1 \le i \le n)$ e \mathbf{h}_j $(1 \le j \le m)$, é utilizado um intervalo estatístico de validação, calculado pela *distância de Mahalanobis* (Kim, 2004) (noção introduzida por Prasanta Chandra Mahalanobis em 1936). Esta distância γ é computada como

$$\gamma_{i,j} = \mathbf{v}_{i,j}^T \, \mathbf{S}_j^{-1}(k) \, \mathbf{v}_{i,j},$$
(2.66)

em que $v_{i,j}$ é o vetor de inovação de uma única associação, entre a *i*-ésima observação e a *j*-ésima predição, e é definido como

$$\mathbf{v}_{i,j} = \mathbf{z}_i - h(\hat{\mathbf{x}}_v, \hat{\mathbf{x}}_{mj}). \tag{2.67}$$

Já $\mathbf{S}_{j}(k)$ é a covariância da inovação $v_{i,j}$, calculada como em 2.61, mas utilizando $h(\hat{\mathbf{x}}_{v}, \hat{\mathbf{x}}_{mj})$:

$$\mathbf{S}_{j}(k) = \mathbf{E}[\mathbf{v}_{i,j} \, \mathbf{v}_{i,j}^{T}]$$
(2.68)

$$= \nabla h_{\mathbf{v},\mathbf{mj}}(k) \mathbf{P}(k|k-1) \nabla h_{\mathbf{v},\mathbf{mj}}^{T}(k) + \mathbf{R}_{i}(k), \qquad (2.69)$$

Como as inovações possuem distribuição gaussiana, γ forma uma distribuição qui-quadrado (χ_3^2). A forma da distribuição χ_3^2 depende da dimensão da inovação. Para o sistema de visão aqui utilizado, como a observação é dada pela posição tridimensional da baliza em relação à câmera, uma confiabilidade de 95%, por exemplo, equivale a uma *distância de Mahalanobis* de 12,8 (obtida das tabelas da distribuição). Desta forma, é escolhido um limiar δ para esta distância e são rejeitadas todas as correspondências tal que

$$\gamma_{i,j} > \delta. \tag{2.70}$$

O intervalo de validação δ pode ser visto como um elipsóide cujo centro está na predição da observação. Uma observação pode ser associada à predição se estiver posicionada dentro da elipse. Ainda assim, como pode ser notado na figura 2.7, pode haver ambigüidades nas associações. Uma alternativa seria aumentar a rejeição, diminuindo δ , mas isso poderia acarretar na não associação da observação de uma baliza presente no mapa. Ao invés disso, se qualquer predição ou observação for associada mais de uma vez, é necessário resolver a ambigüidade. O método mais comum para solucionar este problema, e que é utilizado neste trabalho, é fazer a correspondência com o *vizinho mais próximo* (Bailey, 2002). Dado um conjunto de possíveis correspondências $c = \{c_1, c_2, ..., c_n\}$, a verossimilhança Λ_k de cada associação $c_k \in c$ pode ser calculada a partir da função de distribuição



Fig. 2.7: Exemplo 2D da correspondência das observações e da ambigüidade de predições e observações. As predições $h(\cdot, \cdot)$ possuem elipses indicando o limite δ . As observações são representadas por \mathbf{z}_a e \mathbf{z}_b . As setas indicam a distância de Mahalanobis. Percebe-se que \mathbf{z}_a é uma observação válida para as duas predições, enquanto \mathbf{z}_a e \mathbf{z}_b podem ambas corresponder a $h(\mathbf{x}_v, \mathbf{x}_{mi})$.

de probabilidade da inovação de c_k , v_k :

$$\Lambda_{k} = \frac{1}{(2\pi)^{n/2}\sqrt{|\mathbf{S}_{k}|}} \exp\left(-\frac{1}{2}\,\mathbf{v}_{k}^{T}\,\mathbf{S}_{k}^{-1}\,\mathbf{v}_{k}\right).$$
(2.71)

Com uma métrica equivalente, tomando o logaritmo da equação acima:

$$\ln\left(\frac{1}{(2\pi)^{n/2}\sqrt{|\mathbf{S}_k|}}\exp\left(-\frac{1}{2}\mathbf{v}_k^T\mathbf{S}_k^{-1}\mathbf{v}_k\right)\right) = -\frac{1}{2}\mathbf{v}_k^T\mathbf{S}_k^{-1}\mathbf{v}_k - \ln\left((2\pi)^{n/2}\sqrt{|\mathbf{S}_k|}\right) \quad (2.72)$$

$$= -\frac{1}{2} \mathbf{v}_{k}^{T} \mathbf{S}_{k}^{-1} \mathbf{v}_{k} - \frac{n}{2} \ln(2\pi) - \frac{1}{2} \ln|\mathbf{S}_{k}|. \quad (2.73)$$

Eliminando o elemento constante e fazendo outras pequenas transformações, obtém-se a distância normalizada N_k , que minimizada indica a correspondência mais provável.

$$N_{k} = \mathbf{v}_{k}^{T} \mathbf{S}_{k}^{-1}(k) \, \mathbf{v}_{k}, +\ln|\mathbf{S}_{k}(k)|.$$
(2.74)

Deste modo, a metodologia de associação escolhe no conjunto c a correspondência que minimiza N_k . Vale notar que este método é também conhecido como associação de *máxima verossimilhança*, dado o fato de que minimizar a equação 2.74 equivale a maximizar a equação 2.71.

2.6.5 Aumentando o Estado

As observações não associadas representam novas características a serem incluídas no mapa, de modo que, quando forem percebidas novamente pelo sistema de visão, possam ser utilizadas na atualização do estado do sistema. O processo de extensão do mapa (Bailey, 2002; Kim, 2004), tal que este seja construído incrementalmente, é realizado a partir da estimativa do estado do sistema no momento da observação \mathbf{z}_i e da inversa do modelo de observação, apresentada na equação 2.23. Deste modo,

$$\hat{\mathbf{x}}_{aug}(k|k) = f_a(\hat{\mathbf{x}}(k), \hat{\mathbf{z}}_i(k)) = \begin{bmatrix} \hat{\mathbf{x}}_v(k) \\ \hat{\mathbf{x}}_m(k) \\ g(\hat{\mathbf{x}}_v(k), \mathbf{z}_i(k)) \end{bmatrix}, \qquad (2.75)$$

 $\operatorname{com} g(\cdot, \cdot)$ definido conforme a equação 2.23.

Atualizar a matriz de covariância, no entanto, não é tão simples. Quando uma característica é adicionada ao mapa, os erros de estimação da postura do dirigível são refletidos diretamente no erro de localização da característica, já que a nova característica é dada por $g(\hat{\mathbf{x}}_v(k), \mathbf{z}_i(k))$ e portanto depende de $\hat{\mathbf{x}}_v(k)$. Isto implica em uma alta correlação entre a localização da baliza e a postura do dirigível, o que quer dizer que a incerteza presente no estado do veículo deve ser propagada para a incerteza da recém adicionada baliza.

Para acomodar esta situação, a matriz $\mathbf{P}(k)$ deve mudar de acordo com:

$$\mathbf{P}_{aug}(k|k) = \nabla f_a \begin{bmatrix} \mathbf{P}(k|k) & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_i(k) \end{bmatrix} \nabla f_a^T$$
(2.76)

onde o Jacobiano ∇f_a é dado por

$$\nabla f_a = \begin{bmatrix} \mathbf{I}_v & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_v & \mathbf{0} \\ \nabla g_{x_v} & \mathbf{0} & \nabla g_{z_i} \end{bmatrix}.$$
 (2.77)

Observando que, à exceção das últimas linhas e colunas referentes à nova característica do mapa, a matriz Jacobiano é diagonal, é possível calcular a transformação que leva a $\mathbf{P}_{aug}(k|k)$ de maneira computacionalmente mais eficiente:

$$\mathbf{P}_{aug}(k|k) = \begin{bmatrix} \mathbf{P}_{v} & \mathbf{P}_{vm} & (\nabla g_{x_{v}} \mathbf{P}_{v})^{T} \\ \mathbf{P}_{vm}^{T} & \mathbf{P}_{m} & (\nabla g_{x_{v}} \mathbf{P}_{vm})^{T} \\ \nabla g_{x_{v}} \mathbf{P}_{v} & \nabla g_{x_{v}} \mathbf{P}_{vm} & \nabla g_{x_{v}} \mathbf{P}_{v} \nabla g_{x_{v}}^{T} + g_{z_{i}} \mathbf{R}_{i} \nabla g_{z_{i}}^{T} \end{bmatrix}$$
(2.78)

A relação entre a incerteza da postura do veículo e a incerteza da posição da baliza adicionada ao mapa pode ser percebida em detalhes na equação 2.78. Conforme o sistema de SLAM é executado, no entanto, o modelo demonstrado acima evolui: como já foi mencionado, se uma baliza já presente no mapa é observada novamente, os erros provenientes da IMU tornam-se observáveis – já que a postura do dirigível pode ser calculada novamente – e, portanto, podem ser estimados. Quando isto ocorre, a incerteza representada pelas matriz \mathbf{P}_v diminui e, com ela, a incerteza na localização das balizas, já que a postura do dirigível e a posição da baliza estão correlacionadas. A redução destes valores na matriz de covariância \mathbf{P} indicam sua dissociação, de forma que, conforme o sistema evolui e as balizas são repetidamente observadas, a relação entre o erro na postura do dirigível e o erro de posicionamento das balizas tende a zero. Neste ponto a estimativa da postura do dirigível aproximase do ideal e a posição das balizas no mapa não é mais corrigida mesmo que uma observação esteja distante de sua representação no mapa.

É digno de nota que, de acordo com a metodologia proposta, o mapa inteiro é sempre mantido no estado do sistema, de modo a garantir que balizas antigas sejam novamente detectadas posteriormente, o que permite uma correção significativa da estimativa de todas as variáveis do estado sistema. No entanto, muito embora a situação não ocorra no escopo deste trabalho, ambientes menos propícios para um sistema de visão podem provocar marcas ambíguas no mapa; ou seja, mais de uma representação para o mesmo ponto real. Corrigir esta situação envolve não apenas detectar os conjuntos de representações duplicadas, mas também computar a melhor estimativa para a combinação do conjunto em uma única representação utilizando para tanto a matriz de covariância do estado do sistema. No entanto, tal metodologia de reconhecimento e correção de duplicatas no mapa não é abordada neste trabalho.

Por outro lado, com esta representação de mapa e incerteza, remover balizas da representação do ambiente é simples, bastando excluir a baliza do estado e as linhas e colunas correspondentes da matriz de covariância. Deste modo, a remoção da *i*-ésima marca do estado, para um mapa com *N* balizas, resulta no estado

Í

$$\hat{\mathbf{x}}_{r}(k|k) = \begin{bmatrix} \hat{\mathbf{x}}_{v}(k|k) \\ \hat{\mathbf{x}}_{m1}(k|k) \\ \vdots \\ \hat{\mathbf{x}}_{m(i-1)}(k|k) \\ \hat{\mathbf{x}}_{m(i+1)}(k|k) \\ \vdots \end{bmatrix}$$
(2.79)

e na matriz de covariância

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{v,v} & \mathbf{P}_{v,m1} & \dots & \mathbf{P}_{v,m(i-1)} & \mathbf{P}_{v,m(i+1)} & \dots \\ \mathbf{P}_{m1,v} & \mathbf{P}_{m1,m1} & \dots & \mathbf{P}_{m1,m(i-1)} & \mathbf{P}_{m1,m(i+1)} & \dots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \\ \mathbf{P}_{m(i-1),v} & \mathbf{P}_{m(i-1),m1} & \dots & \mathbf{P}_{m(i-1),m(i-1)} & \mathbf{P}_{m(i-1),m(i+1)} & \dots \\ \mathbf{P}_{m(i+1),v} & \mathbf{P}_{m(i+1),m1} & \dots & \mathbf{P}_{m(i+1),m(i-1)} & \mathbf{P}_{m(i+1),m(i+1)} & \dots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \end{bmatrix} .$$
(2.80)

Com estas equações, o sistema de auto-localização e mapeamento representado pelo ciclo mostrado na figura 2.6 foi descrito completamente.

2.7 Conclusão

Neste capítulo foi apresentada uma metodologia para Localização e Mapeamento Simultâneos para um veículo robótico aéreo dotado de uma IMU e um sistema de visão que detecte a posição relativa de pontos do ambiente. Para descortinar a solução proposta, foram desenvolvidos o modelo cinemático do dirigível – a partir dos dados da IMU – e o modelo de observação das balizas do ambiente – que utiliza a câmera –, bem como os Jacobianos necessários para a utilização do EKF, que foi empregado na fusão sensorial. Foram descritos os algoritmos do ciclo de estimação, incluindo não apenas as etapas do EKF, mas também a correspondência entre as observações e o ajuste da dimensão do mapa.

Capítulo 3

Estrutura de software

Neste capítulo, é descrita em detalhes a estrutura de software tanto da implementação da metodologia definida no capítulo 2 quanto do ambiente de simulação utilizado na validação dos algoritmos de auto-localização e mapeamento simultâneos.

Muito embora os algoritmos de estimação sejam validados em simulação, o seu desenvolvimento representa um passo importante para a posterior implantação da metodologia na aeronave real. Deste modo, todo o projeto de desenvolvimento de software é orientado a objetos, na linguagem C++, cujo arcabouço permite grande modularidade e flexibilidade, podendo ser transferido do ambiente de simulação para o sistema embarcado sem modificações em sua estrutura. Do mesmo modo, o ambiente de simulação foi elaborado tal que a interface com o módulo de estimação possa ser implementada no sistema embarcado, tornando a transição transparente ao módulo de auto-localização e mapeamento.

Na primeira seção do capítulo (3.1), são descritas as funcionalidades da biblioteca matemática sobre a qual foram apoiados os algoritmos de estimação e simulação. A seguir, na seção 3.2, são definidas as classes que determinam a estrutura do módulo de auto-localização e mapeamento, ressaltando sua versatilidade. Já em 3.3, é delineada a implementação do ambiente de simulação, compreendendo a dinâmica do dirigível e um conjunto de sensores virtuais. Por fim, na seção 3.4, é realçada a maneira como os módulos interagem durante a estimação da posição da aeronave e do mapa.

Além disso, também referente a desenvolvimento de software, no apêndice A é descrito um sistema de gravação sincronizada de imagens da câmera embarcada e sinais provenientes da IMU.

3.1 Biblioteca matemática

A biblioteca matemática é a fundação tanto do módulo de auto-localização e mapeamento quanto do ambiente de simulação do dirigível. Compõem a biblioteca classes para a manipulação de vetores e matrizes, operações com quaternions e representações para funções matemáticas. A seguir, é detalhada a estrutura de software utilizada na implementação desta biblioteca.

3.1.1 Vetores, matrizes e quaternions



Fig. 3.1: A figura mostra a representação dos vetores, matrizes e quaternions. A dupla implementação de matrizes e vetores agiliza o desenvolvimento e aumenta a velocidade das operações envolvendo os vetores tridimensionais.

A base matemática do sistema de SLAM e do ambiente de simulação consiste de operações e transformações envolvendo vetores e matrizes. No entanto, as funcionalidades exigidas pelo do ambiente de simulação são muito diferentes das necessárias ao módulo de fusão sensorial e estimação para SLAM. Enquanto a matemática vetorial e matricial do ambiente de simulação consiste basicamente de aritmética e transformações geométricas tridimensionais, o módulo de auto-localização e mapeamento compreende também vetores de dimensionalidade variável e algoritmos mais complexos, como cálculo de inversas de matrizes.

Por este motivo, no intuito de tornar o desenvolvimento mais simples e melhorar o desempenho,



Fig. 3.2: Implementações da interface de função matemática. A generalidade obtida facilita a implementação no dirigível real, já que os modelos dos sensores são passíveis de alteração.

visando também a preparação para a futura implantação no dirigível real, a biblioteca matemática possui duas representações diferentes de matrizes e vetores, que se interligam através de rotinas de conversão e da interface em comum que ambos implementam. A figura 3.1 mostra a estrutura deste módulo matemático.

Um dos conjuntos, voltado para o estimador, representa matrizes e vetores cuja dimensão é passível de variação, tanto na inicialização quanto posteriormente, através da remoção ou inclusão de novos elementos. Além disso, estas classes contêm outras operações de grande utilidade em algoritmos de estimação, como por exemplo o cálculo da inversa e da pseudo-inversa de uma matriz.

O outro grupo de classes, direcionado para computação gráfica, consiste de vetores e matrizes tridimensionais. A este grupo é acrescida a classe que implementa quaternions, que interage com os vetores e matrizes tridimensionais para representar atitude e aplicar rotações. Dada a limitação dimensional das classes, mesmo os algoritmos mais simples são realizados com maior velocidade, o que aponta uma das razões para esta segunda representação vetorial.

3.1.2 Funções matemáticas

O desenvolvimento de uma interface de software para a representação de funções matemáticas deriva diretamente da busca por generalidade. Se uma função é implementada em um método da classe que a utiliza, esta classe torna-se intrinsicamente atrelada à função, de modo que uma alteração da função implica na criação de uma nova classe, muito embora sua funcionalidade não seja, em essência, diferente.

Ao utilizar uma classe de interface para representar funções matemáticas, as funções em questão tornam-se atributos das classes que as utilizam. Como uma invocação passa a ser feita através do

objeto que instancia a classe, trocando-se o objeto pode-se trocar a função. Na prática, isto significa que uma substituição deste tipo fica transparente para o restante do sistema em que a função está inserida.

Qualquer classe que estenda a interface de função matemática pode, por exemplo, ser utilizada durante o processo de estimação, representando o modelo cinemático do dirigível ou o modelo de observação do estado do sistema. Isto na prática significa que estes modelos podem ser trocados com total independência do restante da implementação do sistema de auto-localização e mapeamento. Deste modo, um modelo mais preciso de um sensor, ou mesmo o modelo de um novo sensor, pode ser integrado ao sistema com facilidade, sem alterações no estimador em si.

Para que tal funcionalidade seja atingida, a interface de software da função matemática contém três métodos:

- Aplicação: calcula o resultado da aplicação da função a partir de dois vetores passados como parâmetros. No caso da metodologia de fusão sensorial já detalhada no capítulo 2, os dois parâmetros normalmente representam o estado do sistema e a leitura de um sensor;
- Jacobiano com relação ao primeiro parâmetro: também recebe dois parâmetros e retorna a matriz Jacobiano da função em relação às variáveis do primeiro deles, que normalmente representa o estado do sistema;
- Jacobiano com relação ao segundo parâmetro: recebe dois parâmetros e retorna o Jacobiano com relação ao segundo, que representa a leitura de algum dos sensores.

É interessante notar que, no escopo da metodologia de fusão sensorial estabelecida, trata-se também da inversa da função de observação. Muito embora a inversa da função de observação seja implicitamente determinada quando definido o modelo de observação em si, optou-se por excluir da interface métodos para tratar da inversa de uma função. Isto significa que a função inversa deve ser representada em outra implementação da interface.

Deste modo, conforme mostra a figura 3.2, para o desenvolvimento do sistema as funções dos modelos cinemático e de observação foram desenvolvidas estendendo a interface especificada. Na tabela 3.1, são exibidas as equações implementadas em cada classe.

Classe	Aplicação	Jacobiano 1	Jacobiano 2
Modelo cinemático (IMU)	2.6	2.15	2.20
Modelo de observação (câmera)	2.30	2.32	Х
Inversa do modelo de observação	2.23	2.27	2.26

Tab. 3.1: Tabela explicitando quais equações foram implementadas nas classes que estendem a interface de função matemática

3.2 Sistema de auto-localização e mapeamento

Como grande parte da estrutura de software está calcada em interfaces, permitindo a alteração independente de diversos algoritmos, o sistema de auto-localização e mapeamento na verdade consiste de um módulo de integração entre os sensores e o estimador.

Este módulo é utilizado para determinar qual estimador do estado do sistema executará a fusão sensorial entre IMU e sensor visual, e quais funções matemáticas serão utilizadas para representar o modelo cinemático, o modelo de observação e a inversa do modelo de observação. É este módulo também que recebe as requisições de atualização da estimativa do estado do sistema, de modo que ele armazena tanto a estimativa do estado como a incerteza a ela associada. Além disso, o módulo de auto-localização e mapeamento faz a ponte com as leituras dos sensores, contendo também o modelo de incerteza dos mesmos.

3.2.1 Estimador

O estimador é uma interface implementada, neste projeto, em uma classe que representa o filtro de Kalman estendido. Fazem parte da interface da classe os métodos de predição e de atualização do estado e da incerteza do sistema. O método de inicialização do estimador, também presente na interface, determina os três modelos necessários para o funcionamento da metodologia de fusão sensorial: cinemático, de observação e inverso da observação.

É interessante notar que, muito embora representem parte da estratégia de estimação adotada, as metodologias para a correspondência entre observações e predições e para o aumento do estado não são representadas como funções da interface do estimador. Isto acontece por duas razões: i) estas metodologias estão intimamente relacionadas à atualização do estado do sistema, e por isto compartilham parte do algoritmo de correção da postura do dirigível e do mapa, de modo que a invocação dos algoritmos durante a atualização implica em um ganho de desempenho; ii) a interface das funções e o momento da chamada podem variar conforme a implementação do estimador, o que não seria comportado em uma organização modular, com cada algoritmo implementado em uma classe diferente. Esta alternativa flexibilizaria a implementação de cada algoritmo mas tornaria rígida a estrutura geral da metodologia.

Assim, o método de predição determina o estado do sistema utilizando o modelo cinemático do sistema e corrige a covariância de acordo. O método de atualização, por sua vez, realiza a correspondência entre as características preditas e observadas, corrige o estado do sistema e a covariância utilizando o modelo de observação e aumenta o estado do mapa se possível.

A figura 3.3 mostra a relação entre as classes. Na implementação da classe que representa o EKF, foram implementadas as equações do filtro, tais quais definidas na seção 2.6.1



Fig. 3.3: Organização do sistema de SLAM. O Filtro de Kalman estende a interface do estimador, que está contido na classe que representa o sistema de SLAM em si.

3.3 Ambiente de simulação

Esta seção detalha o ambiente de simulação do dirigível robótico aéreo AS800, desenvolvido para a realização dos experimentos em fusão sensorial. O principal objetivo de tal ambiente é propagar para o módulo de auto-localização e mapeamento leituras sensoriais, similares às obtidas de sensores reais, embarcados na aeronave. Para que isto ocorra, não apenas a dinâmica do dirigível e a central inercial que a detecta devem ser consideradas, mas também a representação dos arredores da aeronave, já que o sensor visual é exteroceptivo e depende de características do espaço que o circunda para a obtenção de informações. Sempre que a simulação avança no tempo, todos os sensores e modelos devem ser atualizados de acordo, mantendo a coerência da representação virtual do sistema embarcado.

Outro aspecto de interesse do ambiente de simulação é a observação dos experimentos, de modo que a câmera virtual não se limita apenas a transmitir dados, mas também imagens em si, o que possibilita tanto a análise visual das imagens da câmera embarcada quanto o monitoramento da estratégia de estimação da postura do próprio dirigível, como será detalhado mais adiante.

Inicialmente, na seção 3.3.1, é descrita a dinâmica do dirigível e a maneira como esta é integrada a outros pontos do ambiente de simulação. A seguir, na seção 3.3.2 é especificada a estrutura de software da porta de visualização, que permite a geração de imagens do experimento. Já na seção 3.3.3, é detalhado o funcionamento do conjunto de sensores virtuais fornecidos pelo ambiente de simulação.

3.3.1 Dinâmica do dirigível

A dinâmica do dirigível utilizada no ambiente de simulação é extraída de um simulador já existente do AS800, desenvolvido para o projeto AURORA (Gomes and Ramos, 1998; de Paiva et al., 1999). Este modelo da dinâmica da aeronave, não linear em seis graus de liberdade, é o ponto de partida do ambiente de simulação, já que o resultado da aplicação dos comandos dos atuadores (deflexão das quatro superfícies aerodinâmicas além de vetorização e empuxo do grupo propulsor formado por dois motores) resulta na movimentação da aeronave e conseqüentemente na leitura dos sensores virtuais.

Através da aplicação da Lei de Newton, obtém-se a equação geral do modelo dinâmico:

$$\mathbf{M}\dot{\mathbf{v}} = \mathbf{F}_d(\mathbf{v}) + \mathbf{F}_a(\mathbf{v}) + \mathbf{F}_p + \mathbf{F}_g, \qquad (3.1)$$

onde **M** é a matriz de massa que inclui tanto a inércia do dirigível quanto os elementos de inércia virtuais associados à dinâmica de veículos flutuantes; $\mathbf{v} = [u, v, w, p, q, r]^T$ é o vetor das velocidades lineares (u, v, w) e angulares (p, q, r) da aeronave; \mathbf{F}_d é o vetor das dinâmicas contendo termos das forças de Coriolis e centrífuga, além das forças induzidas pelo vento; \mathbf{F}_a é o vetor de forças e momentos aerodinâmicos; \mathbf{F}_p é o vetor de momentos e forças propulsivas e \mathbf{F}_g representa as forças e momentos gravitacionais.

A meta do modelo matemático é exatamente o cálculo das três velocidades lineares e das três velocidades angulares, definidas pelo vetor **v**, que determinam completamente o movimento do dirigível, em relação aos eixos já definidos do sistema de coordenadas IMU (conforme especificado na seção 2.3). No entanto, o cálculo de **v** exige, além de informações sobre o ambiente e o próprio dirigível, dados dos propulsores, o que implica também na trajetória do vôo. Deste modo, uma rota é determinada para o dirigível, utilizando segmentos de reta de latitude, longitude e altitude específicas, enquanto que estratégias de controle são utilizadas para assegurar o seguimento desta trajetória pelo dirigível. O controle lateral utiliza um controlador PID descrito em (Azinheira et al., 2000), ao passo que o controle longitudinal envolve um controlador proporcional-derivativo (PD) com realimentação de estado e um controlador proporcional de velocidade (Bueno et al., 2002).

Esta relação da dinâmica do dirigível com o restante do ambiente de simulação é tratada em uma classe responsável pelo cálculo da postura de todos os objetos móveis da cena virtual. Para tanto, são utilizadas as acelerações lineares do dirigível, extraídas durante o cálculo das velocidades lineares, e as velocidades angulares, representadas pelos últimos três componentes de **v**. Como a postura da câmera em relação ao sistema de coordenadas da IMU é conhecida, a câmera também é posicionada precisamente com base nos dados do simulador.

3.3.2 Porta de visualização

A porta de visualização do ambiente de simulação possui duas finalidades principais: i) simular a câmera embarcada, produzindo imagens equivalentes às obtidas do sistema real; ii) proporcionar uma análise visual dos experimentos, tanto pela imagem da câmera embarcada quanto por imagens externas.

Assim como outros pontos chave da estrutura de software, a porta de simulação também foi modularizada, tal que outras técnicas de renderização possam ser implementadas sem impacto no restante do sistema. No entanto, na validação da metodologia exposta no capítulo 2, o renderizador utilizado é o POV-Ray (http://www.povray.org/). O POV-Ray possui um modelo de cena de fácil compreensão e produz imagens de altíssima qualidade, muito embora o tempo de renderização destas imagens seja longo (por volta de 30 segundos), impedindo sua obtenção em tempo real. Nas figuras 3.4 e 3.5 estão alguns exemplos de imagens geradas pelo POV-Ray a partir da descrição do mundo e do dirigível elaboradas para a simulação.



(a) Imagem superior do dirigível e dos arredores.



(b) A visão em um ângulo diferente mostra outros detalhes da cena.

Fig. 3.4: Renderizações do POV-Ray.

A interface de comunicação com o renderizador é, na verdade, muito simples e se dá através de dois métodos:

- posicionamento: a localização dos objetos móveis no mundo virtual, incluindo a própria câmera, é enviada à interface do renderizador, tal que a cena seja especificada;
- requisição de imagem: quando a imagem é requisitada ela é renderizada e retornada crua, sem nenhuma compressão, com 8 bits por canal de cor.

Além destes métodos, outros são utilizados para a inicialização, definindo a resolução da imagem, o aspecto e a distância focal da câmera.



(a) Imagem de um experimento. O dirigível translúcido mostra a postura estimada pelo sistema de auto-localização e mapeamento.



(b) Imagem obtida da câmera embarcada.

Fig. 3.5: Mais exemplos de renderizações do POV-Ray.

No entanto, como o POV-Ray é um programa independente e não uma biblioteca que possa ser anexada ao restante dos módulos, a implementação da interface do renderizador, para a interação com o ambiente de simulação, acontece através de um sistema de comunicação entre processos. Quando uma imagem é requisitada, o processo que executa o ambiente de simulação se divide em dois, requisita a renderização da cena, e aguarda um aviso de término e a imagem em si, que são enviados em um espaço de memória compartilhada, acessível de ambos os processos. O POV-Ray é finalizado e o ambiente de simulação continua executando, com a interface do renderizador propagando a imagem resultante.

3.3.3 Sensores virtuais

Até agora, com o simulador da dinâmica e a porta de visualização, o ambiente de simulação apenas exibe os movimentos do dirigível. Nesta seção, é definido o conjunto de sensores virtuais que simulam o sistema embarcado, tal que o módulo de auto-localização e mapeamento possa ser testado.

Sensores inerciais

A emulação dos sensores inerciais é simples, considerando a interação já existente com simulador da dinâmica do dirigível. Para que a coerência entre o simulador e os sensores seja mantida, os mesmos dados enviados ao restante do sistema, para a atualização da postura do dirigível, são enviados ao emulador da IMU. No entanto, ao responder uma requisição de leitura, o emulador insere no sinal obtido um ruído gaussiano, cuja média e desvio padrão são estabelecidos na inicialização do sensor virtual.

É importante ressaltar a relação do sensor com a passagem do tempo. A leitura da IMU tem como ponto de partida as informações obtidas do modelo dinâmico da aeronave e estão sempre em sincronia com o mesmo. Isto implica que uma requisição de leitura retorna sempre os mesmos valores muito similares até que o ambiente de simulação seja levado ao instante seguinte, quando a dinâmica do veículo pode ser alterada, e com ela a base das leituras da IMU virtual. Isto implica que, além do método utilizado para a obtenção de leituras dos sensores, um outro método, invocado pelo algoritmo de controle da simulação, atualiza a base das leituras da IMU conforme a simulação evolui.

Sensor visual

Um dos aspectos do sensor visual, já descrito na seção 3.3.2, é a obtenção de imagens da câmera virtual. A mesma metodologia utilizada para a visualização dos experimentos é empregada para a obtenção de imagens do sensor visual embarcado virtual, bastando para tanto ajustar sua posição e orientação em relação à postura do dirigível. O POV-Ray permite também a configuração de outros

parâmetros, que podem ser utilizados tanto para personalizar a câmera quanto para simular defeitos de fabricação. Estes são: distância focal, aspecto e eixos diretores.

O outro aspecto compreende a emulação da análise da imagem, que resulta na observação das balizas. Esta emulação se dá através da geometria da descrição da cena virtual, e não da imagem em si. Na cena virtual, cada baliza possui uma posição georreferenciada, \mathbf{x}_{mi} . A partir da posição real da câmera e de sua atitude, é obtida a representação da baliza no referencial da câmera (utilizando-se a equação 2.31 com $\mathbf{v}_i(k) = 0$), o que seria exatamente sua observação perfeita no instante *k* da transformação, $\mathbf{z}_i(k)$.

No entanto, para que o sensor visual seja emulado corretamente, é necessário saber se a baliza é visível na imagem, situação exposta na figura 3.6. Sabendo-se que o eixo focal corresponde ao eixo -z, tal qual já descrito na seção 2.1, uma baliza **z** está na imagem apenas se:

$$\frac{\mathbf{z}_{z}}{-f} > 1 \tag{3.2}$$

$$-\frac{l}{2} \le \mathbf{z}_{x} \frac{-f}{\mathbf{z}_{z}} \le \frac{l}{2} \tag{3.3}$$

$$-\frac{a}{2} \leq \mathbf{z}_y \frac{-f}{\mathbf{z}_r} \leq \frac{a}{2},\tag{3.4}$$

onde f é a distância focal, l é a largura da imagem projetada e a a altura. Se a equação 3.2 não for verdadeira, a baliza não pode ser vista, pois está atrás do plano imagem da câmera. Se qualquer das equações 3.3 ou 3.4 for falsa, a baliza, independentemente de estar em frente ou atrás da câmera, encontra-se fora dos limites da imagem, e também não é visível.

Uma vez que a baliza é visível, **z** representa a sua observação perfeita. No entanto, como tanto imperfeições mecânicas da câmera quanto falhas de análise da imagem incorrem em erros, um ruido gaussiano, de média e desvio padrão determinados na inicialização do sensor visual virtual, é acrescido à observação emitida pelo sensor.

3.4 Integração

O módulo de auto-localização e mapeamento interage com o restante do sistema através do conjunto sensorial do ambiente de simulação, como pode ser visto na figura 3.7.

É interessante notar que o tempo de simulação decorrido entre uma iteração e outra é importante para os processos de simulação e estimação, já que, no modelo cinemático, as informações inerciais precisam ser integradas na variação da postura do dirigível. No entanto, como o simulador não funciona em tempo real, dada a necessidade de renderização da cena pela a câmera virtual, o cronômetro torna-se também um sensor virtual do ambiente de simulação. Quando o sistema é levado ao próximo instante da simulação, através de uma função que emula a passagem de tempo exibida no algoritmo



Fig. 3.6: A projeção da baliza m_1 no plano imagem está fora dos limites da imagem, de modo que a equação 3.3 é falsa. A baliza m_2 tem, no referencial da câmera, coordenada z menor que a distância focal, de modo que a equação 3.2 não é verdadeira. A baliza m_3 é visível na imagem da câmera.

1 a seguir, as leituras dos sensores são atualizadas. Inicialmente, o tempo decorrido desde a chamada anterior é obtido do cronômetro virtual. Neste intervalo, as informações inerciais perfeitas fornecidas pelo simulador dinâmico no instante anterior são integradas no cálculo da nova postura do dirigível e da câmera. Estas são transmitidas ao coordenador de cena, que atualiza a descrição do modelo a ser renderizado.

Muito embora o ambiente de simulação seja sempre atualizado, os sensores por vezes não são modificados, refletindo o fato de os sensores visual e inercial exigirem diferentes intensidades de processamento. Deste modo, de acordo com a configuração do ambiente de simulação, tanto a IMU quanto o sensor visual são atualizados apenas quando a condição de atualização de cada um é verdadeira. Isto reflete o fato de que, na implementação real do sistema embarcado do dirigível, a aquisição de dados da IMU e a aquisição de imagens da câmera ocorrem a taxas de amostragem distintas.

Se o sensor visual deve ser atualizado, uma nova imagem é renderizada e, a partir da postura da câmera, as informações visuais, compreendendo as coordenadas cartesianas das balizas no referencial da câmera, são calculadas. Se novas leituras da IMU virtual devem ser calculadas, os sinais inerciais são repassados ao sensor, no qual um ruído será acrescentado às leituras.

O laço principal do sistema, por sua vez, é exibido no algoritmo 2. Antes da evolução do ambiente de simulação, as leituras da IMU são recuperadas, já que estas devem ser integradas por um período

Algoritmo 1 Atualização temporal do ambiente de simulação

- 1: CRONÔMETRO.ATUALIZA()
- 2: $tempo \leftarrow CRONÔMETRO.TEMPODECORRIDO()$
- 3: *infoInercial* ← SIMDINÂMICA.PEGALEITURA()
- 4: COORDENADORDECENA.ATUALIZA(infoInercial,tempo)
- 5: *posturaAeronave* ← COORDENADORDECENA.PEGAPOSTURAAERONAVE()
- 6: *posturaCâmera* ← COORDENADORDECENA.PEGAPOSTURACÂMERA()
- 7: INTERFACERENDERIZADOR.ATUALIZACENA(posturaAeronave, posturaCâmera)
- 8: se novalmagem então
- 9: SensorVisual.imagem \leftarrow INTERFACERENDERIZADOR.RENDERIZAIMAGEM()
- 10: SENSORVISUAL.ATUALIZALEITURAS(posturaCâmera)
- 11: fim se
- 12: se novoInercial então
- 13: IMU.ATUALIZALEITURAS(*infoInercial*)
- 14: **fim se**



Fig. 3.7: Estrutura global do ambiente de simulação e do módulo de SLAM: O simulador dinâmico e o renderizador funcionam como fontes de dados externas. A ponte entre o ambiente de simulação e o estimador são apenas as leituras dos sensores, permitindo que o ambiente de simulador seja substituído pelos sensores reais implementando apenas a interface pré-estipulada.

só obtido na atualização do ambiente, emulando um INS. Mesmo que as leituras provenientes do sensor inercial não sejam alteradas, o módulo de SLAM atualiza a estimativa do estado do sistema utilizando os sinais inerciais. Por outro lado, a atualização da câmera só ocorre se houver uma nova imagem disponível.

Algoritmo 2 Laço principal do sistema

1:	enquanto continua faça
2:	$tempo \leftarrow \text{Ambiente.Cronômetro.tempoDecorrido()}$
3:	$dadosInerciais \leftarrow Ambiente.IMU.recuperaLeituras()$
4:	Ambiente.próximaEtapa()
5:	SLAM.ATUALIZAINERCIAL(dadosInerciais, tempo)
6:	se Ambiente.novaImagem() então
7:	$dadosVisuais \leftarrow Ambiente.SensorVisual.recuperaLeituras()$
8:	SLAM.ATUALIZACÂMERA(dadosVisuais)
9:	fim se
10:	fim enquanto

3.5 Conclusão

Neste capítulo foi apresentada a estrutura de software desenvolvida para este projeto. Foram detalhadas não apenas as implementações específicas que definem o comportamento do sistema de SLAM, mas também as interfaces de software que serviram de alicerce à implementação, conferindo-lhe versatilidade.

Além do sistema de SLAM, foi também descrito o ambiente de simulação desenvolvido para validação da metodologia e análise visual dos experimentos.

Capítulo 4

Simulação e resultados

Este capítulo descreve os estudos de simulação e os resultados obtidos com o sistema de autolocalização e mapeamento descrito no capítulo 2. O objetivo dos estudos é verificar como a fusão das informações visuais aos dados inerciais influencia na estimação do estado da aeronave, em condições o mais próximas possível das experimentadas em um vôo real.

Inicialmente, na seção 4.1, serão descritas as condições nas quais foi realizada a simulação, incluindo aspectos da dinâmica do dirigível e do funcionamento dos sensores. Posteriormente, na seção 4.2, serão expostos os resultados da simulação propriamente ditos. Enfim, na seção 4.3, é feita a análise do desempenho da metodologia de auto-localização e mapeamento, à luz de informações sobre a convergência do sistema.

4.1 Configuração da simulação

Os estudos de experimentação em simulação foram realizados no ambiente de simulação já descrito na seção 3.3. A seguir, são descritas as condições da dinâmica do dirigível e dos sensores nas quais foi validada a metodologia do capítulo 2.

4.1.1 Condições do vôo

Para determinar a trajetória do dirigível, foram traçadas, no ambiente virtual de atuação do veículo robótico aéreo, uma seqüência de retas de referência, definidas por pontos de latitude e longitude. Associadas às retas, são especificados perfis de altitude, tal que o percurso esteja completamente definido. A partir destas informações, estratégias de controle lateral e longitudinal são utilizadas para que o dirigível siga a trajetória determinada. Como citado no capítulo precedente, o controle lateral utiliza um controlador PID descrito em (Azinheira et al., 2000), enquanto que o controle longitudinal

envolve um controlador de altitude proporcional-derivativo (PD) com realimentação de estado e um controlador proporcional de velocidade (Bueno et al., 2002).

A fim de testar o sistema de localização e mapeamento simultâneos em diversas condições, o percurso escolhido foi composto de diversas curvas e uma variação de altitude, tal que a aeronave aproxima-se do solo conforme segue a trajetória. Além disso, o percurso foi elaborado de maneira a provocar uma variação considerável no número de balizas perceptíveis pelo sistema de visão, permitindo a observação do desempenho do sistema diante desta oscilação. Finalmente, durante o trajeto o dirigível é exposto a vento e turbulência moderados, que afetam sua trajetória.

No que diz respeito ao ambiente de atuação do dirigível, o terreno sobrevoado durante o experimento é irregular e apresenta uma inclinação. Desta forma, as balizas dispostas sobre ele – e detectáveis pelo sistema de visão – não são coplanares, o que demonstra a versatilidade da metodologia.

Antes de iniciar o experimento, a aeronave é posicionada no início da trajetória determinada. Como o resultado da execução é a posição relativa entre o mapa e o dirigível, a posição inicial considerada pelo sistema de auto-localização não importa e não afeta seu desempenho. No entanto, para a melhor visualização dos resultados, o estado do dirigível no instante inicial do experimento foi feito igual ao calculado pelo simulador da dinâmica do mesmo. Deste modo, comparações entre a estimação e o estado real da aeronave são mais facilmente realizadas.

4.1.2 Configuração dos sensores

Um dos aspectos mais importantes da configuração do experimento em simulação, juntamente com o posicionamento das balizas e o número de observações a cada atualização do sistema de visão, é o comportamento dos sensores inerciais e do próprio sistema de visão.

O conjunto de sensores virtuais, descrito em detalhes na seção 3.3.3, é modelado de modo a aproximar-se das suas equivalências reais. Deste modo, como acontece na infraestrutura real da aeronave (Bueno et al., 2002; de Paiva et al., 2006), os sensores inerciais da IMU virtual são atualizados a cada 0,05 *segundo* (20 *Hz*). Suas leituras de aceleração linear ao longo dos eixos do referencial do dirigível, *u*, *v* e *w* (visíveis na figura 4.1(a)), contêm, respectivamente, um erro de média $[-0,03; -0,03; -0,03] m/s^2$ e variância [0,04; 0,04; 0,04]. Já as velocidades angulares da aeronave, dadas pela IMU em torno dos eixos, *p*, *q* e *r* (também destacados na figura 4.1(a)), contêm um erro de média [0,01; 0,01; 0,01] rad/s e variância [0,0025; 0,0025; 0,0025].

O sistema de visão, por outro lado, exige no sistema real um maior tempo de processamento, de modo que seu contraposto virtual é atualizado a cada 0, 1 *segundo* (10 Hz). Além da menor freqüência, o sistema de visão está sujeito a erros com média zero, mas de maior variância. Também é importante notar que, na maioria das estratégias de obtenção direta da posição de balizas, o cálculo



Fig. 4.1: Esquema do funcionamento dos sensores inercial e visual.

da profundidade (ao longo do eixo focal da câmera, z_i na figura 4.1(b)) acarreta em um erro de maior variância. Deste modo, o erro do sensor visual foi modelado com média [0; 0; 0] e variância [2,25; 2,25; 4,0] m^2 .

O experimento possui duração de 130 segundos. Isto significa que há 2600 iterações do algoritmo de SLAM, sendo que em 1300 delas acontece, juntamente com uma nova leitura da IMU, a atualização do sistema de visão. Muito embora o experimento seja controlado e a variância dos sensores, conhecida, a modelagem da variância no sistema de SLAM foi feita como em um sistema real, através da análise principalmente do vetor de inovação e da evolução da variância do sistema. Desta forma, a variância do erro em u, v e w é modelada como [0,1;0,1;0,1], a variância do erro em p, q e r é modelada por [0,003;0,003;0,003] e a variância do erro do sensor visual é modelada como sendo [2,5;2,5;3,5]. As diferenças entre os valores estimados e os reais, indicados no parágrafo anterior, são naturais e provenientes não apenas da estratégia de análise mas também da polarização do erro da IMU e do algoritmo de correspondência entre as balizas. A polarização da IMU faz com que a variância modelada para o erro da IMU seja maior. Por outro lado, o ajuste do modelo do erro do sensor visual leva em consideração a capacidade de associação do algoritmo de correspondência entre previsões não eram associadas, gerando duplicidade de representação no mapa. Isto diminui a eficiência global do SLAM. Com os valores escolhidos cada baliza do ambiente é mapeada apenas uma vez.

Como a IMU é um sensor proprioceptivo de erro polarizado, a tendência natural do uso da IMU seria uma estimativa que deriva do estado real conforme o vôo é realizado. Por outro lado, apesar de exteroceptivo, o sistema de visão possui uma variância maior e provê informações com uma freqüên-

cia menor. Com a metodologia elaborada no capítulo 2, no entanto, espera-se que a observação das balizas permita a correção do estado do sistema, tal que a derivação dos sensores inerciais fique sob controle. Ao mesmo tempo, as leituras da IMU devem fornecer um contraponto mais preciso (de menor variância) para a evolução do dirigível, de modo que a variância do estado do dirigível seja menor que a apresentada pelo sensor visual.

A seguir, são expostos os resultados obtidos.

4.2 Desempenho do sistema

De modo a apresentar os resultados com maior clareza, esta seção é segmentada em quatro etapas. Inicialmente, são exibidos apenas os resultados de posicionamento, que proporcionam uma visão mais direta e intuitiva da eficiência do sistema, já que a trajetória do dirigível pode ser percebida com clareza. Em seguida, são apresentados os resultados da mesma execução do sistema para a atitude do dirigível, que juntamente com a posição estabelecem completamente a postura da aeronave. Posteriormente, são mostrados os resultados da estimativa da velocidade. Finalmente, é exibida a composição do mapa ao longo da execução do experimento simulado.

Como referência, é utilizada a estimativa do estado do sistema obtida utilizando-se apenas os sensores inerciais. Nas figuras comparativas, os valores reais obtido diretamente do simulador estão sempre em azul ou marcados por círculos, os valores da referência obtidos apenas com a IMU estão em verde ou marcados por losangos e os valores obtidos da fusão da IMU com o sistema de visão, conforme a metodologia de SLAM, estão em vermelho ou marcados por triângulos. Em todos os gráficos, o experimento inteiro é exibido (130 segundos e 2600 iterações).

4.2.1 Posição

Na figura 4.2 é mostrada a trajetória real do dirigível, em azul, e a trajetória obtida utilizando apenas um INS, alimentado pela IMU, em verde. Como já mencionado, a longo prazo a estimativa deriva (o erro é crescente). No entanto, para um curto período de tempo, a estimativa do deslocamento do dirigível é precisa, como pode ser percebido no início do vôo. Após 130 segundos, ao fim do experimento (marcado na trajetória real do dirigível), a estimativa do INS não tem nenhuma relação com a posição real.

Por este motivo, como mostra a figura 4.3, o sistema de visão é um bom complemento para a IMU. Uma vez que a câmera é um sensor de baixa freqüência, as informações de alta freqüência da IMU permitem a estimativa da posição com uma maior taxa de atualização. O sistema de visão, por sua vez, não permite que a posição estimada se afaste da posição real, fazendo com que a IMU



Fig. 4.2: A linha azul mostra o trajeto real do dirigível e a linha verde mostra a estimativa da posição do dirigível utilizando apenas os sensores inerciais. O erro aumenta sem limites.

continue calculando a postura do dirigível eficientemente ao longo de todo o vôo. Na figura 4.2, depois de um certo tempo de execução, o uso do INS apenas, calculado a partir das leituras da IMU, não tem nenhum valor para a estimativa da posição georreferenciada da aeronave. Por outro lado, com a metodologia de SLAM, na qual se faz a fusão entre os dados inerciais da IMU e as informações obtidas pelo sistema de visão, a estimativa é precisa durante os 130 segundos do experimento de simulação.

Na figura 4.4, é evidenciado o comportamento do sistema de SLAM em comparação com o simples uso dos sensores inerciais. Durante a primeira curva, após aproximadamente 22 segundos de simulação – e portanto 440 iterações do sistema e 220 atualizações do sensor visual – a estimativa do INS já está a 10 metros da posição real, enquanto que o sistema de SLAM estima a uma distância de 0,58 metro. Rapidamente, após 35 segundos, o erro do INS diverge e a distância entre as posições estimada e real aumenta para 24 metros, ao passo que para o sistema de SLAM a distância entre estimativa e real é de 2,6 metros.

A figura 4.5 mostra o valor da estimativa da posição do dirigível, $\hat{p}(k|k)$, para cada eixo do sistema georreferenciado. Os gráficos mostram a deriva do estimador inercial puro (INS) de maneira mais acentuada, em contraposição à estimativa fornecida pela metodologia de SLAM. Já a figura 4.6 mostra o erro na estimação do posicionamento do dirigível para cada eixo do sistema de coordenadas


Fig. 4.3: A linha vermelha exibe a posição do dirigível conforme determinado pelo sistema de autolocalização e mapeamento, enquanto em azul é mostrada a posição real. A observação das balizas impede a divergência da estimativa.

georreferenciado. O maior desvio da trajetória real ocorre ao longo do eixo Norte, cuja média é 1,57 metro e o desvio padrão é de 1,07 metro . No eixo Leste a média da magnitude do erro é de 0,68 metro (com desvio de 0,42 metro) e na altitude, a magnitude do erro apresenta uma média de 0,67 metro (com desvio de 0,53 metro).

Já a figura 4.7 mostra a norma do erro de posicionamento, o que corresponde à distância entre a posição real da aeronave e a posição estimada da mesma. A maior distância registrada durante o experimento foi de 4,12 metros, com uma média de 2,0 metros e desvio padrão de 1 metro.

4.2.2 Atitude

A figura 4.8 mostra os valores de cada um dos componentes do quaternion presente no estado do sistema de auto-localização, indicando a atitude do dirigível no sistema de coordenadas de terra. Os três valores na parte inferior da figura representam o componente vetorial imaginário do quaternion, enquanto que q(1) representa a parte real do quaternion. Nota-se que, visualmente, a distância entre a estimação e o real é imperceptível, o que não acontece com a linha que representa a estimação obtida apenas do uso do INS. Os valores de q(1) e q(4) variam muito mais que os outros porque o dirigível



Fig. 4.4: Comparação direta das estimativas obtidas apenas da IMU (linha verde) e da IMU juntamente com o sistema de visão (linha vermelha). O trajeto real é mostrado em azul.

faz seu trajeto rotacionando em torno do eixo [0, 0, 1] do sistema de coordenadas georreferenciado. Quando um quaternion é utilizado na representação de rotação, o componente do eixo z da parte vetorial do quaternion e a parte real do mesmo representam rotações no eixo z. A variação nos outros dois componentes acontece devido à turbulência e ao vento, além da trajetória descendente do dirigível. A figura 4.9 fornece os mesmos resultados convertidos em ângulos de Euler.

Os erros em em cada valor do quaternion podem sem observados na figura 4.10, ao passo que a norma do erro é vista na figura 4.11. Considerando o valor absoluto do erro em cada eixo, a média do erro na parte real q(1) é de 0,004, enquanto que para q(2) é de 0,075, para q(3) é de 0,055 e para q(4) é de 0,0038. Apenas para fornecer uma noção mais intuitiva do significado deste erro, com a representação da atitude convertida para ângulos de Euler, as médias dos erros de *rolagem*, *arfagem* e *guinada* são de, respectivamente, -0,0037, -0,0056 e -0,0106 radiano. Já a norma do erro, na representação de quaternion, possui média 0,0124, enquanto que na representação de ângulos de Euler tem média de 0,027 radiano.



Fig. 4.5: Estimativas da posição para cada eixo do referencial de terra. Em verde (losangos), a estimativa utilizando apenas a IMU, em vermelho (triângulos) a estimativa do sistema de SLAM e em azul o valor real (círculos).

4.2.3 Velocidade

A estimação da velocidade linear no referencial de terra apresenta um aspecto interessante do sistema, pois ela não é observada direta ou indiretamente pela câmera, não aparecendo no Jacobiano das equações do sistema de visão (capítulo 2.4). Mesmo assim, dados os termos cruzados da matriz de covariância **P**, que relacionam a velocidade à posição e à orientação do dirigível, é possível corrigir a velocidade durante a etapa de atualização da metodologia proposta. Este comportamento é percebido claramente na figura 4.12. É importante lembrar que o dirigível faz um ciclo em torno do eixo *z* do sistema de coordenadas georreferenciado. Deste modo, a diferença entre a estimativa obtida apenas do inercial (linha verde) e a velocidade real (em azul) para os eixos Norte e Leste não parece aumentar muito e é notada pela diferença de fase entre os gráficos. No entanto, este erro é integrado para o cálculo da posição, resultando no gráfico já mostrado na figura 4.2. Por outro lado, a velocidade de variação da altitude – que tem média -0, 167, já que o dirigível está em trajetória descendente – evidencia a polarização do erro e a disparidade apenas aumenta.

A figura 4.13 mostra o erro de estimação em cada coordenada do referencial de terra utilizando o sistema de auto-localização e mapeamento, enquanto que a figura 4.14 mostra a norma deste erro. A



Fig. 4.6: Erro em cada um dos eixos do sistema de coordenadas georreferenciado.

média da norma do erro é de 0,23 m/s.

4.2.4 Mapa

Para destacar a eficiência da metodologia de SLAM no mapeamento do ambiente, é exibida uma seqüência de quatro gráficos, que representam a evolução do mapa durante um mesmo vôo, juntamente à trajetória do dirigível. Neles, os círculos mostram a localização real de cada baliza, enquanto as cruzes vermelhas mostram a estimativa (quando existente) das mesmas. Para a trajetória, as posições estimada e real seguem o mesmo padrão de cor das balizas.

A primeira figura da seqüência (figura 4.15) mostra o mapa após apenas uma iteração. Neste ponto, a distância média entre a posição real e a estimada das oito balizas observadas é de 2,86 metros. Conforme os marcos são observados novamente, no entanto, o erro diminui consideravelmente, tal que após apenas cinco outras observações das mesmas oito balizas, a distância média cai para 1,56



Fig. 4.7: Norma do erro de posicionamento do sistema de auto-localização.

metro.

Na figura 4.16, após 25 segundos de vôo (500 iterações do sistema e conseqüentemente 250 observações do ambiente), diversas outras balizas foram observadas, sendo que a precisão das mais recentes é naturalmente menor que a das outras, já observadas muitas vezes. Deste modo, a distância média entre as posições estimada e real é de 1,88 metro.

No instante representado pela figura 4.17, o dirigível já mapeou praticamente todas as balizas, apenas duas permanecem não observadas.

Vale notar que, como dirigível inicia estacionário e acelera até aproximadamente dez metros por segundo, as balizas detectáveis no início da trajetória são observadas pelo sistema de SLAM diversas vezes, já que a velocidade do dirigível é muito baixa neste ponto da trajetória. Mesmo assim, neste ponto a média da norma do erro de estimação é 1,78 metro.

A figura 4.18 mostra o dirigível após 1500 iterações do sistema e 750 observações, sobrevoando um terreno já mapeado. O não surgimento de ambigüidades no mapa (duas representações para a mesma baliza) demonstra que o sistema está bem calibrado e, principalmente, que a estratégia de associação das balizas é eficaz no contexto do experimento simulado. Finalmente, a figura 4.19 mostra o mapa ao término do vôo.

Na figura 4.20 é exibida a média da norma do erro de mapeamento a cada iteração do algoritmo de auto-localização e mapeamento. Como já mencionado, as balizas mapeadas no início do experimento



Fig. 4.8: Valor de cada componente do quaternion que representa a orientação do dirigível. Em azul (círculos) o estado real, em vermelho (triângulos) o estimado pelo sistema de SLAM e em verde (losangos) o obtido apenas do INS.

foram observadas muitas vezes. Isso acarreta, tal qual foi descrito no capítulo 2, na dissociação entre o estado da aeronave e posição das balizas. Por este motivo, quando o dirigível observa as mesmas balizas novamente depois de mais de mil iterações, a discrepância entre observação e predição, de acordo com a seção 2.6.3, não provoca alterações significativas no mapeamento. Assim, é fácil perceber na figura o momento em que o dirigível observa balizas já mapeadas no início do experimento, por volta de 65 segundos. Neste ponto, a média do erro cai e fica estável, destacando esta situação. No entanto, após alguns instantes o erro sobe novamente, o que é natural dado que na primeira passagem o dirigível estava mais alto (observando mais balizas ao mesmo tempo) e rápido, de modo que o as balizas são mais suscetíveis a variações nas observações.

Na figura 4.21 é possível ver a média do erro em cada iteração para cada eixo do sistema de coordenadas georreferenciado. A oscilação, principalmente antes do instante t = 40 segundos ocorre conforme novas balizas são mapeadas; posteriormente, os gráficos mostram uma tendência de convergência suave. É interessante observar a queda brusca do erro por volta de 40 segundos de simulação.



Fig. 4.9: Seguindo a mesma notação de cores do gráfico anterior, são exibidos os valores da atitude do dirigível em ângulos de Euler.

Neste ponto, a trajetória do dirigível inicia mais um ciclo e o sistema de visão volta a observar as balizas percebidas no início do experimento. Quando isto ocorre, é possível observar o erro acumulado ao longo da simulação. Como todo o estado do sistema é atrelado pela matriz de covariância, o posicionamento de todas as balizas do mapa – mesmo as que não estão sendo observadas – é corrigido, o que causa a diminuição abrupta do erro.

Enfim, a figura 4.22 mostra o mapa final de outro ângulo, a fim de realçar o erro também na altitude.

4.3 Análise dos resultados

A fim de determinar o real valor dos resultados expostos na seção 4.2 e a convergência da metodologia empregada para obtê-los, é necessário analisar o comportamento do sistema de auto-localização e mapeamento simultâneos em detalhes, de modo a separar as eventuais fragilidades da metodologia das dificuldades naturais do processo de estimação e fusão sensorial.



Fig. 4.10: Erros de orientação do sistema de SLAM em cada componente do quaternion.

4.3.1 Número de associações e a dimensionalidade do sistema

Talvez um dos mais importantes aspectos da análise dos resultados seja o número de correspondências encontradas entre observações e características do mapa a cada execução da etapa de atualização do filtro de Kalman estendido. A figura 4.23 mostra a evolução destes valores. Cabe aqui observar a dimensionalidade do problema: o estado do dirigível, composto de 10 valores representando posição, atitude e velocidade linear no referencial de terra, possui dimensão 9, dado que a representação de quaternions para a atitude é redundante. Além disso, para cada característica do mapa, a dimensão do sistema aumenta em 3. Deste modo, para *N* marcas, a dimensão do sistema é 9+3N.

No instante da atualização, tal qual descrito na seção 2.6.3, cada correspondência entre observação e baliza do mapa fornece ao sistema 3 equações linearmente independentes, correspondentes à



Fig. 4.11: Norma do erro da estimativa do quaternion. A variação acompanha a percebida na posição.

posição relativa da baliza em relação à câmera. No entanto, no referencial global, o que uma única correspondência efetivamente determina é apenas a distância entre a câmera e a baliza observada. Em torno da baliza, infinitas posturas permitem à câmera embarcada obter a mesma observação. Contudo, se três balizas são observadas em uma mesma imagem, é possível determinar unicamente a posição e a orientação do dirigível em relação às balizas. Além disso, em condições ideais de movimentação do dirigível extrair das observações outras relações.

Apesar disso, todas as medidas realizadas a partir das observações são relativas ao posicionamento das próprias balizas, o que significa que todas dependem de um posicionamento global, em geral arbitrariamente definido no ponto de partida do dirigível. Para que o posicionamento global seja corrigido, é necessário que, em algum momento, o sistema obtenha uma informação absoluta de localização, seja por um GPS ou pela identificação de uma baliza de posição global conhecida. Como isto não acontece, a observação, em condições ideais, fornece *rank* 6 + 3N.

Isto, no entanto, não representa um problema, já que o intuito do sistema é posicionar o dirigível com relação ao ambiente, de modo que a ausência de posicionamento global absoluto não impede o funcionamento da estratégia de SLAM apresentada no capítulo 2. Porém, sempre que menos de 3 balizas são observadas em uma mesma iteração, a atualização do sistema é ineficiente. Para mais detalhes, em (Kim, 2004) é desenvolvida a análise de dimensionalidade que demonstra matematicamente que, dadas acelerações adequadas da aeronave, a matriz de observabilidade do problema possui



Fig. 4.12: Velocidade do dirigível em cada eixo do sistema georreferenciado. Em azul (círculos) os valores reais; em verde (losangos) e vermelho (triângulos), respectivamente, a estimativa dos valores utilizando apenas o INS e utilizando a IMU e o sensor visual.

rank 6 + 3N.

Observando novamente o gráfico da figura 4.23, e comparando-o com os gráficos exibidos na seção anterior, é fácil fazer uma associação entre o número de balizas observadas e o erro de estimação, estando o exemplo mais gritante ao fim do experimento. A figura 4.19, que mostra o mapa e a trajetória ao término do experimento, evidencia que nos últimos segundos da simulação o dirigível, além de estar mais próximo do solo (diminuindo a visão da câmera), se afasta da área populada pelas balizas e, conforme a figura 4.23, faz menos correspondências a cada iteração. Esta situação é responsável pela variação do comportamento do sistema de SLAM, observada nas figuras da seção 4.2. Mesmo assim, em todas as figuras é possível notar que, nas últimas 60 iterações (correspondentes aos três últimos segundos de simulação), quando três ou mais correspondências são novamente feitas com freqüência, os erros rapidamente diminuem, destacando o bom funcionamento e conergência da metodologia.



Fig. 4.13: Erro presente na estimação da velocidade para cada um dos eixos

4.3.2 Configuração do sistema e convergência

Em um experimento da metodologia com o dirigível real, o estado do sistema é desconhecido e as únicas informações disponíveis para análise e validação do desempenho do estimador são as medições utilizadas durante a execução da estratégia de SLAM. A análise da escolha do modelo de erro dos sensores e da convergência do sistema pode ser feita, portanto, através da observação da variância do sistema e da inovação que, calculada a cada atualização conforme a equação 2.60, representa a medida mais direta do desempenho do sistema.

Para comprovar a convergência do sistema pode-se observar a variância da posição georreferenciada do dirigível. A figura 4.24 mostra a variância relacionada a cada coordenada da posição do dirigível no referencial de terra. Observa-se que na segunda metade do vôo, os picos de incerteza estão mais baixos; resultado da maior confiança no posicionamento das balizas. No fim do experimento, o erro provocado pelo menor número de correspondências se reflete no aumento da variância,



Fig. 4.14: Norma do erro de estimação da velocidade.

mantendo o sistema coerente em seu funcionamento.

As figuras 4.25, 4.26 e 4.27 mostram a inovação da primeira baliza observada a cada iteração (em vermelho), juntamente com o desvio padrão previsto para o valor da inovação (em azul claro) e com um intervalo de confiança de 95% (em azul escuro). As elevações súbitas do desvio padrão estimado indicam pontos em que é alterada a baliza sendo observada. Fazendo uma análise direta dos gráficos, percebe-se que o modelo de erro é adequado, tal que a inovação está praticamente sempre dentro do intervalo de confiança. Se a estimativa **R** (equação 2.30) da variância do erro de observação representar erros maiores, a precisão da atualização diminui e o erro de estimação aumenta. Por outro lado, se **R** indicar erros menores, balizas já mapeadas podem não ser associadas às suas observações, degradando o desempenho do sistema e gerando no mapa marcos redundantes.

No que diz respeito à consistência do estimador, uma inspeção visual dos gráficos indica que as seqüências de inovação possuem média zero e são brancas. No entanto, a fim de obter um resultado mais rigoroso com relação à média do vetor de inovação como um todo (e não apenas da primeira observação), inicialmente é computada, a cada observação, a inovação normalizada ao quadrado, conforme:

$$q(k) = \mathbf{v}^T \,\mathbf{S}^{-1} \,\mathbf{v}.\tag{4.1}$$

Se a premissa do filtro estiver correta, então q(k) é χ^2 com um número de graus de liberdade m(k)



Fig. 4.15: Início do vôo: apenas algumas balizas detectadas (cruzes) do total de balizas existentes (círculos).

igual ao tamanho do vetor de inovação no instante k (que corresponde ao número de associações entre observações e balizas do mapa no instante k, mostradas na figura 4.23, multiplicado por três). Portanto,

$$E[q(k)] = m(k). \tag{4.2}$$

A equação 4.2 fornece um teste para verificar a não-polarização da inovação. Para aproximar a esperança, foram realizadas dez execuções da simulação, sendo armazenados os valores de q(k) para cada uma delas. A figura 4.28 mostra o valor desta média para uma, cinco e dez execuções. Percebe-se que, conforme o número de execuções incluídas na média aumenta, os valores de q(k) (em vermelho) condensam-se mais próximos de m(k) (em azul), o que é esperado.

Enfim, a figura 4.29 exibe novamente a média para dez execuções, juntamente com um intervalo de confiança de 95%. É evidente que, no ambiente controlado da simulação, a inovação deve se comportar como esperado. No entanto, apesar da inovação estar sempre dentro do intervalo de confiança estipulado, quando o número de correspondências entre o mapa e a observação é baixo, os valores q(k) deixam de estar em torno de m(k), tendendo a ser menores (como acontece em torno da iteração 2500). O fato dos valores não se condensarem em torno do número de correspondências é um indício de problemas de funcionamento do estimador. Este resultado mais uma vez indica que com



Fig. 4.16: Após 25 segundos de vôo: o dirigível evolui, as balizas já observadas diversas vezes tiveram suas posições corrigidas.

menos de três correspondências, a etapa de atualização do filtro de Kalman estendido (seção 2.6.3) é comprometida. No entanto, nos outros casos, a hipótese de que a equação 4.2 é verdadeira com uma probabilidade de 95% é inquestionável.

4.4 Conclusão

Neste capítulo foram exibidos os resultados da aplicação da metodologia de SLAM em um um vôo simulado. Foram detalhadas as condições de vôo, dos sensores e das variáveis envolvidas na metodologia. Diversos gráficos foram mostrados, enfatizando o desempenho do sistema para a estimação de todas as variáveis do estado. Enfim, foi realizada a análise dos resultados sob a ótica da dimensionalidade do problema e da seqüência de inovação.



Fig. 4.17: Após 50 segundos de vôo: Novas balizas são detectadas e incluídas no mapa. Neste trecho há um menor número de balizas associadas a cada iteração e o erro de posicionamento de algumas delas se torna visível.



Fig. 4.18: Após 75 segundos de vôo: todas as balizas já foram incluídas no mapa.



Fig. 4.19: Fim do experimento (130 segundos): apenas correções menores são feitas no mapa.



Fig. 4.20: Média da norma do erro de mapeamento para as balizas presentes no mapa.



Fig. 4.21: Distância média para as balizas reais em cada um dos eixos do referencial de terra.



Fig. 4.22: Resultado do mapeamento de outro ângulo, juntamente com a trajetória completa do dirigível e sua estimativa.



Fig. 4.23: Número de correspondências a cada observação do ambiente.



Fig. 4.24:



Fig. 4.25: Sequência correspondente à inovação no eixo x do referencial de câmera, da primeira observação associada ao mapa a cada iteração.



Fig. 4.26: Seqüência correspondente à inovação no eixo y do referencial de câmera, da primeira observação associada ao mapa a cada iteração.



Fig. 4.27: Seqüência correspondente à inovação no eixo z do referencial de câmera, da primeira observação associada ao mapa a cada iteração.



Fig. 4.28: Esperança de q para 1, 5 e 10 execuções da simulação.



Fig. 4.29: Média de q para 10 execuções, juntamente com o intervalo de confiança de 95%.

Capítulo 5

Conclusões

Os principais focos desta dissertação foram o desenvolvimento, a implementação e a validação em simulação de uma metodologia para o mapeamento e auto-localização simultâneos, visando a navegação precisa de um dirigível robótico aéreo. Para tanto, foi desenvolvida uma estratégia de fusão sensorial entre uma IMU e uma câmera, utilizando um filtro de Kalman estendido. Desta forma, podem ser citados como os principais resultados obtidos:

- o desenvolvimento de uma solução de SLAM em seis graus de liberdade utilizando um EKF para estimar eficientemente a postura do dirigível durante o vôo e a localização das balizas do mapa, que é construído incrementalmente;
- a metodologia elaborada independe de fontes externas de informações, como um mapa prédefinido ou GPS, permitindo sua utilização em uma maior variedade de condições;
- o módulo de localização e mapeamento representa um importante primeiro passo para a implementação do sistema de SLAM no dirigível AS800B, a plataforma robótica do projeto AU-RORA. Os algoritmos, desenvolvidos com este objetivo, estão prontos para a utilização no sistema embarcado do dirigível;
- a flexibilidade da implementação faz do módulo de SLAM e do simulador um arcabouço para testes e validação de outras estratégias de navegação de estrutura similar;
- os resultados obtidos mostram a eficiência e a convergência da metodologia que, na ausência de observações suficientes, degrada de maneira consistente, de acordo com a precisão do sistema inercial. Ao mesmo tempo, voltando a observar um número apropriado de balizas, retoma atividade normal e os níveis de incerteza voltam a cair.

 o desenvolvimento de uma biblioteca para o registro sincronizado das imagens da câmera embarcada e dos dados da IMU, o que constitui uma ferramenta importante para a análise posterior do vôo, permitindo estudos mais detalhados.

Muito embora o módulo de SLAM esteja pronto para a inserção no sistema do dirigível AS800B, os algoritmos de processamento de imagens que fazem da câmera embarcada o sensor visual esperado pela metodologia ainda não são funcionais. Desta forma os primeiros passos dando continuidade a esta dissertação compreendem o desenvolvimento do módulo de visão computacional e a validação da metodologia elaborada no capítulo 2 utilizando dados reais. Além deste horizonte, há também a perspectiva da inclusão de novos sensores no processo de integração sensorial. Assim, os seguintes passos contemplam a evolução deste trabalho:

- desenvolvimento de algoritmos de visão computacional para detecção e rastreamento de balizas artificiais, considerando alternativas de visão estereoscópica ou monocular, como no trabalho de (Davison, 2003);
- elaboração de otimizações computacionais para acomodar o processamento de um mapa com maior número de balizas;
- desenvolvimento de estratégias mais robustas para a correspondência entre balizas e observações;
- validação experimental da metodologia com dados obtidos de um vôo real do dirigível;
- introdução na fusão sensorial de um modelo de odometria visual, baseado na decomposição de homografia. A vantagem desta abordagem é a utilização de uma mesma imagem em modelos diferentes, resultando essencialmente em outra informação, que pode ser integrada às demais;
- desenvolvimento de algoritmos de visão para ambientes não-estruturados, eliminando a necessidade das balizas artificiais no ambiente.

Referências Bibliográficas

- B. Anderson and J. Moore. *Optimal Filtering*. Prentice Hall, 1979.
- J. R. Azinheira, E. C. de Paiva, J. J. G. Ramos, and S. S. Bueno. Mission path following for an autonomous unmanned airship. In *International Conference on Robotics and Automation*, San Francisco, USA, 2000.
- Tim Bailey. *Mobile Robot Localisation and Mapping in Extensive Outdoor Environments*. PhD thesis, Australian Centre for Field Robotics, University of Sydney, August 2002.
- J. Borenstein and Y. Koren. Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions* on Systems, Man, and Cybernetics, 19(5):1179–1187, 1989.
- Geovany A. Borges. *Cartographie de l'environnement et localisation robuste pour la navigation de robots mobiles*. PhD thesis, Département de Robotique, LIRMM/CNRS/Université Montpellier II, 2002.
- T. G. R. Bower. *Perception: Essays in the honor of James J. Gibson*, chapter The evolution of sensory systems, pages 141–153. Cornell University Press, 1974.
- M. Brady. Progress towards a system that can acquire pallets and clean warehouses. In *Fourth International Symposium of Robotics Research*, 1987.
- S. S. Bueno, J. R. Azinheira, J. G. Ramos, E. C. de Paiva, P. Rives, A. Elfes, J. R. H. Carvalho, and G. F. Silveira. Project aurora - towards an autonomous robotic airship. In *Workshop on Aerial Robotics. IEEE International Conference on Intelligent Robot and Systems*, Lausanne, Switzerland, 2002.
- W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 114(1-2):3–55, 1999.

- José A. Castellanos, J. M. M. Montiel, José Neira, and Juan D. Tardós. The spmap: A probabilistic framework for simultaneous localization and map building. *IEEE Transactions on Robotics and Automation*, 1999.
- José A. Castellanos, José Neira, and Juan D. Tardós. Multisensor fusion for simultaneous localization and map building. *IEEE Transactions on Robotics and Automation*, 2001.
- R. Chatila and J. P. Laumond. Position referencing and consistent world modeling for mobile robots. In *IEEE International Conference on Robotics and Automation*, 1985.
- Paul Churchland. *Matéria e consciência: uma indrodução contemporânea à filosofia da mente.* Editora Unesp, 1998.
- I. J. Cox. Blanche an experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Transactions on Robotics and Automation*, 1991.
- Michael Csorba. *Simultaneous Localisation and Map Building*. PhD thesis, Department of Engineering Science, University of Oxford, 1997.
- Andrew Davison. Real-time simultaneous localisation and mapping with a single camera. In *Ninth Intl. Conference on Computer Vision*, 2003.
- E. C. de Paiva, S. S. Bueno, S. B. V. Gomes, J. J. G. Ramos, and M. Bergerman. A control system development environment for aurora's semi-autonomous robotic airship. In *Int. Conference on Robotics and Automation*, 1999.
- E. C. de Paiva, J. R. Azinheira, J. G. Ramos, A. Moutinho, and S. S. Bueno. Project aurora: infrastructure and flight control experiments for a robotic airship. *Journal of Field Robotics*, 23, 2006.
- M. W. M. G. Dissanayake, P. Newman, Hugh F. Durrant-Whyte, Steve Clark, and M. Csorba. An experimental and theoretical investigation into simultaneous localisation and map building. In *The Sixth International Symposium on Experimental Robotics VI*, pages 265–274, London, UK, 2000. Springer-Verlag.
- M. W. M Gamini Dissanayake, Paul Newman, Steven Clark, and Hugh F. Durrant-Whyte. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on Robotics and Automation*, 2001.
- A. Elfes. Sonal-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, 1987.

- A. Elfes. *Occupancy Grids: A Probabilistic Framework for Robot Perception and Navigation*. PhD thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, 1989.
- A. Elfes, S. S. Bueno, M. Bergerman, and J. J. G. Ramos. A semi-autonomous robotic airship for environmental monitoring missions. In *Int. Conf. on Robotics and Automation*, 1998.
- A. Elfes, S. S. Bueno, M. Bergman, E. C. de Paiva, G. J. Ramos, and J. R. Azinheira. Robotic airships for exploration of planetary bodies with an atmosphere: Autonomy challenges. *Autonomous Robots*, 2003.
- G. Giralt, R. Sobek, and R. Chalita. Multi-level planning and navigation system for a mobile robot: a first approach to hilare. In *Sixth International Joint Conference on Robotics and Automation*, 1979.
- P. Goel, S. Roumeliotis, and G. Sukhatme. Robust localization using relative and absolute position estimates, 1999.
- S. B. V. Gomes and J. J. G. Ramos. Airship dynamic modeling for autonomous operation. In *Int. Conference on Robotics and Automation*, Leuven, Belgium, 1998.
- M. Grewal and A. Andrews. Kalman Filtering Theory and Practice. Prentice-Hall, 1993.
- J. Guivant and E. Nebot. Solving computational and memory requirements of feature-based simultaneous localization and mapping algorithms. *IEEE Transactions on Robotics and Automation*, 2003.
- José Guivante. *Efcient Simultaneous Localization and Mapping in Large Environments*. PhD thesis, Australian Centre for Field Robotics, University of Sydney, May 2002.

D. Guzzoni, A. Cheyer, L. Julia, and K. Konolige. Many robots make short work. AI Mazazine, 1997.

John Heil. *Philosophy of Mind*. Routledge, 2 edition, July 2004.

Ted Honderich, editor. The Oxford Companion to Philosophy. Oxford University Press, 1995.

http://www.oberhumer.com/opensource/lzo/. Lzo real-time data compression library.

http://www.oberhumer.com/opensource/ucl/. Ucl data compression library.

http://www.povray.org/. Povray - persistence of vision raytracer.

http://www.zlib.net/. zlib compression library.

- E. Hygounenc, I. Jung, P. Souères, and S. Lacroix. The autonomous blimp project of laas-cnrs: achievements in flight control and terrain mapping. *Intl. Journal of Robotics Research*, 2004.
- R. E. Kalman. A new approach to linear filtering and prediction problems. *Trans. ASME, Journal of Basic Engineering*, 1960.
- G.A. Kantor, D. Wettergreen, J.P. Ostrowski, and S. Singh. Collection of environmental data from and airship platform. In *SPIE Conference on Sensor Fusion and Decentralized Control in Robotic Systems IV*, 2001.
- Jonghyuk Kim. *Autonomous Navigation for Airborne Applications*. PhD thesis, Australian Centre for Field Robotics, University of Sydney, May 2004.
- L. Kleeman. Optimal estimation of position and heading for mobile robots using ultrasonic beacons and dead reckoning. In *IEEE International Conference on Robotics and Automation*, 1992.
- Jack Langelaan and Steve Rock. Towards autonomous uav flight in forests. In AIAA Guidance, Navigation and Control Conference, 2005.
- John Locke. *Ensaio Acerca do Entendimento Humano*, volume XVIII of *Os Pensadores*. Editora Abril, 1973.
- Deborah Modrak. Aristotle: The Power of Perception. University of Chicago Press, June 1989.
- Stefano Panzieri, Feredica Pascucci, and Giovanni Ulivi. An outdoor navigation system using gps and inertial platform. *IEEE/ASME Transactions on Mechatronics*, June 2002.
- I. T. Ruiz, Y. Pelillot, D. M. Lane, and C. Salson. Feature extraction and data association for auv concurrent mapping and localisation. In *IEEE International Conference on Robotics and Automation*, 2001.
- Ken Shoemake. Animating rotation with quaternion curves. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 45–54, 1985.
- R. Smith, M. Self, and P. Cheeseman. A stochastic map for uncertain spatial relationships. In *Fourth International Symposium on Robotics Research*, 1987.
- R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous Robot Vehicles*. Springer Verlag, 1988.

- S. Snyder, B. Schipper, L. Vallot, N. Parker, and C. Spitzer. Differential gps/inertial navigation approach/landing flight test results. *IEEE Transactions on Aerospace and Electronic Systems Magazine*, 1992.
- Salah Sukkarieh. Low Cost, High Integrity, Aided Inertial Navigation Systems for Autonomous Land Vehicles. PhD thesis, Australian Center for Field Robotics, University of Sydney, 2000.
- S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.
- T. Tsumura. Survey of automated guided vehicle in japanese factory. In *IEEE International Conference on Robotics and Automation*, 1986.
- Alessandro C. Victorino. La commande référencée capteur: une approache robuste au problème de la navigation, localisation et cartographie simultanée pour un robot d'intérieur. PhD thesis, INRIA Universidade de Nice, Sophia Antipolis, France, 2002.
- R. Welch and D. Warren. *Handbook of perception and human performance*, chapter Intersensory Interactions. Wiley, 1986.
- D. Wimmer, M. Bildstein, K. H. Well, M. Schlenker, P. Kingl, and B. Kröplin. Research airship "lotte": development and operation of controllers for autonomous flight phases. In *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, 2002.
- Allen Wu, Eric Johnson, and Alison Proctor. Vision-aided inertial navigation for flight control. In *AIAA Guidance, Navigation and Control Conference*, 2005.
- H. Zhang and J. Ostrowski. Visual servoing with dynamics: Control of an unmanned blimp, 1998. URL citeseer.ist.psu.edu/article/zhang98visual.html.

Trabalhos Publicados Pelo Autor

- Cesar Castro, Samuel Bueno, Alessandro Victorino. "Towards a SLAM solution for a robotic airship". *3rd International Conference on Informatics in Control, Automation & Robotics* (ICINCO 2006), Setúbal, Portugal, 2006.
- Cesar Castro, Alessandro Victorino, Samuel Bueno. "Auto-localização e mapeamento para um dirigível robótico". XVI Congresso Brasileiro de Automática (CBA 2006), Salvador, Brasil, 2006.
- 3. Cesar Castro, Leonardo Dinnouti, Samuel Bueno, Geraldo Silveira. "Uma Abordagem Integrada Para Controle Servo Visual E Monitoramento Em Tempo Real". *XV Congresso Brasileiro de Automática* (CBA 2004), Gramado, Brasil, 2004.

Apêndice A

Gravação sincronizada de dados

A capacidade de armazenar seqüências de vídeo obtidas da câmera embarcada, bem como de outros sensores existentes no dirigível, é de fundamental importância para o desenvolvimento de estudos sobre a plataforma real. Isto se deve, em parte, à dificuldade envolvida na realização de um vôo - o que requer a mobilização de uma equipe de pessoas, de equipamentos, de infra-estrutura de apoio, atendendo a restrições climáticas, operacionais, etc; é crucial, portanto, aproveitar ao máximo os ensaios de vôo na obtenção de novos dados. Outra motivação é a possibilidade e necessidade de utilizar estes dados em análises e estudos realizados *offline*.

Parte do equipamento embarcado disponível no dirigível é uma câmera de vídeo digital conectada ao processador PC/104 por um barramento *firewire* (IEEE 1394). Está também à disposição um sistema de aquisição de imagens desta câmera, que disponibiliza uma imagem por vez. Também no dirigível, como já mencionado ao longo deste trabalho está uma unidade de medição inercial (IMU), que fornece ao sistema as acelerações lineares do dirigível, juntamente às taxas de rotação do mesmo. Além disso, estão embarcados no dirigível uma sonda de vento, um altímetro e um GPS. Para que tanto a câmera e a IMU quanto os outros sensores embarcados sejam utilizados eficientemente em um instante posterior, é necessário não apenas armazenar as informações, mas fazê-lo de maneira sincronizada, tal que haja uma relação temporal entre os dados.

Em vista desta necessidade, foi elaborada uma metodologia de gravação, contemplando compactação e armazenamento de imagens juntamente a informações dos outros sensores e informações de sincronia, além de uma configuração de hardware compatível com os requerimentos do dirigível, de modo que, além da câmera e dos sensores, possa ser embarcado juntamente um disco rígido para o armazenamento dos dados coletado. A estrutura da biblioteca que realiza estas funções é detalhada a seguir.

A.0.1 A biblioteca

Conforme mencionado, no que diz respeito a imagens, a solução desenvolvida faz papel similar ao de um *codec* de vídeo. Durante a gravação, a imagem recebida do sistema de aquisição de imagens do dirigível é anexada às informações complementares dos outros sensores, bem como informações de sincronia, como um *timestamp* da coleta dos dados. Este conjunto pode, então, ser concatenado a outros similares, obtidos no decorrer do vôo. A cada *N* conjuntos de dados, o sistema compacta o bloco inteiro sem perdas. O bloco de dados comprimidos é, por sua vez, concatenado a um conjunto de informações necessárias para a descompressão e interpretação do bloco quando da descompressão. Este novo conjunto de dados pode ser armazenado em disco conforme desejado. Na etapa de visualização (ou análise), a biblioteca realiza o processo oposto: o bloco é descomprimido nos diversos conjuntos sensoriais que o compõe e estes, destacados, resultam em um par dados/imagem. Como será detalhado a seguir, as informações utilizadas para a descompressão são descritivas o bastante para que seja possível acessar uma imagem sem a necessidade de descomprimir todos os blocos. Isto garante menor consumo de memória e menor tempo de processamento na busca de dados específicos.

Um aspecto importante da arquitetura da biblioteca é sua versatilidade. A biblioteca não distingue imagens; apenas conjuntos de dados de um tamanho especificado. Estes conjuntos de dados são armazenados seqüencialmente, comprimidos, acompanhados de um cabeçalho próprio, contendo o tamanho do bloco de dados e uma informação de tempo, destinada à sincronia dos dados. Além disso, o conjunto de blocos (imagens) que faz o vídeo contém também um cabeçalho, cuja informação mais importante é o número de blocos presentes.

Quaisquer outros dados que se deseje armazenar juntamente com a imagem – como por exemplo todo o conjunto de leituras da telemetria – podem ser incluídos, juntamente com a imagem, para compressão, desde que seja possível separá-los posteriormente. Isto quer dizer que, desde que o tamanho da estrutura anexada a imagem seja conhecido, qualquer informação pode ser incluída.

Além disto, o arquivo de vídeo é gravado bloco a bloco (ou seja, de *N* em *N* imagens) e as informações dos cabeçalhos, que não são compactados, são suficientes para que a descompressão também possa ser feita bloco a bloco. Deste modo, é possível requisitar à biblioteca um bloco aleatório, sem a necessidade de descompressão do restante dos quadros.

Outras questões abordadas no desenvolvimento da biblioteca de gravação, além da sincronia com os dados de telemetria, são a velocidade e a taxa de compressão. No caso de gravação de imagens advindas da câmera real embarcada no dirigível, estas duas variáveis são requisitos de grande importância, já que há no sistema embarcado limitações temporais (sob pena de que os dados, se gravados em freqüência muito baixa, não tenham grande valia) e de espaço (pois um vídeo armazenado sem perdas consome um espaço muito grande em disco). Em face a esta dificuldade, a biblioteca de gravação foi feita de maneira a facilitar a troca de algoritmo de compressão, permitindo que diversas soluções sejam testadas.

Deste modo, foram realizados testes com as seguintes bibliotecas:

- LZO (http://www.oberhumer.com/opensource/lzo/) a biblioteca LZO foi utilizada pela NASA em missões interplanetárias e possui uma alta velocidade de compressão. No entanto, a taxa de compressão é baixa, o que implica na gravação de grandes arquivos no disco rígido. Como a velocidade de gravação do disco é um dos principais gargalos do sistema, o desempenho do algoritmo é insatisfatório;
- UCL (http://www.oberhumer.com/opensource/ucl/) o mesmo desenvolvedor da LZO elaborou o algoritmo de compressão da biblioteca UCL. Com desempenho oposto ao do obtido com LZO, a taxa de compressão é elevada, mas a velocidade de compressão é baixa;
- ZLIB (http://www.zlib.net/) esta biblioteca utiliza mais memória que as outras opções investigadas e a velocidade de compressão é menor que a obtida com a LZO. No entanto, sua taxa de compressão elevada compensa a velocidade de processamento.

Vale ressaltar que a metodologia para a gravação de imagens no dirigível está em andamento e alterações no hardware do projeto podem alterar a escolha da biblioteca de compressão utilizada na gravação sincronizada.

A.0.2 Estrutura de Software

Na figura A.1 está representado um esquema simplificado da comunicação entre os diversos módulos do sistema de gravação em uma operação de armazenamento, enfatizando as diferentes origens que a imagem pode ter, bem como a anexação de outras informações sensoriais e do *timestamp* no quadro gerado pela biblioteca.

A interface de compressão gerencia o processo de montagem do quadro, coletando a imagem e os outros dados e invocando a biblioteca de compactação para gerar um bloco e as informações de descompactação. O bloco é então levado ao gerenciador de quadro, que anexa a este o instante da geração (que é utilizado em conjunto com quaisquer outras informações para a posterior sincronia dos dados sensoriais) realiza o armazenamento em disco.



Fig. A.1: Estrutura do sistema de gravação.

Apêndice B

Quaternions

Descritos pela primeira vez pelo matemático irlandês William Howan Hamilton em 1843, quaternions são uma extensão não-comutativa de números complexos. Assim como números complexos podem ser representados por uma coordenada bi-dimensional no plano complexo, quaternions podem ser representados por uma coordenada no espaço quadri-dimensional. Desta forma, um quaternion pode ser descrito por:

$$\mathbf{q} = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k},\tag{B.1}$$

ou de maneira simplificada, como a composição de um valor real e um vetor tridimensional imaginário:

$$\mathbf{q} = \left[w \left[x \, y \, z \right] \right] \tag{B.2}$$

Os números w, x, y, z são reais. Já i, j, k, por sua vez, se relacionam a números reais através da seguinte relação:

$$i^2 = j^2 = k^2 = ijk = -1 \tag{B.3}$$

Da equação B.3 são obtidas as seguintes:

$$ij = k, \quad jk = i, \quad ki = j$$

 $ji = -k, \quad kj = -i \quad ik = -j$ (B.4)

Quaternions podem ser utilizados para representar uma diversidade de transformações, mas são na prática utilizados para representar rotações (Shoemake, 1985). A representação de rotação por um quaternion (4 valores) é mais compacta que a representação por uma matriz 3x3 ortonormal (9 valores). Além disso, a partir de um eixo e um ângulo, é fácil compor o quaternion que representa a rotação, o que não é tão simples utilizando ângulos de Euler ou matrizes. A equação abaixo mostra

um quaternion que representa uma rotação de α radianos em torno do vetor $\mathbf{v} = [x, y, z]$:

$$\mathbf{q} = \cos\frac{\alpha}{2} + \sin\frac{\alpha}{2} \left(\frac{xi + yj + zk}{\|\mathbf{v}\|}\right) \tag{B.5}$$

Quando concatenando diversas rotações em um computador, que possui precisão finita, inevitavelmente ocorrem erros de arredondamento. Um quaternion com um pequeno erro continua representando aproximadamente a rotação esperada após sua normalização (um quaternion de rotação possui sempre norma unitária). Por outro lado, uma matriz 3x3 que deixe de ser ortonormal por conta de erros de precisão exige uma correção computacionalmente muito mais intensiva.

A representação por quaternions evita também o problema conhecido como *gimbal lock*, que acontece com ângulos de Euler quando as rotações independentes fazem com que dois eixos de rotação coincidam. Nesta situação, um dos graus de liberdade é perdido.

A transformação de um vetor \mathbf{v} por um quaternion \mathbf{q} é dada por

$$[0, \mathbf{v}'] = \mathbf{q} \otimes [0, \mathbf{v}] \otimes \overline{\mathbf{q}}, \tag{B.6}$$

em que o vetor original é utilizado como a parte imaginária de um quaternion de parte real zero e $\overline{\mathbf{q}}$ representa o conjugado de \mathbf{q} ,

$$\overline{\mathbf{q}} = [w, -x, -y, -z]. \tag{B.7}$$

A multiplicação entre cada par de quaternions, indicada na equação por \otimes , é por sua vez computada tal qual acontece com números complexos, utilizando, portanto, as relações explicitadas no início desta seção.

Apesar da utilização de quaternions possuir as vantagens já descritas, em diversas situações é necessário efetuar a conversão a ângulos de Euler, como no caso da transferência da transformação a outro sistema ou biblioteca. Neste caso, a conversão pode ser feita através das seguintes equações:

$$a = \arctan\left(\frac{2yw - 2xz}{1 - 2y^2 - 2z^2}\right) \tag{B.8}$$

$$r = \arcsin\left(2xy + 2zw\right) \tag{B.9}$$

$$g = \arctan\left(\frac{2xw - 2yz}{1 - 2x^2 - 2z^2}\right) \tag{B.10}$$

onde *a*, *r* e *g* representam, respectivamente, arfagem, rolagem e guinada. É necessário, no entanto, ter especial cuidado com singularidades quando a rolagem se aproxima de ± 90 graus. Estes casos devem ser tratados especialmente.